

# Shape from Sharp and Motion-Blurred Image Pair

C. Paramanand · A. N. Rajagopalan

Received: 29 December 2012 / Accepted: 28 November 2013 / Published online: 11 December 2013  
© Springer Science+Business Media New York 2013

**Abstract** Motion blur due to camera shake is a common occurrence. During image capture, the apparent motion of a scene point in the image plane varies according to both camera motion and scene structure. Our objective is to infer the camera motion and the depth map of static scenes using motion blur as a cue. To this end, we use an unblurred–blurred image pair. Initially, we develop a technique to estimate the transformation spread function (TSF) which symbolizes the camera shake. This technique uses blur kernels estimated at different points across the image. Based on the estimated TSF, we recover the complete depth map of the scene within a regularization framework.

**Keywords** Motion blur · Blur kernel · Transformation spread function · Belief propagation · Depth map

## 1 Introduction

Motion blur has long been an important topic in the image processing literature. During image capture, due to the effect of averaging of light intensities at the camera sensors, the relative motion between the scene and the camera results in motion blur. With the increase in the usage of handheld cameras, the study of motion blur is receiving considerable attention. Although the main objective has been to remedy the effect of blur, works also exist that focus on inferring valuable information about the object or camera motion by using blur

as a cue (Boracchi 2009; Caglioti and Giusti 2008; Dai and Wu 2008; Klein and Drummond 2005; Sellent et al. 2011). In this paper, our objective is to use motion blur induced by camera motion for recovering the depth map of static scenes. Depth from motion blur can serve as an alternative to depth from defocus (DFD) (Chaudhuri and Rajagopalan 1999) (a popular approach for structure estimation). DFD techniques generally do not allow for any camera motion. In contrast, depth from motion blur will be useful for autonomous vehicle navigation, and for enabling 3D awareness in handheld imaging devices wherein camera motion inevitably results in motion blur.

In a 3D scene, when a camera moves, the apparent motion of the scene points in the image plane depends on the value of depth. Hence, motion blur can serve as a cue for scene depth (Fox 1988). The blur at an image point is characterized by a blur kernel also known as the point spread function (PSF). For the case of pure in-plane camera translations, the shape of the PSF reflects the camera motion. In fact, the shape of the PSF is preserved at all the image points except for a scale factor which is related to the scene depth (Sorel and Flusser 2008). Points that are near to the camera are more blurred than those that are farther. Based on the extent of blurring at a point, the depth values can be estimated (Sorel and Flusser 2008; Paramanand and Rajagopalan 2010a, 2012). However, when the camera is free to undergo rotations or axial motion, the shape of the blur kernel varies across different image points. Depth estimation in such a scenario is quite a challenging task as compared to the case wherein camera motion is restricted to in-plane translations.

Our work in this paper focuses on estimating depth from a pair of images in which one of the observations is a blurred version of the other due to camera shake. For depth recovery, presence of translational component in the camera motion is a must. When the depth of the scene is constant, instead of relat-

C. Paramanand (✉) · A. N. Rajagopalan  
Department of Electrical Engineering, Indian Institute of Technology, Madras, Chennai, India  
e-mail: paramanand@gmail.com

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

ing the observations using a space-variant PSF, the blurred image can be modeled as a weighted average of geometrically transformed (homography) instances of the reference image. Each homography is assigned a weight that denotes the fraction of the exposure duration for that transformation. We refer to the weights of the transformations as the transformation spread function (TSF). The TSF is a generalization of the motion blur kernel; while the blur kernel denotes the motion of a point light source at a pixel during image capture, the TSF denotes the transformations undergone by the image plane due to camera motion (Paramanand and Rajagopalan 2010b; Tai et al. 2011; Whyte et al. 2010). It must be noted that small displacements between the two observations get subsumed within our TSF-model. However, it is not practically feasible to accommodate large shifts within a TSF. In this paper, we assume that the camera displacement between the two observations is small (which is typical of camera shakes).

For scenes with depth variations, due to the effect of parallax, the transformations undergone by the image points will vary not only due to camera motion but also with respect to scene depth. Hence, a single TSF cannot be used to relate the reference and the blurred images. Interestingly, the TSF (for a given depth) provides valuable information about the camera motion that can be used to deduce blur kernels of image points at other depths. Given an unblurred-blurred image pair, we determine a few blur kernels at randomly chosen points in the image. We develop a method to simultaneously solve for relative depth values at the locations of the blur kernels and the TSF (corresponding to a particular depth which is regarded as the reference). This step is somewhat analogous to the approach followed in structure from motion wherein camera poses and scene geometry are solved using feature correspondences (Hartley and Zisserman 2004). While the blur kernel signifies the motion undergone by a particular scene point during the exposure, the TSF represents the different poses undergone by the camera. Using the knowledge of the reference TSF, we determine a dense depth map of the complete scene (up to a scale factor). In fact, the 2D translational motion scenario addressed in (Paramanand and Rajagopalan 2012) turns out to be a very special case of the above situation.

### 1.1 Related Work

“Shape from optical defocus”, which is a well-established area of research, involves utilization of optical blur as a cue for the estimation of 3D structure (Chaudhuri and Rajagopalan 1999; Favaro et al. 2008). Depth estimation is based on determining the extent of blurring in observations that are captured with different lens settings. The practical applicability of these methods is limited in the sense that most of the techniques assume that the camera motion is

restricted and/or known (Bhavsar and Rajagopalan 2012). Also, the lens settings used for obtaining the observations have to be known correctly. Furthermore, in some scenarios, there is need for additional hardware in the form of telecentric lenses to avoid magnification (Watanabe and Nayar 1998).

In contrast, motion blur is very commonly prevalent and since it is also a cue for depth, motion blurred images have been used for depth estimation (Fox 1988; Lin and Chang 2006). These methods assume that the scene can be approximated by a set of planar patches. Simultaneous depth estimation and restoration from space-variant blurred images is performed using variational methods in (Favaro et al. 2004; Favaro and Soatto 2004; Sorel and Flusser 2008). Favaro et al. (2004) consider the scenario in which the observations of a scene are simultaneously motion blurred and optically defocused. The latent image, global motion field (due to 2D translational motion) and the scene structure are jointly estimated. In (Favaro and Soatto 2004), the scene is considered to be composed of different objects move along different directions. The motion field (for each independent motion), depth-map and the restored image are estimated from the motion blurred observations that are captured with different exposure times. Sorel and Flusser (2008) have proposed a technique to estimate the 3D shape and the restored image using two observations which are blurred in different ways. They consider the PSF to be of arbitrary shape (due to non-uniform velocity of the camera). In their method, the PSF is initially determined from the blurred observations by choosing regions of constant depth. Using this PSF, the image and the depth-map are simultaneously estimated. Our related work in (Paramanand and Rajagopalan 2010a, 2012) uses two observations of the scene and determines the depth at a point by estimating the extent of relative blur between the observations. Depth estimation is formulated as a recursive state estimation problem and is solved using an unscented Kalman filter (UKF). It must be mentioned that in all the above depth estimation techniques, the relative motion between the camera and scene is restricted to fronto-parallel translations wherein the extent of blurring at a point is directly related to the scene depth. However, realistic camera shakes include even in-plane rotations (Levin et al. 2009). Zheng et al. (2011) consider the case of motion blur in 3D scenes due to general camera motion in low light scenes. They identify the extremal points of the traces in the blurred image due to point light sources. From these point correspondences, the camera motion and the depth values at the light sources are recovered.

Although the focus of this paper is on depth estimation, for completeness, we would like to digress a little to briefly review notable works on *restoration* of motion blurred images. Traditionally, image restoration techniques have modeled blurring due to camera shake as a convolution with a single blur kernel (Fergus et al. 2006; Shan et al. 2008).

However, since camera shakes typically induce space-variant blur (Levin et al. 2009), recent deblurring techniques effectively model the motion blurred image as an average of projectively transformed instances of the latent image (i.e., using the TSF notion). Tai et al. (2011) have proposed a deblurring scheme based on modifying the Richardson Lucy deconvolution technique for space-variant blur. However, they do not address the problem of determining the blurring function (the TSF). Whyte et al. (2010, 2012) propose an image restoration technique for motion blur arising due to non-uniform camera rotations. They represent the blurring function on a 3D grid corresponding to the three directions of camera rotations. For the case of blind image restoration, the kernel estimation framework in (Fergus et al. 2006) is employed. When a noisy version of the original image is available, a least-squares energy minimization approach is used for finding the blurring function. In another deblurring scheme by Gupta et al. (2010), the latent image and blurring function are iteratively solved by considering camera motion to be comprised of 2D translations and in-plane rotations. In the restoration technique by Joshi et al. (2010), sensors are attached to the camera to determine the blurring function. Tai et al. (2010b) propose a deblurring scheme that uses coded exposure and some simple user interactions to determine the PSF. Hirsch et al. (2011) propose a new approach to restore non-uniform motion blur by using the efficient filter flow framework. It must be mentioned that none of these deblurring methods account for changes in blur due to depth variations. Although, the methods in (Sorel and Flusser 2008) and (Xu and Jia 2012) address this issue, they make the restrictive assumption that blurring is primarily due to 2D camera translations.

In this work, we develop a method to estimate the 3D scene structure from two observations captured such that one of the images is a motion blurred version of the other. Although our emphasis in this work is on depth estimation, the strategy adopted herein can serve as a valuable step forward towards efficient deblurring of 3D scenes.

We wish to point out that the scenario of unblurred-blurred images considered in this paper is not uncommon as is evinced by several existing works including those of Sorel and Sroubek (2009), Yuan et al. (2007), Whyte et al. 2010, 2012, Babacan et al. (2010). A sharp unblurred image can be captured with a high ISO gain setting and a low exposure time while a blurred observation can be captured with longer exposure time and low ISO. The image restoration techniques in (Sorel and Sroubek 2009; Yuan et al. 2007; Whyte et al. 2010, 2012; Babacan et al. 2010) adopt the use of an unblurred-blurred image pair that are captured with different ISO gains and exposure times. In some recent techniques (Sellent et al. 2011; Portz et al. 2012) that determine the optical flow of a dynamic scene, the ISO gain and exposure interval are altered across video frames to arrive at unblurred-blurred observations. Tai et al. (2010a) simultaneously cap-

ture a low-resolution unblurred image and a high-resolution blurred observation using a hybrid camera. A similar principle can even be adopted to get an unblurred-blurred pair at the same resolution.

The initial step of our method is to estimate the camera motion that led to blurring. The deblurring algorithms in (Whyte et al. 2010; Hu and Yang 2012) estimate the TSF and the latent image by alternate minimization. However, for 3D scenes, these methods cannot be employed to infer the camera motion. To this end, we propose to use blur kernels estimated at different randomly chosen image locations. In a small region around a point, we assume the blur to be space-invariant and estimate the blur kernel using patches cropped from the image pair. The blur kernel at an image point depends on the camera motion as well as its depth. We determine the TSF (which denotes the camera motion) corresponding to a reference depth value that resulted in the observed blur kernels at different image points. We relate the TSF with the observed PSFs by a linear equation and simultaneously solve for the TSF and the relative depth values at the locations of the PSFs. During image capture, out of all possible transformations, the reference image would have undergone very few transformations. Hence, we impose a sparsity prior on TSF by including its  $l_1$ -norm in the objective function. Using the estimated TSF, we infer camera motion which then enables us to perform depth estimation for the whole scene. We model the depth map as a Markov random field (MRF) and formulate a cost function using the image generation model and the MRF prior. The maximum a posteriori (MAP) value of the depth label at each pixel is estimated using loopy belief propagation (BP) (Felzenszwalb and Huttenlocher 2006).

In this work, we treat the camera motion to primarily consist of in-plane rotations and 2D translations since this motion is typical of camera shakes (Levin et al. 2009; Gupta et al. 2010). This model holds good in many scenarios due to the fact that image stabilizers present in camera lenses have built-in gyro sensors that compensate for out-of-plane camera rotations, but *not* in-plane translations and rotations (Canon 2012). Our framework can even tolerate some out-of-plane rotation as discussed later. Furthermore, we show that our approach can also be extended to scenarios when there is axial translation or out-of-plane rotation during camera shake.

**Contributions** This work advances the state-of-the-art in the area of non-uniform motion blurring in many ways. (i) While the depth estimation works in (Sorel and Flusser 2008), (Paramanand and Rajagopalan 2012) and (Favaro and Soatto 2004) restrict the camera motion to pure in-plane translations, our motion blur model allows for a more general camera motion. (ii) Even though several deblurring works such as (Whyte et al. 2012; Gupta et al. 2010) use the TSF model, the blur variation due to parallax in 3D scenes is neglected. In this work,

we elegantly relate the PSF at each point with the reference TSF by accounting for parallax effects. (iii) In (Whyte et al. 2012; Hu and Yang 2012), the TSF is directly estimated from image intensities assuming a constant depth scenario. However, these methods cannot be applied for 3D scenes. Our method uses locally estimated blur kernels which enables us to account for parallax effects. (iv) For a given a set of blur kernels (corresponding to different points in the scene) our algorithm can classify them according to their depth values. (v) The method described in (Zheng et al. 2011) determines the 3D structure of point light sources in dark scenes. In contrast, we recover a *dense* depth map of scenes with normal lighting. (vi) Finally, in comparison to our own earlier work (Paramanand and Rajagopalan 2010b), this paper contains significant extensions. In (Paramanand and Rajagopalan 2010b), TSF estimation was based on minimizing a least squares error between image intensities sans any prior. The method required user assistance for selecting a region within which the depth remained constant. The technique proposed in this paper is fully automatic and there is no user interaction involved. The proposed BP-based framework allows us to incorporate regularization more effectively in comparison to (Paramanand and Rajagopalan 2010b) which used local computations for depth estimation. The technique proposed in this paper is applicable even when there is camera out-of-plane rotation which was not considered in (Paramanand and Rajagopalan 2010b). The experimental section in this paper is far more exhaustive and thorough as compared to (Paramanand and Rajagopalan 2010b).

The paper is organized as follows: In Sect. 2, we discuss the motion blur model. We describe TSF estimation in Sect. 3. While depth recovery is presented in Sect. 4, experimental results are discussed in Sect. 5. We conclude with Sect. 6.

