Depth Estimation and Inpainting with an Unconstrained Camera

Arnav V. Bhavsar
arnav.bhavsar@gmail.com
A. N. Rajagopalan
raju@ee.iitm.ac.in

Image Processing and Computer Vision Lab
Department of Electrical Engineering
Indian Institute of Technology Madras
Chennai, India

Abstract

Unrestricted camera motion and the ability to operate over a range of lens parameters are often desirable when using an off-the-shelf camera. Variations in intrinsic and extrinsic parameters induce defocus and pixel motion, both of which relate to scene structure. We propose a depth estimation approach by elegantly coupling the motion and defocus cues. We further advocate a natural extension of our framework for inpainting both depth and image, using the motion cue. Unlike traditional inpainting, our approach also considers defocus blur. This ensures that the image inpainting is coherent with respect to defocus. We use the belief propagation method in our estimation approach, which also handles occlusions and uses the color image segmentation cue.

1 Introduction

For most practical cameras, one can typically control the intrinsic parameters viz. the focal length, aperture, focusing distance etc. Moreover, often the extrinsic parameters viz. camera rotation and translation can be freely varied. Such variations induce image effects such as parallax, occlusions, zooming, defocus blur etc. These are often inevitable due to restrictions on depth of field (DOF) and field of view (FOV) resulting from the limits on camera parameters. Importantly, the defocus and the motion effects can also serve as depth cues.

We formulate a general framework for depth estimation which respects the practical limits on the camera parameters. We elegantly couple the blur and motion cues through the camera parameters, as both are related to the scene depth. Indeed the shape estimation domains such as stereo [17], depth from defocus (DFD) [12] and shape from focus (SFF) [10, 15], where one restricts either the camera motion and/or the internal parameters, are essentially special cases of a framework such as ours. In general, one would like to operate the camera in an arbitrary fashion and such restrictions need to be relaxed. Thus, it is important to generalize the depth estimation task to offer more freedom in operating the camera.

We also extend our approach for inpainting images as well as depth, using observations with missing areas. Images can contain missing regions due to common faults in camera sensors and lenses. These include sensor contamination by dust and humidity while changing lenses [21], sensor damage from over-exposure to sunlight etc. Similarly, lens damage due to shocks as well as climatic effects, occlusions due to lens depositions/attachments etc can also lead to image artifacts [8, 14]. We exploit the motion cue for inpainting which allows the

BMVC 2010 doi:10.5244/C.24.84

correspondence and color information, missing in some images, to be present in others. Our approach also respects the fact that observations have different amounts of blur. The pixel mapping considers the blurring process in depth and image inpainting. The image inpainting occurs such that visual coherence with respect to defocusing is maintained.

1.1 Related work

Few works consider both blur and motion cues