## 2 Motion Blur Model

When the camera motion is not restricted to in-plane translations, the apparent motion of scene points in the image will vary at different locations resulting in space-variant blurring. The convolution model with a single blur kernel does not hold in such a scenario. However, when the scene depth is constant, the blurred image can be accurately modeled as a weighted average of the warped instances of the original image (Paramanand and Rajagopalan 2010b; Tai et al. 2011; Whyte et al. 2010; Gupta et al. 2010).

Let the image of a scene captured by a still camera be denoted by  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ . Let  $\mathbf{X} = [X \ Y \ Z]^T$  denote the spatial coordinates of a point in the scene with the camera center as the origin. The projection of  $\mathbf{X}$  in the image plane  $(x, y)$  is given by  $x = \frac{vX}{Z}$  and  $y = \frac{vY}{Z}$  where  $v$  denotes the ‘focal length’. Using homogeneous coordinates, the image

point  $\mathbf{x} = [x \ y \ 1]^T$  can be written as  $K_v \mathbf{X}$ . In this discussion,  $K_v$  is assumed to be of the form

$$K_v = \begin{bmatrix} v & 0 & 0 \\ 0 & v & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Due to camera motion during image capture, at each instant of time  $\tau$ , the coordinates of the 3D point  $\mathbf{X}$  changes to  $\mathbf{X}_\tau = R_\tau \mathbf{X} + T_\tau$  with respect to the camera where  $T_\tau = [T_{X_\tau} \ T_{Y_\tau} \ T_{Z_\tau}]^T$  is the translation vector. The rotation matrix  $R_\tau$  is parameterized (Whyte et al. 2012) in terms of  $\theta_X$ ,  $\theta_Y$  and  $\theta_Z$  (the angles of rotation about the three axes) using the matrix exponential

$$R_\tau = e^{\Theta_\tau} \text{ where } \Theta_\tau = \begin{bmatrix} 0 & -\theta_{Z_\tau} & \theta_{Y_\tau} \\ \theta_{Z_\tau} & 0 & -\theta_{X_\tau} \\ -\theta_{Y_\tau} & \theta_{X_\tau} & 0 \end{bmatrix} \quad (2)$$

We consider that all of the scene points are at a distance  $d_o$  from the camera. Consequently, the point  $\mathbf{x}_\tau$  at which  $\mathbf{X}_\tau$  gets projected in the camera can be obtained through a homography  $H_\tau$  as  $\mathbf{x}_\tau = H_\tau \mathbf{x}$  where

$$H_\tau = K_v \left( R_\tau + \frac{1}{d_o} T_\tau [0 \ 0 \ 1] \right) K_v^{-1} \quad (3)$$

Let  $g_\tau$  denote the image captured at time instant  $\tau$ . For the sake of simplicity, we use the same notation ( $\mathbf{x}$ ) for the homogeneous coordinates as well as for the coordinates in the image plane. Then we can write  $g_\tau(\mathbf{x}) = f(H_\tau^{-1} \mathbf{x})$  where  $H_\tau^{-1}$  denotes the inverse of  $H_\tau$  [(since  $g_\tau(H_\tau \mathbf{x}) = f(\mathbf{x})$ ]. The blurred image  $g$  can be considered as the average of the light intensities observed in the image plane during exposure. The blurred image intensity at an image point  $\mathbf{x}$  is given by

$$g(\mathbf{x}) = \frac{1}{T_e} \int_0^{T_e} f(H_\tau^{-1} \mathbf{x}) d\tau \quad (4)$$

where  $T_e$  is the total exposure duration.

Note that, when averaging over time, the temporal information (order of the set of transformations undergone by the reference image) is lost. But this is a non-issue for the problem on hand. The blurred image can be more appropriately modeled in terms of the reference image, by averaging it over the set of possible transformations (resulting from the camera motion). Let  $\mathbf{T}$  denote the set of all possible transformations and  $\lambda$  denote a transformation. We define the *transformation spread function* (TSF)  $\omega : \mathbf{T} \rightarrow \mathbb{R}_+$  as a mapping from the set  $\mathbf{T}$  to non-negative real numbers. For each transformation  $\lambda \in \mathbf{T}$ , the value of the TSF  $\omega(\lambda)$  denotes the fraction of the total exposure duration for which the camera was in the position that caused the homography  $H_\lambda^{-1}$  on the image coordinates. It is to be noted that the term  $\lambda$  denotes the transformation parameters corresponding to the homography matrix  $H_\lambda^{-1}$ , and does not indicate a time instant. The blurred image

can be written as an average of the warped images weighted by the TSF  $\omega$ . i.e.,

$$g(\mathbf{x}) = \int_{\lambda \in \mathbf{T}} \omega(\lambda) f(H_\lambda^{-1}(\mathbf{x})) d\lambda \quad (5)$$

When the camera motion is not restricted, the paths traced by scene points in the image plane can vary across the image resulting in space-variant blur. However, the blurring operation can be described by a single TSF using Eq. (5). The TSF depicts the camera motion during exposure. For instance, if the camera undergoes only in-plane rotations, the TSF will have non-zero weights only for the rotational transformations. Analogous to a blur kernel, the TSF satisfies the relation  $\int_{\lambda \in \mathbf{T}} \omega(\lambda) = 1$  (assuming equal amount of light energy is involved in the formation of  $f$  and  $g$ ).

Alternatively, the blurred image  $g$  can also be modeled with a space-variant PSF  $h$  as

$$g(\mathbf{x}) = f *_v h(\mathbf{x}) = \int f(\mathbf{x} - \mathbf{u}) h(\mathbf{x} - \mathbf{u}, \mathbf{u}) d\mathbf{u} \quad (6)$$

where  $h(\mathbf{x}, \mathbf{u})$  denotes the blur kernel at the image point  $\mathbf{x}$  as a function of the independent variable  $\mathbf{u}$ . The PSF  $h(\mathbf{x}, \mathbf{u})$  represents the displacements undergone by a point light source at  $\mathbf{x}$  during the exposure and is weighted according to the fraction of the exposure time the light source stays at the displaced position. It can be written (Sorel and Flusser 2008) as

$$h(\mathbf{x}, \mathbf{u}) = \frac{1}{T_e} \int_0^{T_e} \delta(\mathbf{u} - \bar{\mathbf{x}}_\tau) d\tau \quad (7)$$

where  $\delta$  indicates the 2D Dirac Delta function and  $\bar{\mathbf{x}}_\tau$  is the instantaneous displacement. Consider a point light source at  $\mathbf{x}$  in the reference image  $f$ . At time  $\tau$ , due to the transformation  $H_\tau^{-1}$  on the image co-ordinates, the displacement of the point light source is  $\bar{\mathbf{x}}_\tau = H_\tau \mathbf{x} - \mathbf{x}$ . Substituting, Eq. (7) in (6), we get back the blurring model of Eq. (4).

The PSF  $h$  at an image point  $\mathbf{x}$  can be obtained from the TSF  $\omega$  by finding the displacement induced due to each of the possible transformations. This relationship can be written as

$$h(\mathbf{x}, \mathbf{u}) = \int_{\lambda \in \mathbf{T}} \omega(\lambda) \delta(\mathbf{u} - (H_\lambda \mathbf{x} - \mathbf{x})) d\lambda \quad (8)$$

Substituting  $h(\mathbf{x}, \mathbf{u})$  from Eq. (8) in Eq. (6) gives us the blurring model in Eq. (5). Note that if the camera motion is confined to 2D translations, the PSF and TSF will be equivalent to one another.

## 2.1 Effect of Parallax

Note that, when there are depth variations in the scene, due to parallax, the apparent motion of the scene points in the

image plane is also related to the 3D structure of the scene. Consequently, the blur kernel at an image point depends even on its depth value. Hence, blurring cannot be modeled using Eq. (5) with a single TSF. However, since blurring is due to camera motion (which is the same for all the points in the scene), if the TSF corresponding to a particular depth is given, the induced transformations at other depths can be inferred. For the sake of simplicity, initially, we restrict the camera motion (and thereby the set  $\mathbf{T}$ ) to be composed of in-plane rotations and 2D translations. The effects of translation along the optical axis and out-of-plane rotation will be discussed later.

Following our motion model, the set of transformations  $\mathbf{T}$  becomes a 3D space defined by the axes  $t_x$ ,  $t_y$  and  $\theta$ . Let  $R_\lambda$  (which denotes rotation by an angle  $\theta_\lambda$  about the optical axis), and  $T_\lambda = [T_{X_\lambda} \ T_{Y_\lambda} \ 0]^T$  be the rotation and translation, respectively, that caused the transformation  $\lambda$  for points at depth  $d_o$ . The homography given in the Eq. (3) simplifies to the following 2D transformation

$$H_\lambda(\mathbf{x}) = \begin{bmatrix} \cos \theta_\lambda & -\sin \theta_\lambda \\ \sin \theta_\lambda & \cos \theta_\lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_{x_\lambda} \\ t_{y_\lambda} \end{bmatrix}$$

where  $[t_{x_\lambda} \ t_{y_\lambda}]^T$  and  $\theta_\lambda$  represent the translation and rotation parameters of  $H_\lambda$ , respectively. The translation parameters for this depth are given by  $t_{x_\lambda} = \frac{v T_{X_\lambda}}{d_o}$  and  $t_{y_\lambda} = \frac{v T_{Y_\lambda}}{d_o}$ . Let  $\omega_o$  denote the TSF corresponding to a depth of  $d_o$ . The TSF  $\omega_o$  defines the weight for a transformation  $\lambda \in \mathbf{T}$  (caused due to  $R_\lambda$  and  $T_\lambda$ ) for points at depth  $d_o$ . For a point  $(i, j)$  [(having depth  $d(i, j)$ ], the weight corresponding to the camera pose remains the same but the effective transformation gets altered according to  $d(i, j)$ . The rotation parameter  $\theta_\lambda$  is not affected by depth changes. However, the translation parameters get scaled according to the relative depth  $k(i, j) = \frac{d(i, j)}{d_o}$  as

$$t_{x_{\lambda_{ij}}} = \frac{t_{x_\lambda}}{k(i, j)}, \quad t_{y_{\lambda_{ij}}} = \frac{t_{y_\lambda}}{k(i, j)} \quad (9)$$

Let  $H_{\lambda_{d(i,j)}}$  denote the transformation with the parameters  $t_{x_{\lambda_{ij}}}$ ,  $t_{y_{\lambda_{ij}}}$  and  $\theta_\lambda$ . Then the PSF at a point  $\mathbf{x} = (i, j)$  is given by

$$h(\mathbf{x}, \mathbf{u}) = \int_{\lambda \in \mathbf{T}} \omega_o(\lambda) \delta(\mathbf{u} - (H_{\lambda_{d(i,j)}}(\mathbf{x}) - \mathbf{x})) d\lambda \quad (10)$$

The blurred image of the 3D scene  $g$  can be related to the original image  $f$  through the space variant blurring operation (Eq. (6)), wherein the PSF  $h$  depends on the camera motion (denoted by the TSF  $\omega_o$ ) and the depth map  $d$  [(according to Eqs. (9), (10)].

In practice, we quantize each coordinate axis of the transformation space  $\mathbf{T}$  to get a discrete set of transformations (also denoted as  $\mathbf{T}$ ). The TSF  $\omega_o$  defined on the discrete transformation space  $\mathbf{T}$  can be considered as a vector in  $\mathbb{R}^{N_T}$  where  $N_T$  is the total number of transformations present in  $\mathbf{T}$ . Let

$h(i, j, ;)$  denote the discrete blur kernel at a pixel  $\mathbf{p} = (i, j)$ . For  $l = 1 \dots N_T$ , let  $\omega_o(l)$  denote the weight corresponding to the  $l$ th transformation in  $\mathbf{T}$  and  $(i_{k(i,j)}^l, j_{k(i,j)}^l)$  denote the co-ordinates when a transformation  $H_{d(i,j)}$  is applied on  $\mathbf{p}$ . Analogous to Eq. (10), the discrete form of the PSF at each pixel  $(i, j)$  can be written as

$$h(i, j; m, n) = \sum_{l=1}^{N_T} \omega_o(l) \delta_K \left( m - (i_{k(i,j)}^l - i) \right), \\ n - (j_{k(i,j)}^l - j) \quad (11)$$

where  $\delta_K$  denotes the 2D Kronecker delta function. Note that the  $l$ th transformation in  $\mathbf{T}$  is modified according to the relative depth  $k(i, j)$  while arriving at  $(i_{k(i,j)}^l, j_{k(i,j)}^l)$ . When the coordinates  $(i_{k(i,j)}^l, j_{k(i,j)}^l)$  take non-integer values, we assign values to its neighboring pixels by bilinear interpolation.

## 2.2 3D Translation

In many situations, the camera motion can be modeled as translations along all three directions. This is encountered in situations when the camera axial motion is dominant (Alveratos et al. 1989). The homography caused due to a translation of  $T_\lambda = [T_{X_\lambda} \ T_{Y_\lambda} \ T_{Z_\lambda}]^T$  simplifies to the following 2D transformation [(from Eq. (3)]

$$H_\lambda \mathbf{x} = \begin{bmatrix} \xi_\lambda & 0 \\ 0 & \xi_\lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_{x_\lambda} \\ t_{y_\lambda} \end{bmatrix}$$

where  $\xi_\lambda$  represents scaling. The transformation space  $\mathbf{T}$  is parameterized by  $t_x, t_y$  and  $\xi$  axes. For the depth value of  $d_o$ , the scale and the translation parameters of  $H_\lambda$  are related to the camera motion as

$$\xi_\lambda = \frac{d_o}{d_o + T_{Z_\lambda}}, \quad t_{x_\lambda} = \frac{v T_{X_\lambda}}{d_o + T_{Z_\lambda}}, \quad t_{y_\lambda} = \frac{v T_{Y_\lambda}}{d_o + T_{Z_\lambda}} \quad (12)$$

We can relate the scale and translation parameters of  $H_{\lambda d(i,j)}$  [for a point having relative depth  $k(i, j)$ ] to the parameters at depth  $d_o$  as

$$\xi_{\lambda d(i,j)} = \frac{k(i, j)}{k(i, j) + \left(\frac{1}{\xi_\lambda} - 1\right)}, \quad t_{x_{\lambda d(i,j)}} = \frac{t_{x_\lambda} \left(1 + \left(\frac{1}{\xi_\lambda} - 1\right)\right)}{k(i, j) + \left(\frac{1}{\xi_\lambda} - 1\right)}, \\ \text{and } t_{y_{\lambda d(i,j)}} = \frac{t_{y_\lambda} \left(1 + \left(\frac{1}{\xi_\lambda} - 1\right)\right)}{k(i, j) + \left(\frac{1}{\xi_\lambda} - 1\right)} \quad (13)$$

From these parameters of the transformation  $H_{\lambda d(i,j)}$ , the blur kernel at any point can be written in terms of its relative depth and the reference TSF  $\omega_o$  using Eq. (11).

To model camera motion consisting of out-of-plane rotation as well as translations, one has to include even the rotation parameters  $\theta_X$  and  $\theta_Y$  as the coordinates of the transformation space  $\mathbf{T}$ . As with the case of in-plane rotation, the rotation parameters do not vary with respect to depth. The translation and scale parameters get altered as per the relation in Eq. (9) (for in-plane translations) and Eq. (13) (for general translations).

## 3 TSF Estimation

Since the extent of blurring at a point depends both on the camera motion and depth, for inferring depth from blur, the knowledge of camera motion (TSF) becomes necessary. Prior non-uniform deblurring techniques (Gupta et al. 2010; Whyte et al. 2012; Hu and Yang 2012) assume that the scene depth does not vary. The approach is to jointly solve for the TSF and the latent image that best explains the blurred image intensity values by incorporating regularization priors. However, for 3D scenes, even the scene depth has to be estimated. In this section, we use the blur kernels that are estimated at randomly selected locations across the image for determining the camera motion (characterized by the TSF). Since, the blur kernel at a point is dependent on the camera motion as well as the depth, we solve for both TSF and the depth values of the selected image points from the blur kernels using an alternate minimization approach.

From  $f$  and  $g$ , we randomly select  $N_p$  locations (denoted by  $\mathbf{p}_l$ ) across the image. Around a point  $\mathbf{p}_l$ , we crop small patches  $f_l$  and  $g_l$  respectively, for  $l = 1$  to  $N_p$ . We approximate the blur to be space-invariant within this patch. The blur kernel  $h_{\mathbf{p}_l}$  is obtained by minimizing the following cost function using conjugate gradient method

$$\underset{h_{\mathbf{p}_l}}{\operatorname{argmin}} \| (g_l - h_{\mathbf{p}_l} * f_l) \|_2 \quad (14)$$

where  $h_{\mathbf{p}_l} * f_l$  denotes the convolution operation. Thus, we get blur kernels  $h_{\mathbf{p}_1} \ h_{\mathbf{p}_2} \dots h_{\mathbf{p}_{N_p}}$  at  $N_p$  locations  $\mathbf{p}_1 \ \mathbf{p}_2 \dots \mathbf{p}_{N_p}$ , respectively. Let  $d_{\mathbf{p}_1} \ d_{\mathbf{p}_2} \dots d_{\mathbf{p}_{N_p}}$ , respectively, denote the depth corresponding to  $\mathbf{p}_1 \ \mathbf{p}_2 \dots \mathbf{p}_{N_p}$ . We solve for the TSF  $\omega_o$  corresponding to a particular value of depth  $d_o$  and also the relative depth values at the location of the blur kernels through alternate minimization.

### 3.1 Camera Motion Estimation (Given Depth)

Considering a particular value of depth  $d_o$  as the reference, and  $\omega_o$  to be the induced TSF at  $d_o$  due to the camera motion. Let  $k_{\mathbf{p}_1} \ k_{\mathbf{p}_2} \dots k_{\mathbf{p}_{N_p}}$ , respectively denote the relative depths of  $\mathbf{p}_1 \ \mathbf{p}_2 \dots \mathbf{p}_{N_p}$  where  $k_{\mathbf{p}_l} = \frac{d_{\mathbf{p}_l}}{d_o}$ . Suppose that these relative depth values are known (this assumption will be relaxed

subsequently). In Eq. (11), we see that the displacement of a pixel  $(i, j)$  due to a transformation depends on the relative depth  $k(i, j)$ . Also, each component of the blur kernel  $h(i, j; m, n)$  is a weighted sum of the components of the TSF  $\omega_o$ . Consequently, if the relative depth  $k_{\mathbf{p}_l}$  is known, the blur kernel  $h_{\mathbf{p}_l}$  can be expressed as  $h_{\mathbf{p}_l} = M_l \omega_o$  for  $l = 1 \dots N_p$ , where  $M_l$  is a matrix whose entries are determined by the location of blur kernel  $\mathbf{p}_l$ , its relative depth  $k_{\mathbf{p}_l}$  and the interpolation coefficients. If the number of elements in the blur kernel is  $N_h$ , then the size of the matrix  $M_l$  will be  $N_h \times N_T$ . A column of  $M_l$  corresponds to a transformation in  $\mathbf{T}$  while a row corresponds to a location in the blur kernel. To generate the entries of  $M_l$ , for each column (that corresponds to a transformation say  $\lambda$ ), the displacement  $H_{\lambda, d_{\mathbf{p}_l}}(\mathbf{p}_l) - \mathbf{p}_l$  of the point  $\mathbf{p}_l$  is evaluated. A unity entry is made at the row which corresponds to the coordinates of the displacement. If the coordinates have fractional values, the neighboring rows are entered based on the bilinear interpolation principle. Since for each column, only a few rows have non-zero entries, the matrix  $M_l$  will be sparse. The matrix-vector product  $M_l \omega_o$  and the operation of Eq. (11) are equivalent.

The term  $M_l \omega_o$  relates the PSF at a point  $\mathbf{p}_l$  with the reference TSF. For determining  $\omega_o$ , we make use of all the estimated PSFs (at  $\mathbf{p}_1 \mathbf{p}_2 \dots \mathbf{p}_{N_p}$ ). In order to express the relationship between the reference TSF and the PSFs with a single equation, we stack all the  $N_p$  blur kernels as a vector  $\bar{h}$ . For each point  $\mathbf{p}_l$ , we generate the matrix  $M_l$  (according to its location and relative depth value). The matrices  $M_l$ s for  $l = 1 \dots N_p$  are concatenated to arrive at a large sparse matrix  $M$ . The relationship between the observed blur kernels and the TSF  $\omega_o$  can then be written as

$$\bar{h} = M \omega_o \quad (15)$$

The matrix  $M$  is of size  $N_p N_h \times N_T$ . In practice,  $\omega_o \in \mathbf{R}^{N_T}$  will be a sparse vector because the camera motion during exposure would result in very few transformations out of all possible elements of  $\mathbf{T}$ . Hence, while solving for  $\omega_o$  we impose a sparsity constraint. To get an estimate of the TSF that is consistent with the observed blur kernels as well as sparse, we minimize the following cost function.

$$\operatorname{argmin}_{\omega_o} \|\bar{h} - M \omega_o\|_2^2 + \lambda_s \|\omega_o\|_1 \quad (16)$$

The sparsity constraint on  $\omega_o$  is enforced by the  $l_1$ -norm term in Eq. (16). To minimize the cost function, we use the toolbox available at (Liu et al. 2009) since it is fast and can handle sparse data of large size. For minimization of Eq. (16) we use the ‘nnLeastR’ function of the toolbox which leads to solution that has only nonnegative components. The implementation of this function is based on the algorithm proposed in (Liu and Ye 2009).

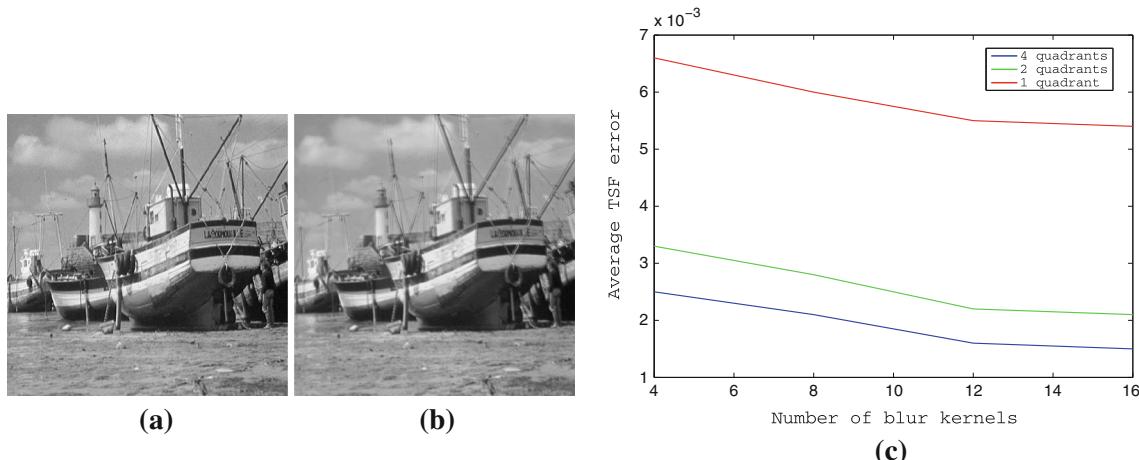
### 3.2 Alternate Minimization

Since, the relative depth values  $k_{\mathbf{p}_1} k_{\mathbf{p}_2} \dots k_{\mathbf{p}_{N_p}}$  are not known, we cannot directly solve for the reference TSF from Eq. (16). We start with an initial estimate of the relative depth values and estimate the reference TSF. From this TSF, we refine our estimate of the relative depth. We repeat the estimation of the reference TSF with the current estimate of the relative depth values and vice versa for a few iterations.

Initially, the relative depths of all the points are assigned to unity. From the estimate of  $\omega_o$  obtained by solving Eq. (16), we determine the value of relative depth  $k_{\mathbf{p}_l}$  at a point  $\mathbf{p}_l$  which generates the blur kernel that is close to the observed blur kernel  $h_{\mathbf{p}_l}$ . The values of the relative depth lie within a range  $k_{\min}$  to  $k_{\max}$ , which denote the minimum and maximum possible values, respectively, of the relative depth. At each point  $\mathbf{p}_l$ , we vary the value of the relative depth from  $k_{\min}$  to  $k_{\max}$  in fine steps (around 0.1). For each label  $\hat{k}_{\mathbf{p}_l}$ , we generate the corresponding blur kernel  $h_{\hat{k}_{\mathbf{p}_l}}$  (from Eq. (11)) by scaling the translations in  $\omega_o$  by  $\hat{k}_{\mathbf{p}_l}$ . The value of  $\hat{k}_{\mathbf{p}_l}$  that minimizes  $\|h_{\hat{k}_{\mathbf{p}_l}} - h_{\mathbf{p}_l}\|_2$  is assigned to  $k_{\mathbf{p}_l}$ . The entries of the matrix  $M$  are then recomputed using the updated relative depths. The estimate of  $\omega_o$  is then updated (Eq. (16)). Following this procedure, after a few iterations (< 10), the estimates of the TSF and the relative depth converge (do not vary). The estimated TSF  $\omega_o$  contains the weights of the transformations corresponding to the unity relative depth. Note that, from the blur kernels, only relative depth values can be estimated. For absolute depth values, the knowledge of the actual values of the camera translation is required.

It must be noted that due to the inherent ambiguity in the estimation of translation and depth values, there can be multiple solutions to the joint TSF and relative depth estimation problem. At a point  $\mathbf{p}_c$ , a transformation with translation parameters  $t_{x_c}$  and  $t_{y_c}$  and relative depth value  $k_{\mathbf{p}_c}$  will lead to the same shift as another transformation with translation parameters  $\gamma t_{x_c}$  and  $\gamma t_{y_c}$  but by considering the relative depth to be equal to  $\frac{k_{\mathbf{p}_c}}{\gamma}$ . Therefore, there can be many possible combinations of relative depth values and TSFs that lead to the observed blur kernels ( $h_{\mathbf{p}_1} h_{\mathbf{p}_2} \dots h_{\mathbf{p}_{N_p}}$ ). From the alternate minimization scheme, we arrive at one of the possible solutions.

We tested our TSF estimation technique based on alternate minimization for the case of in-plane translations and rotations on a synthetic example. We also studied the effect of number of blur kernels and their locations on the accuracy of TSF estimation. The reference image was chosen as shown in Fig. 1a. We assumed a ramp depth map for the scene (shown in Fig. 2c). The relative depth values ranged between 1 and 2.3. We assumed a TSF  $\omega_o$  for the reference depth (where relative depth is unity). The transformation space  $\mathbf{T}$  was defined as follows:  $t_x$  and  $t_y$  ranged between -11 and



**Fig. 1** Synthetic experiment—effect of number of PSFs and their locations on TSF estimation. (a) Reference image. (b) Blurred observation. (c) Average TSF estimation error with respect to number of blur kernels (for all the three cases)

11 in steps of 1 pixel,  $\theta$  took values between  $-1.5$  and  $1.5$  in steps of  $0.25^\circ$ . Consequently, the total number of elements in the transformation space was  $23 \times 23 \times 13 = 6877$ . Along the  $\theta$  axis, the value of the TSF was concentrated between  $-0.75^\circ$  and  $-0.25^\circ$ , and it was negligible at other angles. From the reference TSF, at each pixel, we generated the blur kernel based on its relative depth (Eq. (11)) and subsequently arrive at the blurred observation shown in Fig. 1b.

We considered three different cases; each with different extent of spatial spread. We divided the image equally into four quadrants. In the first case, the locations of the PSFs were chosen randomly but in such a way that the coordinates were restricted to lie within the same quadrant. While in the second case, PSF locations spanned two quadrants, in the third case, patches were selected from all the four quadrants. For each of the three cases, we used 4, 8, 12, and 16 PSFs for TSF estimation. Initially, the PSFs were estimated from image patches. Subsequently, TSF was estimated using the proposed alternate minimization scheme. This process was repeated five times by randomly varying the locations of the PSFs. For comparing the true and estimated TSFs, we defined an error measure  $e_{\text{TSF}}$  which is based on comparing the generated PSFs at different locations. We selected 16 equally spaced locations spread across the image. We generated blur kernels from the true and estimated TSFs at these locations based on their known depth values to arrive at two sets of PSFs. The rms error between these sets of PSFs was regarded as the error measure  $e_{\text{TSF}}$ . The error  $e_{\text{TSF}}$  in the estimate of TSF with respect to the known TSF was evaluated in each of our cases. The mean and variance of the estimation error (over all the five trials) were evaluated. Figure 1 shows the mean error plotted against the number of blur kernels for each case. From the error curves it is apparent that the TSF was estimated accurately when the blur kernels were chosen from all the four quadrants. In fact, the reduction in the

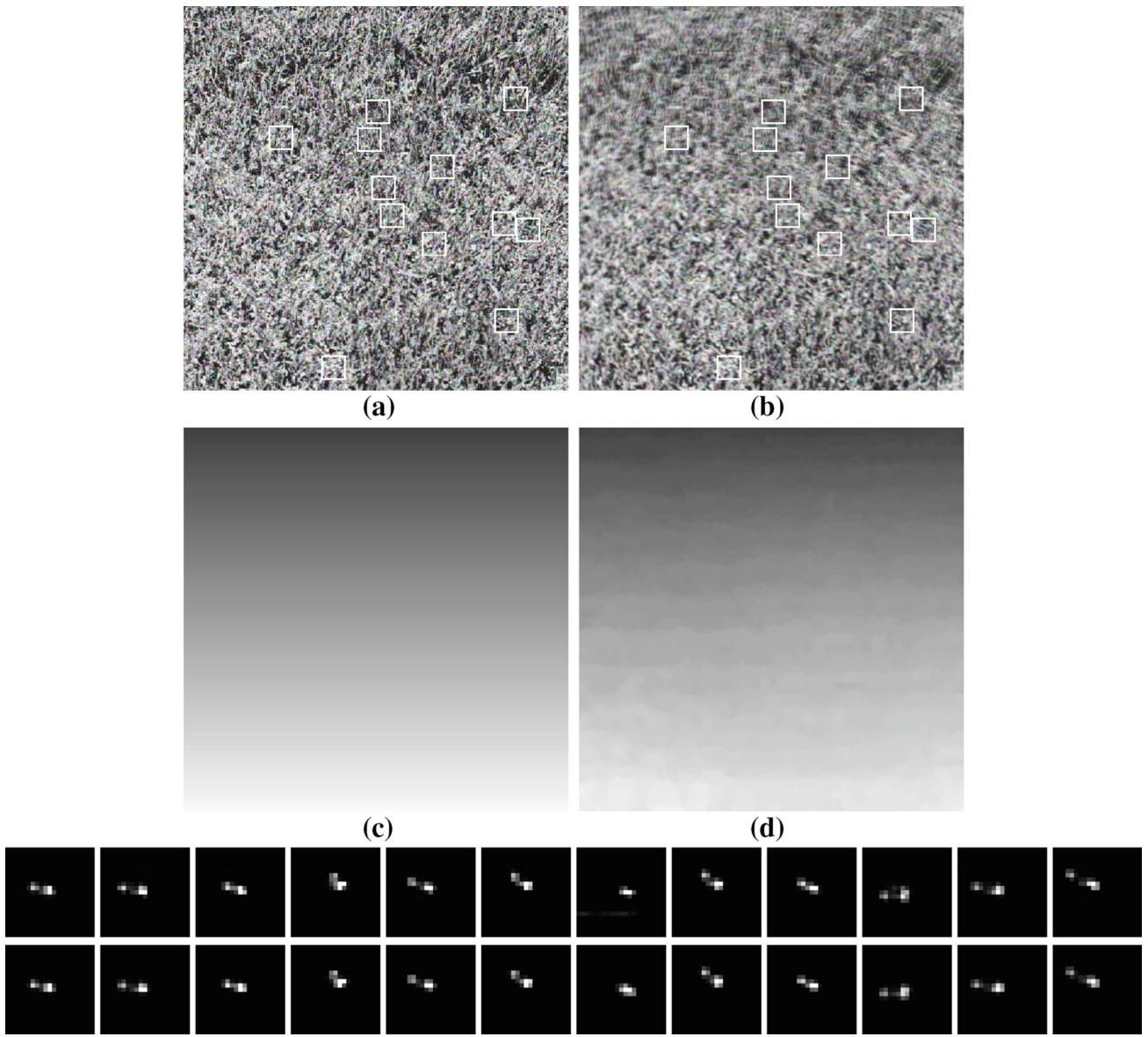
TSF estimation error is significantly more when the number of quadrants is increased from 1 to 4 (from case 1 to case 3) rather than when the number of PSFs is increased. This shows that for TSF estimation to be accurate, the locations of the PSFs should be sufficiently spread across the image. The variance in the estimate was of the order of  $10^{-7}$  for the case of four blur kernels and this reduced to the order of  $10^{-9}$  when there were 12 or 16 blur kernels. Therefore, we can empirically conclude that a reliable TSF can be estimated from about 12 blur kernels whose locations are spread across the image.

#### 4 Depth Estimation

In this section, we seek a solution for the inverse problem of determining the relative depth given the images  $f$  and  $g$ , and the reference TSF  $\omega_o$ . We model the depth map as an MRF and obtain the MAP estimate using the max-product BP algorithm (Felzenszwalb and Huttenlocher 2006).

##### 4.1 Depth Estimation Using BP

For a 3D scene, the observation  $g$  is modeled by space-variant blurring of the original image  $f$  with the PSF  $h(i, j; m, n)$ . With the knowledge of the TSF  $\omega_o$ , the PSF at any image point can be obtained from its relative depth  $k(i, j)$  using Eq. (11). Our objective is to estimate  $k(i, j)$  at each pixel based on the images  $f$  and  $g$ , and the TSF  $\omega_o$  (estimated at depth  $d_o$ ). We formulate a MAP-MRF framework to solve for the relative depth  $k$  at every pixel. To obtain the MAP estimate, we follow an implementation of the loopy BP algorithm proposed in (Felzenszwalb and Huttenlocher 2006). An advantage of using BP over gradient-based approaches is that we avoid the evaluation of derivatives which is quite tedious especially in



**Fig. 2** Synthetic experiment. (a) Reference image. (b) Blurred observation. (c) True depth map. (d) Estimated depth map. *Third row* blur kernels generated using the true TSF and true relative depth values. *Fourth row* blur kernels generated from the estimated TSF and relative depth values

the presence of space-variant PSF due to camera rotations (Sorel and Flusser 2008). The BP-based algorithm used in this work is iterative. The MAP estimate improves after each iteration until convergence. Also, we incorporate regularization by defining a smoothness cost which is based on the standard four neighbors of a pixel and can also preserve discontinuities.

For our depth estimation problem, MAP estimation is to be performed over a grid-graph, with pixel locations as the nodes. The max-product rule works by passing messages between nodes. Each message is an  $L$ -dimensional vector, where  $L$  is the number of labels that a node can take. The message  $m_{pq}^t(\mathbf{f}_q)$  at time  $t$  to a node  $q$  from its neighbouring node  $p$  of the graph is given by

$$m_{pq}^t(\mathbf{f}_q) = \min_{\mathbf{f}_p} \left( D_p(\mathbf{f}_p) + V(\mathbf{f}_p, \mathbf{f}_q) + \sum_{s \in \mathcal{N}(p)|q} m_{sp}^{t-1}(\mathbf{f}_p) \right) \quad (17)$$

where  $D_p(\mathbf{f}_p)$  is the cost for assigning a label  $\mathbf{f}_p$  at node  $p$  (data cost),  $V(\mathbf{f}_p, \mathbf{f}_q)$  is the smoothness cost between the neighbouring nodes  $p$  and  $q$ , and  $s \in \mathcal{N}(p)|q$  denotes the set of nodes in the neighbourhood of  $p$ , not including the node  $q$ . This message passing is iterated for each node until convergence whence the beliefs are computed as

$$b_q(\mathbf{f}_q) = D_q(\mathbf{f}_q) + \sum_{p \in \mathcal{N}(q)} m_{pq}(\mathbf{f}_q) \quad (18)$$

The belief  $b_q(\mathbf{f}_q)$  at each node  $q$  is an  $L$ -dimensional vector. The MAP solution for the label at  $q$  is that  $\mathbf{f}_q$  which minimizes  $b_q(\mathbf{f}_q)$ .

To define the data cost, we locally approximate the blurring model of Eq. (6) as

$$g(i, j) = \sum_{m,n} f(i - m, j - n) h(i, j; m, n) \quad (19)$$

In this model, the PSF in the local neighborhood of  $(i, j)$  is assumed to be the same as the PSF at  $(i, j)$ . The blurred image intensity  $g(i, j)$  is dependent on the reference image  $f$  and the blur kernel  $h(i, j; m, n)$ . Suppose that the value of relative depth at the pixel  $(i, j)$  is equal to  $k$ . Due to the parallax effect, the translations of the reference TSF  $\omega_o$  will get scaled by  $k$  resulting in a blur kernel denoted by  $h_k(i, j; m, n)$  (according to Eq. (11)). We define the data cost of assigning a label  $k$  at a pixel  $(i, j)$  as

$$D_{(i,j)}(k) = |g(i, j) - g_k(i, j)| \quad (20)$$

where  $g_k(i, j) = \sum_{m,n} f(i - m, j - n) h_k(i, j; m, n)$ . Although the model in Eq. (19) is not strictly equivalent to the space-variant blurring model, such an approximation simplifies the definition of the data cost (Bhavasar and Rajagopalan 2012).

The smoothness cost is defined to penalize the difference in the labels of neighboring nodes. To allow for discontinuities, the cost function should take a constant value when the difference becomes large. A commonly used prior is the truncated linear model (Felzenszwalb and Huttenlocher 2006) which is defined as

$$V(\mathbf{f}_p, \mathbf{f}_q) = \min (\mu_p |\mathbf{f}_p - \mathbf{f}_q|, \lambda_{\text{th}}) \quad (21)$$

where  $\mu_p$  is a weighting parameter and the threshold  $\lambda_{\text{th}}$  determines when the cost stops increasing. On convergence, the depth value at each pixel is obtained by calculating the label that minimizes the belief vector (Eq. (18)). The steps involved in our methodology are summarized below:

#### Algorithm Depth from Motion Blur

**Input:** Unblurred ( $f$ ) and blurred image ( $g$ )

**Output:** TSF  $\omega_o$  and relative depth map  $k(\cdot, \cdot)$

1. Estimate PSFs  $h_{p_1} h_{p_2} \dots h_{p_{N_p}}$  from image patches of  $f$  and  $g$  cropped at  $N_p$  random locations

##### 2. TSF estimation:

Initialize the relative depth values  $k_{p_1} k_{p_2} \dots k_{p_{N_p}}$  to unity  
**repeat**

Generate matrix  $M$  from the relative depth values and the locations of blur kernels

Estimate  $\omega_o$  by solving Eq. (16)

Update the relative depths  $k_{p_1} k_{p_2} \dots k_{p_{N_p}}$  from the estimate of  $\omega_o$   
**until** the estimates of  $\omega_o$  and  $k_{p_1} k_{p_2} \dots k_{p_{N_p}}$  converge

##### 3. MAP estimation of (dense) relative depth using BP:

**repeat**

**for** every pixel  $(i, j)$

For a depth label  $k$ , evaluate the blur kernel  $h_k(i, j; m, n)$  from the reference TSF  $\omega_o$

Update the message according to Eq. (17) from the data cost (Eq.

(20)) and smoothness cost (21))

**until** convergence

Evaluate the MAP solution for the relative depth at each pixel (Eq. (18))

## 5 Experimental Results

We evaluated the proposed TSF and depth estimation techniques on synthetic as well as real data. Through synthetic experiments, we study the performance of our TSF estimation and depth estimation techniques for different kinds of textures in the presence of noise, and under modeling errors. All our real images were captured by a digital camera (Canon EOS-60D). For the purpose of implementation, following (Whyte et al. 2010; Gupta et al. 2010), we discretize the transformation space in such a manner that the difference in the displacements of an image point due to two different transformations from the discrete set  $\mathbf{T}$  is at least one pixel. In our motion model, we primarily considered only in-plane translation and rotations. Although, this model will suffice in most scenarios, a camera shake can lead to a more general motion. If we were to additionally consider axial translation and out-of-plane rotation of the camera, the transformation space  $\mathbf{T}$  will be a 6D space (corresponding to the six degrees of freedom for camera motion). In such a case, the value of  $N_T$  (the number of columns of  $M$ ) would be very large rendering TSF estimation severely under-determined. Hence, without loss of generality, instead of considering all the transformations together, we separately consider the following three scenarios: (a) 2D translations and rotations, (b) camera translations along all three directions, and (c) out-of-plane rotation (about  $Y$  axis) and translation along one direction ( $X$  axis).

For a scene with constant depth, blur due to out-of-plane camera rotation can be approximately modeled by translational motion and vice versa (Whyte et al. 2012; Gupta et al. 2010). When the camera undergoes both out-of-plane rotations and translations (parallel to the image plane), the estimation of motion parameters is known to be inherently difficult due to the *translation–rotation ambiguity* (Dannilidis and Nagel 1993). Consequently, the recovery of TSF becomes difficult when the scene depth is constant. On a synthetic experiment, we show that TSF estimation is ill-posed for a constant depth scenario when there is out-of-plane rotation along with translation. However, for a 3D scene, we are able to resolve the *translation–rotation ambiguity* and recover TSF to a reasonable accuracy by using blur kernels estimated at random locations that have different depth values (demonstrated later in this section).

As with any blur-based depth estimation technique, the scene cannot be textureless (Favaro and Soatto 2006). As we are not aware of any technique that accounts for in-plane

camera rotation while estimating depth from motion blur, we are not in a position to compare the experimental results of our method with other techniques in the literature. However, we do demonstrate that the 2D translation model for the camera motion which was followed in (Sorel and Flusser 2008; Paramanand and Rajagopalan 2012) is inadequate for depth estimation for our scenario.

### 5.1 In-plane Rotation and Translation

We initially tested our depth estimation method on a synthetic example. We used a textured image from (Brodatz 1966) as the reference unblurred observation of the scene (Fig. 2a). The relative depth map was assumed to vary between 1 and 2.3 in the form of a ramp as shown in Fig. 2c. The TSF used to generate the blurred observation was the same as the one used in our earlier experiment on TSF estimation (corresponding to Fig. 1 in Sect. 3). The blurred observation generated from the TSF and the depth map is shown in Fig. 2b. The objective was to estimate the TSF and the depth map from the two images. Based on our discussion in Sect. 3, to estimate the TSF, we randomly selected twelve different image points across the image. The blur kernels at these points were obtained using image patches (marked in Fig. 2a, b) cropped from the reference and blurred observations. Using the proposed alternate minimization method, we recovered the TSF. The TSF estimation error  $e_{TSF}$  reduced with each iteration; from 0.0040 at the end of first iteration to 0.0016 after fifth iteration (when convergence was attained). The blur kernels generated from the true TSF (along with the true depth values), and those generated from the estimated TSF and the estimated relative depth values were quite close to each other as shown in the third and fourth rows of Fig. 2, respectively. This indicates that the result of our TSF estimation is quite accurate. From the estimated TSF and the observations, the depth map was obtained by the proposed BP-based method. Due to the translation–depth ambiguity, our depth estimate differed from the true depth map by a scale factor. To evaluate the error, we multiplied our depth estimate with the ratio of the means of the true depth map and the estimated depth map. The scaled version of our estimate (shown in Fig. 2d) is close to the true depth map (Fig. 2c). We evaluated the error

$$\text{in estimation (}ERR\text{)} \text{ as } ERR = \sqrt{\text{Avg} \left[ \left( \frac{\hat{k}}{k} - 1 \right)^2 \right]}, \text{ where}$$

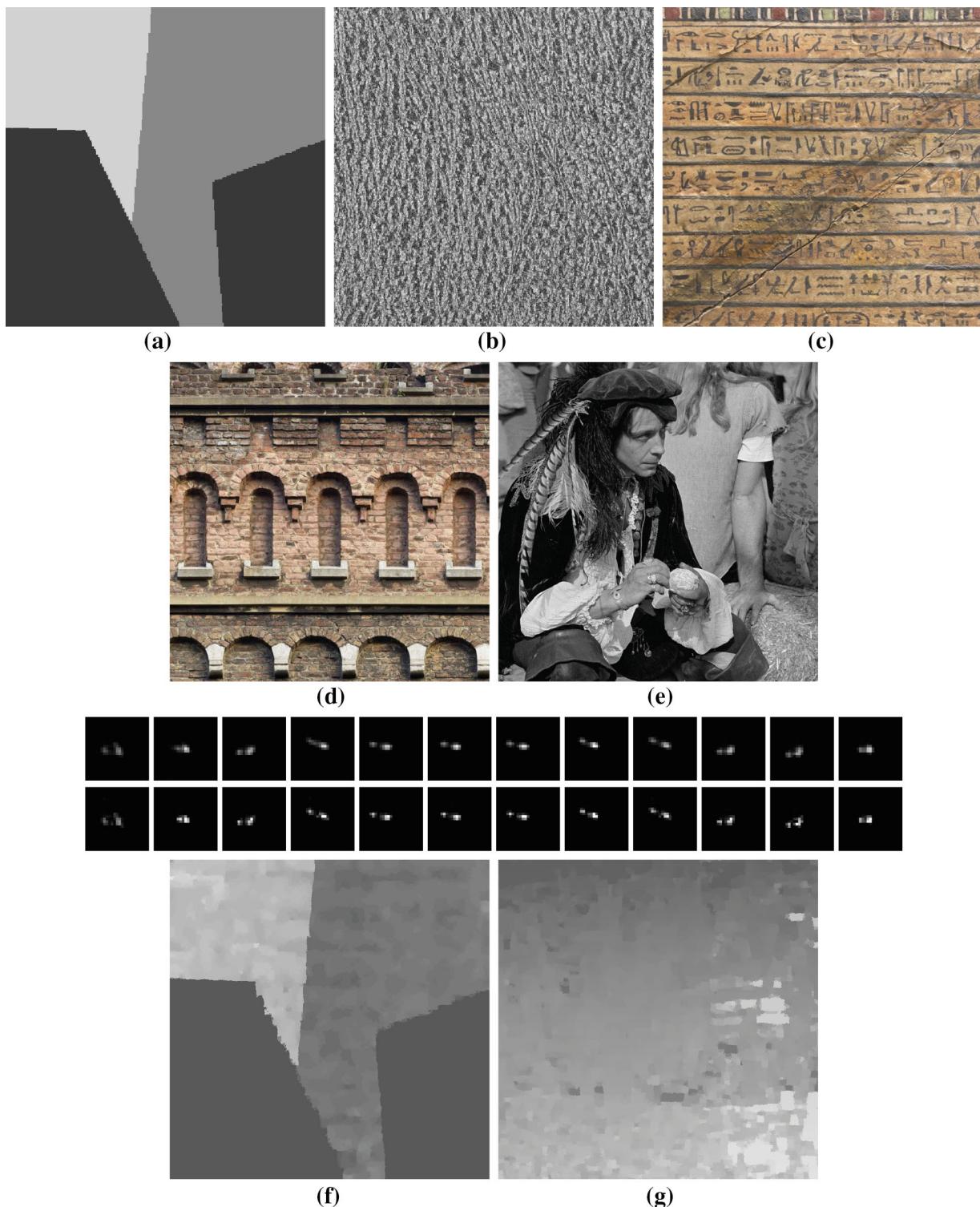
$\text{Avg} [\cdot]$  denotes averaging, and  $\hat{k}$  and  $k$  denote the estimated and actual (relative) depth values, respectively. The  $ERR$  was found to be only 2.8 %. Since the scene was highly textured, the depth estimate is accurate.

We next performed synthetic experiments to test our depth estimation and TSF estimation techniques by considering three different levels of textures for the reference image. We repeated our experiments on two different depth maps (shown

in Figs. 2c, 3a) and two different TSFs (which depicted realistic camera shakes) under the influence of noise of standard deviation 10 (added to the reference unblurred image). We considered the images in Figs. 2a, 3b as highly textured. While Fig. 3c, d were regarded as medium level textures, Figs. 1a and 3e were considered as low level textures. In all our experiments, to estimate the TSF, we randomly selected twelve different image points across the image and estimated blur kernels using image patches from the unblurred-blurred image pair. It is to be noted that there can be slight inaccuracies in the PSF estimates due to lack of texture, noise and when the randomly selected center of the image patch lies near a depth discontinuity (in the case of the depth map in Fig. 3a). For instance, in the case of medium level texture, and when noise standard deviation was 10, the PSFs were estimated as shown in the third row of Fig. 3. The corresponding ground truth PSFs are given in the fourth row of Fig. 3. The estimated PSFs appear to be slightly ‘spread out’ as compared to the true PSFs.

From the proposed alternate minimization scheme, we recovered the TSF. In all of the experiments, the algorithm converged within 10 iterations. The cost function in Eq. (16) as well as the error with respect to the true TSF reduced with every iteration (till convergence). The average error in TSF estimation was 0.0022 for high texture, 0.0032 for medium texture and 0.0038 for low texture. This indicates that as the level of texture reduces, the robustness of TSF estimation to noise reduces. From the estimated TSF and the observations, the depth map was obtained by the proposed BP-based method. The average value of  $ERR$  was 5.6 % for high texture, 8.4 % for medium texture, and 9.3 % for low texture. Figure 3f shows the depth map when medium level texture was used (true depth map is as in Fig. 3a). When the texture content was low, the depth map was obtained as shown in Fig. 3g (true depth map as depicted in Fig. 2c). Figure 3f, g indicate that except for a few artifacts, the estimated depth maps appear similar to the ground truth depth maps.

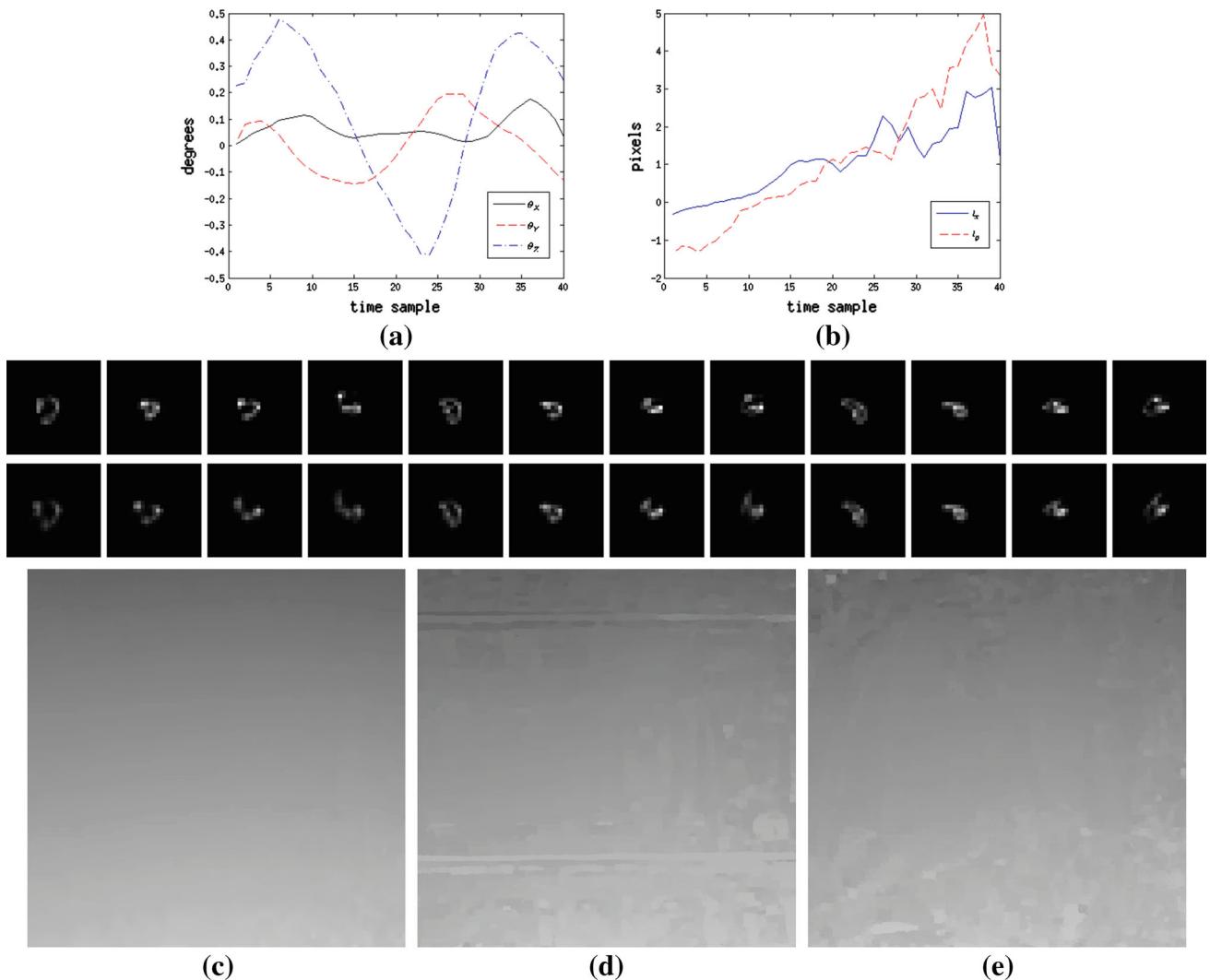
In our next synthetic experiment, we tested our algorithm in a scenario wherein, in addition to in-plane rotations and translations, the camera motion included small amounts of out-of-plane rotations during blurring. This experiment was performed thrice by considering the reference image to be of high texture (Fig. 2a), medium texture (Fig. 3d) and low texture (Fig. 3e). The depth map of the scene was assumed to be a ramp as shown in Fig. 2c. In order to generate the blurred observations, we assumed that the camera rotations (about all the three axes) and translations (at the reference depth) during the exposure were as shown in Figs. 4a, b, respectively. While the in-plane rotation angle  $\theta_Z$  ranged between  $-0.5^\circ$  and  $0.5^\circ$ , the angles  $\theta_X$  and  $\theta_Y$  were within a magnitude of  $0.2^\circ$ . A typical value of  $v = 900$  was chosen as the camera focal length. Blurred observations were obtained from the space-variant PSF that was generated (at every pixel) from



**Fig. 3** Synthetic experiment—effect of texture. **(a)** Ground truth depth map. **(b)** High texture. **(c), (d)** Medium texture. **(e)** Low texture. *Third row* esimated PSFs. *Fourth row* true PSFs. Estimated depth map for noise level of 10 for **(f)** medium texture and **g** low texture

the camera motion and the depth map. From image patches, we estimated blur kernels at different locations. The TSF was estimated from these blur kernels using the proposed

alternate minimization scheme. The error in TSF estimation was 0.0032 for high texture, 0.0035 for medium texture and 0.0036 for low texture. For the case of high texture, the true

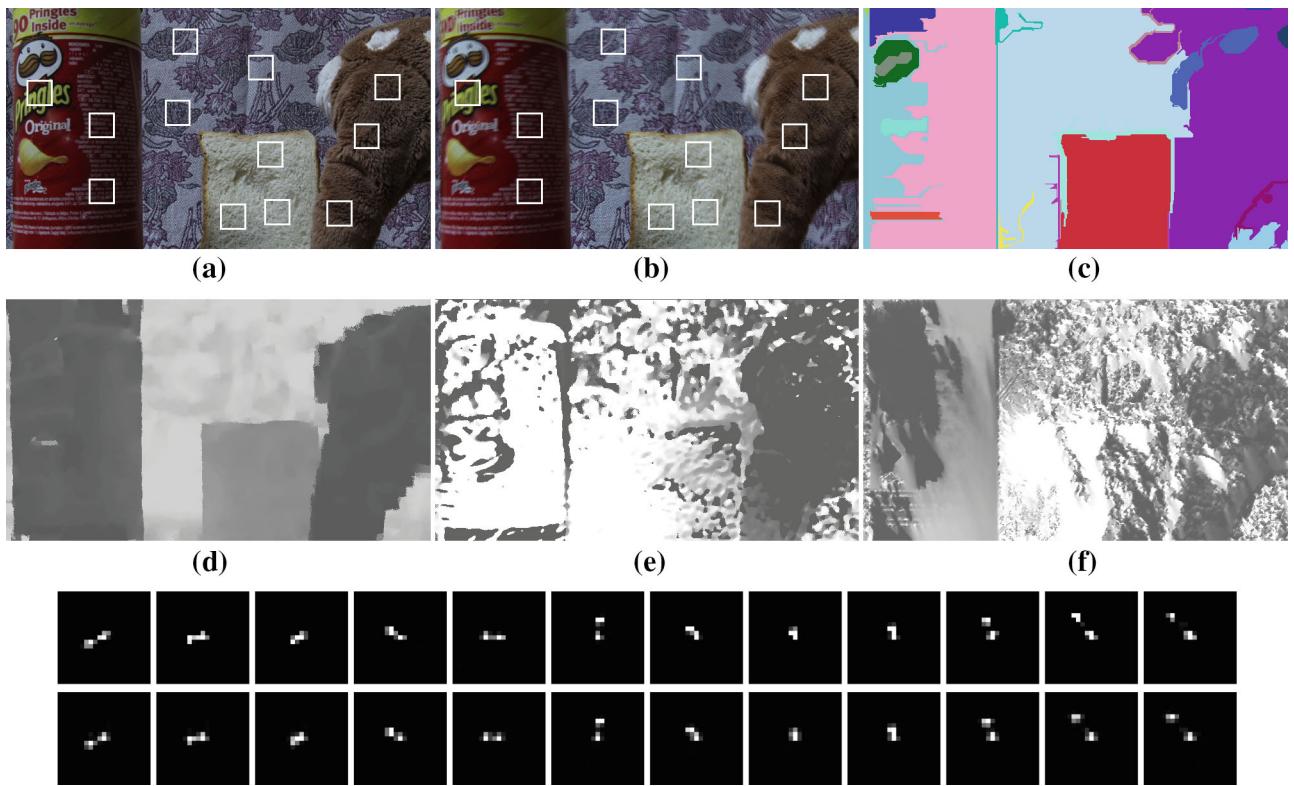


**Fig. 4** Synthetic experiment—robustness to out-of-plane rotation. **(a)** Camera rotations and **(b)** translations as a function of time. *Second row* blur kernels generated from the known camera motion. *Third row*

blur kernels generated from the TSF estimated by the proposed method. Estimated depth map for **(c)** high texture, **(d)** medium texture, and **(e)** low texture

blur kernels (generated at 12 different locations) are shown in the second row of Fig. 4 and the PSFs obtained from the estimated TSF are shown in the third row of Fig. 4. Although, the estimated TSF was 3D and the camera motion was 5D, the blur kernels generated from the known camera motion and depth values were quite close to those generated from the estimated TSF and estimated depth values. From the estimated TSF, we arrive at the depth maps as shown in Fig. 4c (high texture), Fig. 4d (medium texture) and Fig. 4e (low texture). Blurring due to out-of-plane rotation does not vary with respect to depth. But, in this experiment, since out-of-plane rotation gets treated as in-plane translation, the variation of blur with respect to depth is not modeled correctly. Despite this, we see that the depth estimates are close to the true depth values. The  $ERR$  value was 8.1 % for high texture, 9.1 % for medium texture and 9.3 % for low texture.

In the first real experiment, the scene consisted of four objects kept at known distances from the camera as shown in Fig. 5a. The reference image of the scene (Fig. 5a) was captured with an exposure time of 1/3 s at ISO 1000. The blurred observation shown in Fig. 5b was captured by gently shaking the handheld camera while taking care to avoid large camera displacements. The exposure time was 1.6 s at ISO 250. The TSF was estimated using our alternate minimization approach from blur kernels at different image points selected at random. For patch selection, in order to avoid a point at the boundary of two depth layers from being selected as the center of a patch, we subjected the reference image to segmentation using the algorithm in (Felzenszwalb and Huttenlocher 2004). The result of segmentation is shown in Fig. 5c. It is quite unlikely to have a sharp depth discontinuity within a segment. Hence, out of the randomly



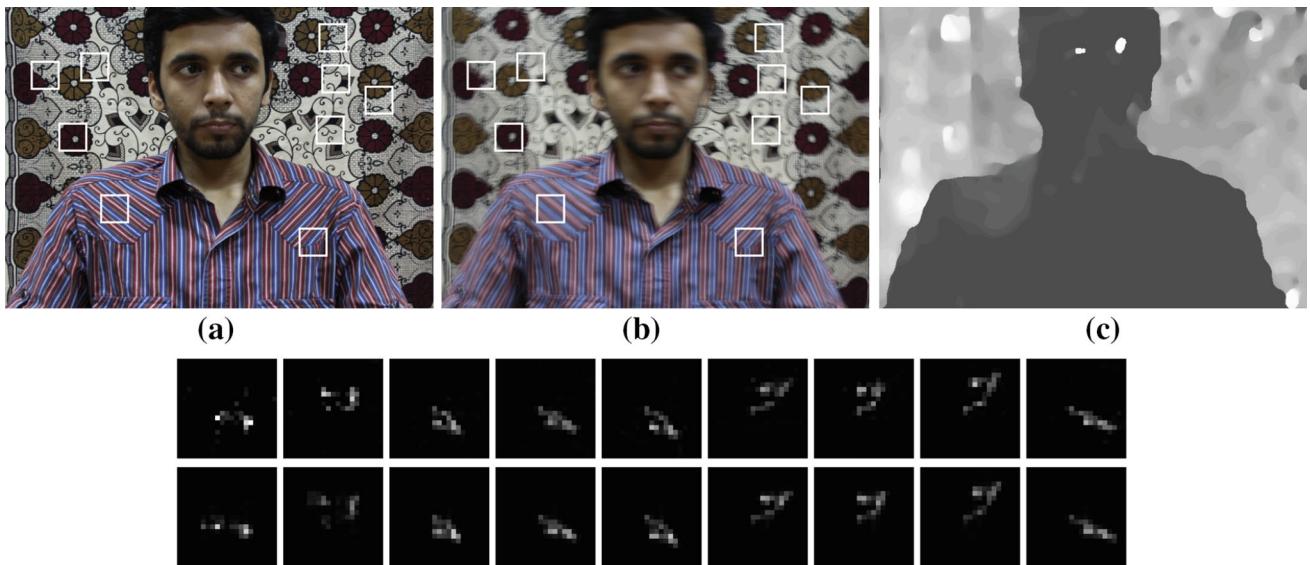
**Fig. 5** Real experiment. **(a)** Reference image. **(b)** Blurred observation. **(c)** Segmented reference image. Estimated depth map **(d)** by the proposed method, **(e)** by considering only in-plane translations, and **(f)**

using the method in (Paramanand and Rajagopalan 2012). *Third row* blur kernels estimated from the patches. *Fourth row* blur kernels generated from the estimated TSF and relative depth values

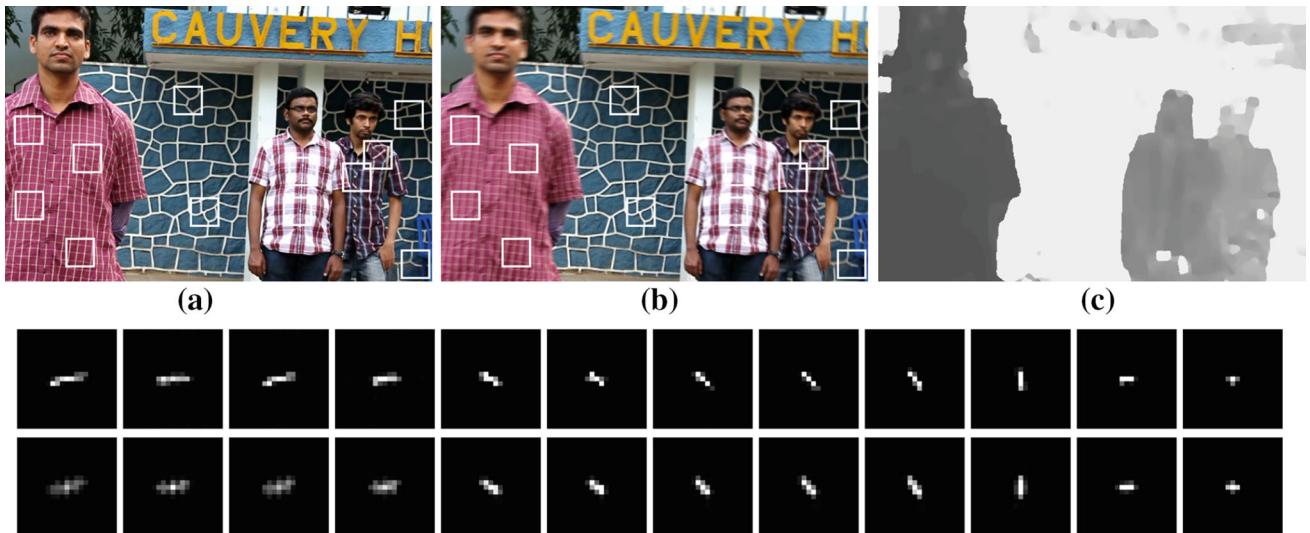
selected patch centers, those that were close to the boundary of a segment were dropped. The selected patches are marked in Fig. 5a, b. The estimated blur kernels are shown in the third row of Fig. 5. The TSF was estimated from these kernels. Along the  $\theta$  axis, the values of the TSF were concentrated between  $-0.25^\circ$  and  $0.75^\circ$ . At these angles, the TSF had significant values of translations (both  $t_x$  and  $t_y$ ). This indicates that the camera underwent both in-plane rotations and 2D translations. The blur kernels derived from the estimated TSF (Fig. 5, fourth row) closely match the observed blur kernels (Fig. 5, third row). From the proposed method, we obtain the relative depth map as shown in Fig. 5d. In Fig. 5d, we see that the depth map correctly captures the scene structure; the region corresponding to the bread slice is assigned a higher relative depth than the two objects that were closer to the camera, and all the background points are assigned a high depth value. We obtained an estimate of the absolute depth using the known distance between the camera and one of the objects of the scene. The depth of the scene ranged between 23 and 55 cm, and our depth estimates were within 5 % in error. For the sake of comparison, we also performed depth estimation on the observations by considering the camera motion to be composed of only in-plane translations (as done in Sorel and Flusser 2008; Paramanand and Rajagopalan 2012). One of the PSFs that was used to estimate

the TSF was considered as the reference. According to the camera motion considered, the PSF at any other point in the image will be a scaled version of the reference PSF wherein, the scale factor denotes the relative depth (Sorel and Flusser 2008). The scale factors of the reference PSF were estimated at every pixel using a BP-based algorithm. The depth map obtained is as shown in Fig. 5e. By using another estimated blur kernel (other than the one used to arrive at the depth map of Fig. 5e), as the reference and applying the depth estimation technique in (Paramanand and Rajagopalan 2012), we obtain the depth map shown in Fig. 5f. In Fig. 5e, f, we see that the depth estimates are grossly incorrect implying that in-plane rotations cannot be ignored.

In our next experiment, we considered a scene of a person standing against a background (Fig. 6a). The blurred observations shown in Fig. 6b was captured from a handheld camera. The blur kernels (determined from the image patches) shown in the second row of Fig. 6 were used for TSF estimation. The blur kernels derived from the estimated TSF (Fig. 6, third row) closely match the observed blur kernels (Fig. 6, second row). Using the proposed BP algorithm, we obtain the depth estimate shown in Fig. 6c. Except for small isolated artifacts, the depth map correctly segments the foreground and background depth layers although this scene has considerably less texture.



**Fig. 6** Real experiment. (a) Reference image. (b) Blurred observation. (c) Estimated depth map. *Second row* blur kernels determined from the image patches. *Third row* blur kernels generated from the TSF estimated by the proposed method



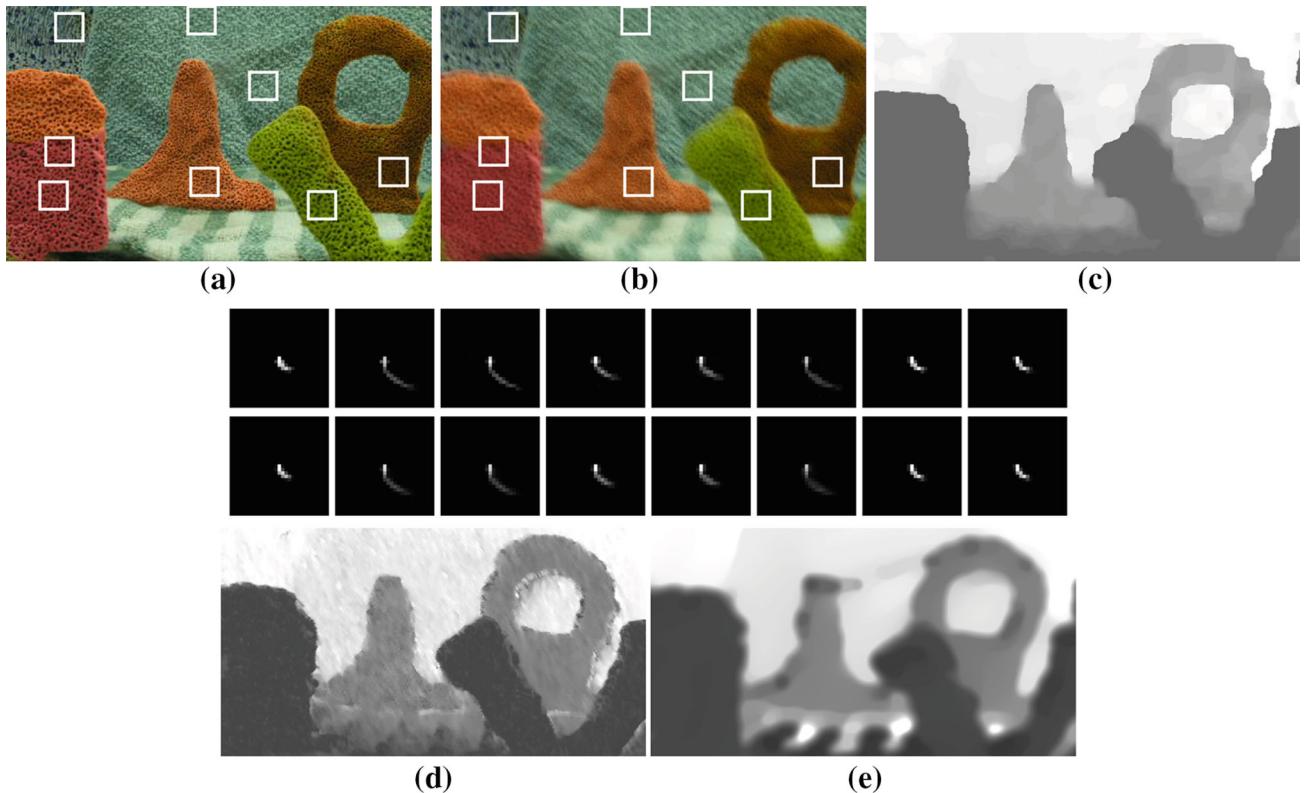
**Fig. 7** Real experiment. (a) Reference image. (b) Blurred observation. (c) Estimated depth map. *Second row* blur kernels determined from the image patches. *Third row* blur kernels generated from the TSF estimated by the proposed method

We also tested our method on images of an outdoor scene. The reference image (Fig. 7a) shows three people standing at different distances from the camera with a wall in the background. While the distances between the camera and the people were 2.3, 4 and 4.5 m, respectively, the background was at 9 m. The blurred observation is shown in Fig. 7b. While capturing the observations, the camera was placed on a tripod for support.

The TSF was estimated using blur kernels that were determined from the image patches (second row of Fig. 7). The blur kernels generated from the estimated TSF and relative depths (Fig. 7, third row) are quite similar to the observed blur kernels (Fig. 7, second row). We arrived at the depth

map from the proposed scheme as shown in Fig. 7c, wherein the intensity value correctly reflects the scene structure. The error in the depth estimates was within 7% of the measured values.

While capturing the blurred observation, if the camera motion is purely rotational and there are no translations, the blur does not vary with respect to changes in depth. Hence, depth recovery is not possible in such a scenario. In a hand-held camera, the variation of blur due to depth (due to translational motion) is readily observed in indoor scenes (where the objects are within a few meters from the camera) (Joshi et al. 2010). However, in outdoor scenes, in order to use motion blur as a depth cue, the translational motion of the cam-



**Fig. 8** Real experiment— inplane translations. (a) Reference image. (b) Blurred observation. *Second row* blur kernels determined from patches. *Third row* blur kernels generated from the estimated TSF. Esti-

mated depth map using (c) the proposed method, (d) [Paramanand and Rajagopalan \(2012\)](#), and (e) [Sorel and Flusser \(2008\)](#)

era should be significantly more (as compared to indoors). Hence, in our outdoor experiment (Fig. 7), we used the support of a tripod in order to reduce the effect of out-of-plane rotations in the camera shake.

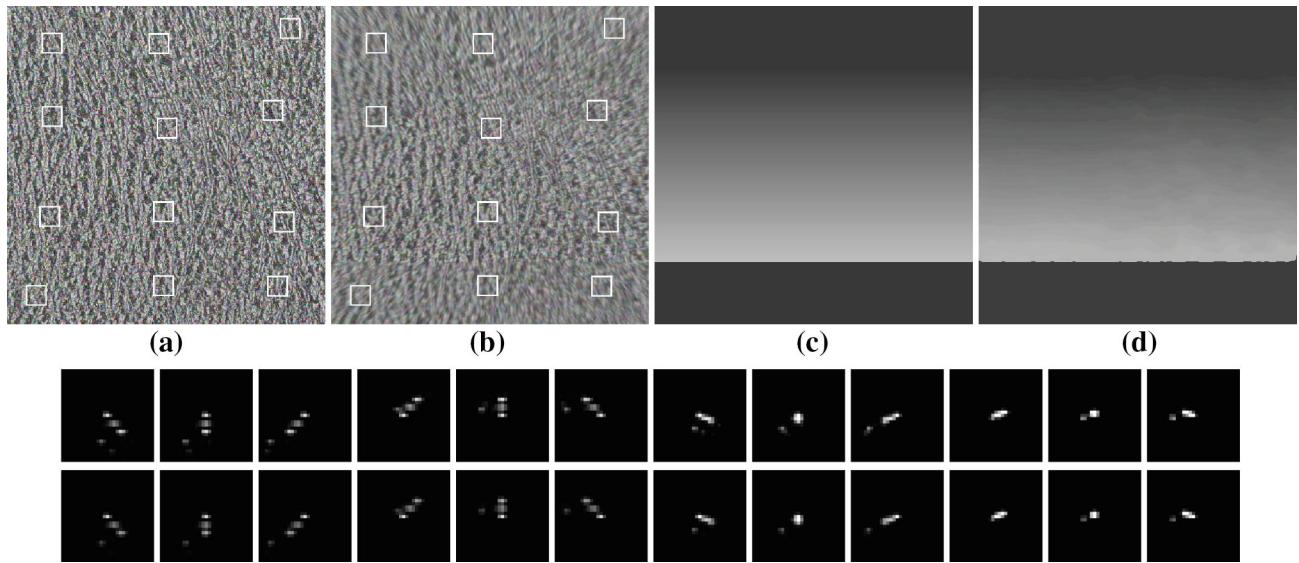
As a special case of our method, we applied our algorithm when the camera motion was restricted to only in-plane translations. We used an unblurred-blurred image pair (shown in Fig. 8a, b) from the data set of a very recent work in ([Paramanand and Rajagopalan 2012](#)) (wherein a translating stage is used to capture images). In the estimated TSF, transformations with nonzero angles of rotation had negligible weights (as expected). While our estimate of the depth map is shown in Fig. 8c, depth estimates from the techniques in ([Paramanand and Rajagopalan 2012](#)) and ([Sorel and Flusser 2008](#)) are shown in Fig. 8d, e, respectively. All the three depth maps are comparable in terms of intensity value. It must be noted that in the depth map of Fig. 8e, the estimates go wrong at a few places (bright spots that appear towards the bottom of the image). Such artifacts are not present in the estimate from the proposed method.

## 5.2 3D Translations

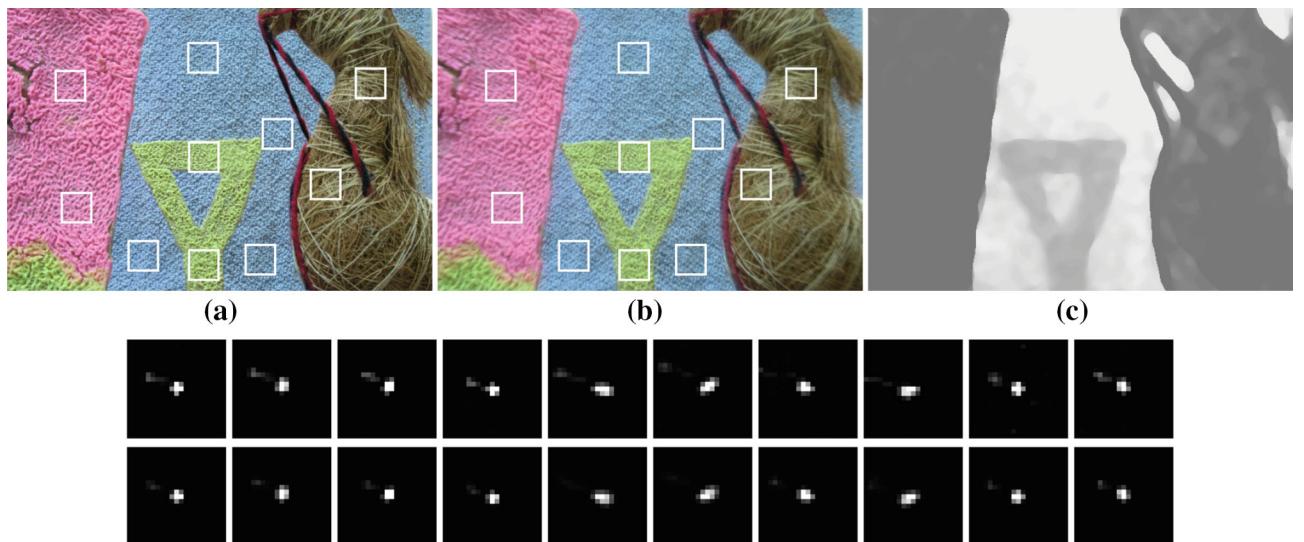
We tested our method for the case of camera translations along all three directions on a synthetic example. The image

shown in Fig. 9a was used as the reference observation. The relative depth map of the scene was assumed to be as shown in Fig. 9c. Its values ranged between 1 and 2. The transformation space  $T$  was parameterized by  $t_x$ ,  $t_y$ , and  $\xi$  axes. Along the  $\xi$  axis, the TSF took significant values between 1 and 1.02, and it was negligible at other scales. We generated the blurred observation from the reference image according to the reference TSF and the depth map. Patches at different image points were cropped from the observations (shown in Fig. 9a, b) to estimate blur kernels. The TSF was determined from the estimated blur kernels using our alternate minimization scheme. While the blur kernels generated by the true TSF and the known relative depth values at the image points are shown in the second row of Fig. 9, those generated from the estimated TSF and estimated depth values are plotted in the third row of Fig. 9. Since, the two sets of blur kernels are close to one-another, we can infer that our TSF estimation is quite accurate. From the TSF estimate, we recovered the depth map using the proposed scheme. Our estimate of the depth map shown in Fig. 9d (after scaling) closely mimics the true depth map (Fig. 9c). The error  $ERR$  was found to be only 3.3 %.

In our real experiment, to capture the observations, we placed the camera on a translating stage. The scene shown in Fig. 10a consisted of objects kept at calibrated distances



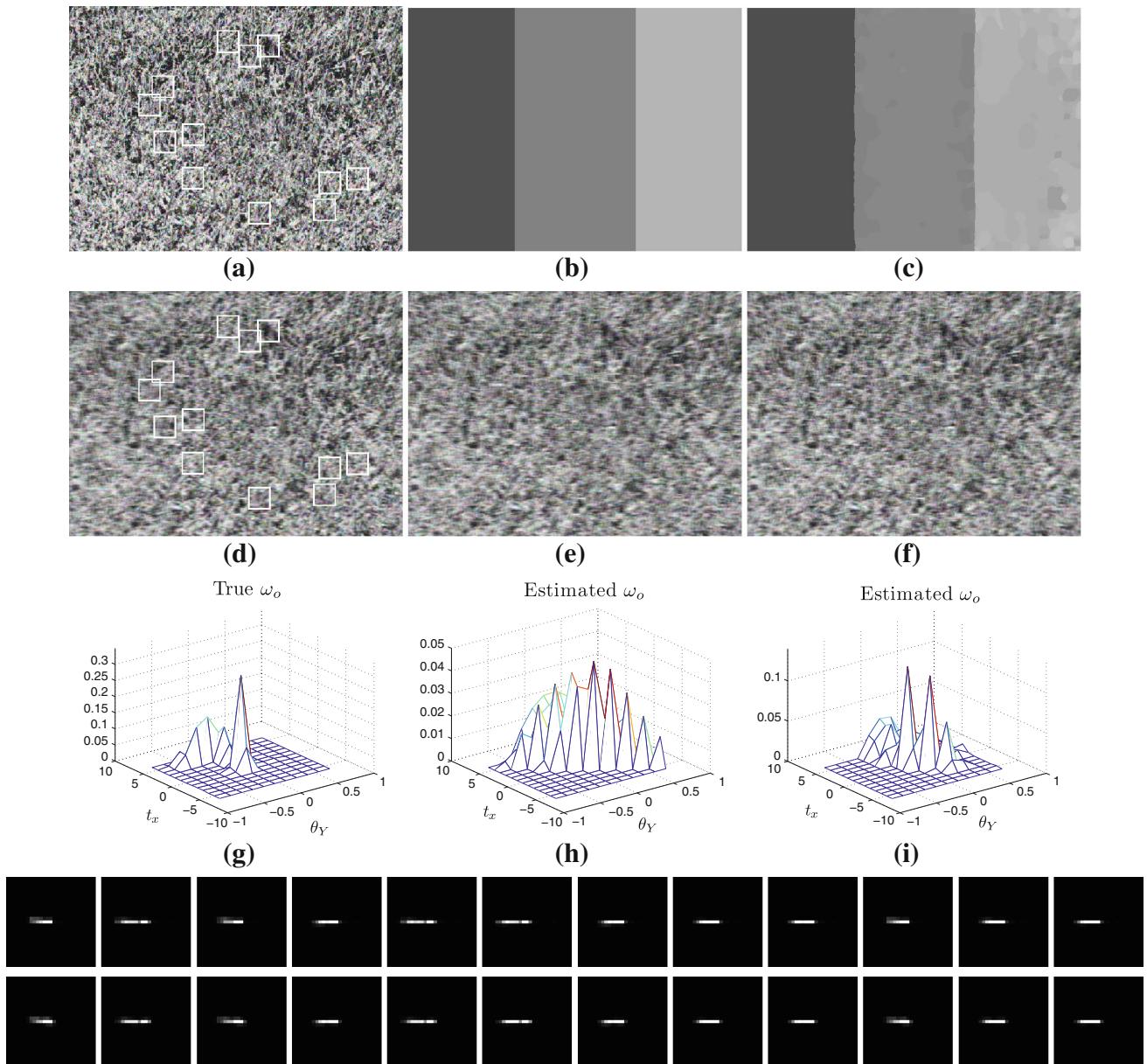
**Fig. 9** Synthetic experiment—3D translations. (a) Reference image. (b) Blurred observation. (c) True depth map. (d) Estimated depth map. Second row blur kernels generated using the true TSF and true relative depth values. Third row blur kernels generated from the estimated TSF and relative depth values



**Fig. 10** Real experiment—3D translations. (a) Reference image. (b) Blurred observation. (c) Estimated depth map. Second row blur kernels determined from the image patches. Third row blur kernels generated from the TSF estimated by the proposed method (Color figure online)

from the camera. The horse and the pink colored model were at the same distance from the camera. The yellow triangular model was kept in between these two objects and the background. The reference image of the scene (Fig. 10a) was captured with a still camera. While capturing the blurred observation (shown in Fig. 10b), we arbitrarily translated the camera along all three directions during exposure. Using image patches, we estimated blur kernels at different image locations (shown in second row of Fig. 10). With these blur kernels, the TSF was estimated by the proposed method. As to be expected, the blur kernels generated using the estimated TSF and the relative

depth values at the locations of the patches (third row of Fig. 10) closely match the observed blur kernels (second row of Fig. 10). The values of the TSF were concentrated between 0.97 and 1 along the scale ( $\xi$ ) axis, and between  $-3$  and  $11$  along  $t_x$  axis indicating that the camera motion was dominant along axial and horizontal directions. Based on the estimated TSF and the observations, we obtained the depth map shown in Fig. 10c which captures the 3D structure of the scene quite well. The depth of the scene ranged from  $20.5$  cm (pink model and the horse) to  $45$  cm (background). The error in our depth estimates was within 6%.



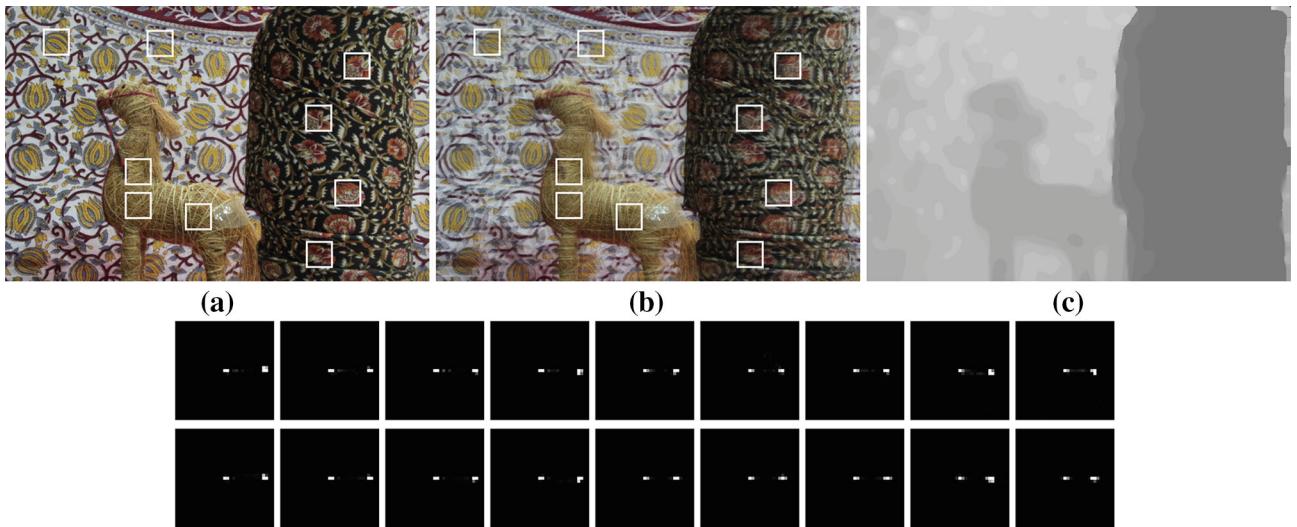
**Fig. 11** Synthetic experiment—presence of translation–rotation ambiguity. (a) Reference image. (b) True depth map. (c) Estimated depth map. (d) Blurred observation. Observation generated by (e) the true TSF, and (f) the estimated TSF (shown in (h)). (g) True TSF. Estimated

TSF using (h) four PSFs from a single depth layer, and (i) blur kernels across the image. *Fourth row* true blur kernels. *Fifth row* blur kernels generated from the estimated TSF

### 5.3 Out-of-plane Rotation and Translation

We tested our method for the scenario wherein the camera underwent out-of-plane rotation and translation together. Initially, we demonstrate the effect of translation–rotation ambiguity and then show that our alternate minimization scheme can resolve this ambiguity to some extent. We considered the camera motion to be composed of out-of-plane rotation about  $Y$  axis (yaw rotation) and translation along  $X$  axis. We specifically chose such a camera motion because the displacement

of an image point induced due to rotation about  $Y$  axis and translation along  $X$  axis are similar leading to translation–rotation ambiguity. The transformation space was parameterized by  $t_x$ , and  $\theta_Y$  (the rotation about  $Y$  axis). The value of  $t_x$  varied from  $-6$  to  $6$  in steps of one pixel while  $\theta_Y$  ranged between  $-0.8$  and  $0.7$  in steps of  $0.15^\circ$ . The value of the focal length  $v$  was assumed to be 400 pixels. The ground truth TSF was assumed to be as shown in Fig. 11g. The reference unblurred image and the scene depth map are shown in Figs. 11a, b, respectively. The depth map contains three



**Fig. 12** Real experiment—presence of translation–rotation ambiguity. (a) Reference image. (b) Blurred observation. (c) Estimated depth map. Second row blur kernels determined from the image patches. Third row blur kernels generated from the TSF estimated by the proposed method

different layers of constant depth. We generated the blurred image (Fig. 11d) from the TSF according to the depth map (by scaling the translation component when generating blur kernels). Initially, we selected four points from one of the depth layers and estimated local PSFs. From these blur kernels, the TSF was estimated as shown in Fig. 11h. It is interesting to note that although the estimated TSF is significantly different from the original TSF of Fig. 11g, the blur kernels generated from the estimated TSF closely matched the blur kernels generated from the original TSF (rms 0.0015). Furthermore, we blurred the original image (by assuming the scene depth to be unity at all points) using the true TSF (Fig. 11g) and the estimated TSF (Fig. 11h), to get the observations as shown in Fig. 11e, f, respectively. The rms error between these two observations was only 2.74. This shows that for the case of yaw rotation and translation along X axis, and when the depth is constant, two different TSFs can result in very similar blurring effects.

We next estimated the TSF using blur kernels determined at twelve different patches from the unblurred-blurred image pair of the 3D scene (Fig. 11a, d). The TSF obtained from the alternate minimization scheme is plotted in Fig. 11i. The blur kernels generated from the estimated TSF and relative depths (fifth row of Fig. 11) were found to be close to the true blur kernels (fourth row of Fig. 11). Although the estimated TSF (Fig. 11i) does not exactly mimic the true TSF (Fig. 11g), the two TSFs are quite close. Also, the locations where the nonzero weights occur are quite similar in the two TSFs. This shows that we are able to resolve the translation–rotation ambiguity to a reasonable extent for the 3D scene. From the estimated TSF, the depth map was recovered as shown in Fig. 11c which matches with the true depth map of Fig. 11b. The value of ERR was 9 %.

Finally, we performed a real experiment for the case of simultaneous out-of-plane rotation and translation. The reference image was captured when the camera was still. To capture the blurred image, the camera was subject to panning. Since the camera motion was restricted, we considered the TSF to be a 2D function denoting out-of-plane rotation about Y axis and translation along x axis. The scene consisted of a horse model kept between two planar objects. The reference image and the blurred observation of the scene are shown in Fig. 12a, b, respectively. The local blur kernels determined from image patches are shown in a grid of  $33 \times 33$  in the second row of Fig. 12. Based on the TSF estimated from alternate minimization, the blur kernels were generated as shown in the third row of Fig. 12. The estimated blur kernels match their corresponding observed blur kernels indicating that the estimated TSF is quite accurate. The estimated depth map from the proposed method shown in Fig. 12c yet again depicts the scene structure correctly.

## 6 Conclusions and Discussions

In this work, we developed a scheme for recovering the 3D structure of a scene using the variation of motion blur as a depth cue. We used the notion of TSF to model non-uniform blur and suitably modified it to accommodate parallax effects. Initially, we proposed a method to determine the camera motion from locally estimated blur kernels. We exploited the linear relationship between the TSF and blur kernels, and solved for the TSF along with the relative depth values at different image points. Experiments show that our TSF estimation technique is accurate. From the recovered TSF, we estimate the dense depth map of the scene within a MAP

MRF framework. The performance of our depth estimation technique was found to be quite good on a variety of examples, and our estimates matched closely with the true depth values. As is the case with any blur-based depth estimation technique, our approach requires the scene to be sufficiently textured. Another limitation is that our method cannot handle large camera displacements between the two observations.

The approach followed in this paper is potentially an important precursor to efficient restoration of 3D scenes blurred by camera shake. We separately considered three different scenarios for camera motion: (a) in-plane rotations and 2D translations, (b) 3D translation, and (c) out-of-plane rotation and translation. A possible improvement to our work can be to simultaneously consider the 6D camera motion. Depth recovery from a single blurred image would be an interesting and challenging extension to our work. Other possible extensions to our work include handling of large displacements across the two observations, dynamic scenes and occlusion effects.

**Acknowledgments** The authors would like to thank the anonymous reviewers for their useful comments and suggestions. They gratefully acknowledge the support given by Department of Atomic Energy Science Research Council, India.

## References

- Alvertos, N., Brzakovic, D., & Gonzalez, R. C. (1989). Camera geometries for image matching in 3-D machine vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(9), 897–915.
- Babacan, S. D., Wang, J., Molina, R., & Katsaggelos, A. K. (2010). Bayesian blind deconvolution from differently exposed image pairs. *IEEE Transactions on Image Processing*, 19(11), 2874–2888.
- Bhavsar, A., & Rajagopalan, A. N. (2012). Towards unrestrained depth inference with coherent occlusion filling. *International Journal of Computer Vision*, 97(2), 167–190.
- Boracchi, G. (2009). Estimating the 3D direction of a translating camera from a single motion-blurred image. *Pattern Recognition Letters*, 30(7), 671–681.
- Brodatz, P. (1966). *Textures: A photographic album for artists and designers*. New York: Dover Publications.
- Canon U.S.A.: Lens Image Stabilization (2012). Retrieved December 25, 2012, from [http://www.usa.canon.com/cusa/consumer/standard\\_display/Lens\\_Advantage\\_IS.html](http://www.usa.canon.com/cusa/consumer/standard_display/Lens_Advantage_IS.html).
- Caglioti, V., & Giusti, A. (2008). On the apparent transparency of a motion blurred object. *International Journal of Computer Vision*, 86(2–3), 243–255.
- Chaudhuri, S., & Rajagopalan, A. N. (1999). *Depth from defocus: A real aperture imaging approach*. New York: Springer.
- Dai, S., & Wu, Y. (2008). Motion from blur. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*.
- Dannilidis, K., & Nagel, H. H. (1993). The coupling of rotation and translation in motion estimation of planar surfaces. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*.
- Favaro, P., & Soatto, S. (2004). A variational approach to scene reconstruction and image segmentation from motion-blur cues. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*.
- Favaro, P., Burger, M., & Soatto, S. (2004). Scene and motion reconstruction from defocused and motion-blurred images via anisotropic diffusion. In *Proceedings of European Conference on Computer Vision (ECCV)*.
- Favaro, P., & Soatto, S. (2006). *3-D shape estimation and image restoration exploiting defocus and motion-blur*. New York: Springer.
- Favaro, P., Soatto, S., Burger, M., & Osher, S. (2008). Shape from defocus via diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(30), 518–531.
- Felzenszwalb, P., & Huttenlocher, D. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 167–181.
- Felzenszwalb, P., & Huttenlocher, D. (2006). Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1), 41–54.
- Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., & Freeman, W. T. (2006). Removing camera shake from a single photograph. *ACM Transactions on Graphics*, 25(3), 787–794.
- Fox, J. S. (1988). Range from translational motion blurring. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*.
- Gupta, A., Joshi, N., Zitnick, L., Cohen, M., & Curless, B. (2010). Single image deblurring using motion density functions. In *Proceedings of European Conference on Computer Vision (ECCV)*.
- Hartley, R., & Zisserman, A. (2004). *Multiple view geometry in computer vision* (2nd ed.). Cambridge: Cambridge University Press.
- Hirsch, M., Schuler, C. J., Harmeling, S., & Scholkopf, B. (2011). Fast removal of non-uniform camera shake. In *Proceedings of International Conference on Computer Vision (ICCV)*.
- Hu, Z., & Yang, M. H. (2012). Fast non-uniform deblurring using constrained camera pose subspace. In *Proceedings of British Machine Vision Conference (BMVC)*.
- Joshi, N., Kang, S. B., & Zitnick, L. (2010). Image deblurring using inertial measurement sensors. *ACM Transactions on Graphics*, 29(4), 1–9.
- Klein, G., & Drummond, T. (2005). A single-frame visual gyroscope. In *Proceedings of British Machine Vision Conference (BMVC)*.
- Levin, A., Weiss, Y., Durand, F., & Freeman, W. T. (2009). Understanding and evaluating blind deconvolution algorithms. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*.
- Lin, H. Y., & Chang, C. H. (2006). Depth recovery from motion blurred images. In *Proceedings of International Conference on Pattern Recognition (ICPR)*.
- Liu, J., Ji, S., & Ye, J. (2009). SLEP: Sparse learning with efficient projections, Arizona State University. Retrieved December 25, 2012 from <http://www.public.asu.edu/~jye02/Software/SLEP>.
- Liu, J., & Ye, J. (2009). Efficient Euclidean projections in linear time. In: *Proceedings of International Conference on Machine Learning (ICML)*.
- Paramanand, C., & Rajagopalan, A. N. (2010a). Unscented transformation for depth from motion-blur in videos. In *Proceedings of IEEE Workshop on Three Dimensional Information Extraction for Video Analysis and Mining (in conjunction with CVPR)*.
- Paramanand, C., & Rajagopalan, A. N. (2010b). Inferring image transformation and structure from motion-blurred images. In *Proceedings of British Machine Vision Conference (BMVC)*.
- Paramanand, C., & Rajagopalan, A. N. (2012). Depth from motion and optical blur with unscented Kalman filter. *IEEE Transactions on Image Processing*, 21(5), 2798–2811.
- Portz, T., Zhang, L., & Jiang, H. (2012). Optical flow in the presence of spatially-varying motion blur. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*.
- Sellent, A., Eisemann, M., Goldlücke, B., Cremers, D., & Magnor, M. (2011). Motion field estimation from alternate exposure images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8), 1577–1589.
- Shan, Q., Jia, J., & Agarwala, A. (2008). High-quality motion deblurring from a single image. *ACM Transactions on Graphics*, 27(3).

- Sorel, M., & Flusser, J. (2008). Space-variant restoration of images degraded by camera motion blur. *IEEE Transactions on Image Processing*, 17(2), 105–116.
- Sorel, M., & Sroubek, F. (2009). Space-variant deblurring using one blurred and one underexposed image. In *Proceedings of International Conference on Image Processing (ICIP)*.
- Tai, Y., Du, H., Brown, M. S., & Lin, S. (2010a). Correction of spatially varying image and video motion blur using a hybrid camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6), 1012–1028.
- Tai, Y., Lin, S., Kong, N., & Shin, S. Y. (2010b). Coded exposure imaging for projective motion deblurring. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*.
- Tai, Y., Tan, P., & Brown, M. S. (2011). Richardson–Lucy deblurring for scenes under projective motion path. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8), 1603–1618.
- Watanabe, M., & Nayar, S. K. (1998). Rational filters for passive depth from defocus. *International Journal of Computer Vision*, 27(3), 205–225.
- Whyte, O., Sivic, J., Zisserman, A., & Ponce, J. (2010). Non-uniform deblurring for shaken images. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*.
- Whyte, O., Sivic, J., Zisserman, A., & Ponce, J. (2012). Non-uniform deblurring for shaken images. *International Journal of Computer Vision*, 98(2), 168–186.
- Xu, L., & Jia, J. (2012). Depth-aware motion deblurring. In *Proceedings of International Conference on Computational Photography (ICCP)*.
- Yuan, L., Sun, J., Quan, L., & Shum, H. Y. (2007). Image deblurring with blurred/noisy image pairs. *ACM Transactions on Graphics*, 26(3), 1–10.
- Zheng, Y., Nobuhara, S., & Sheikh, Y. (2011). Structure from motion blur in low light. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*.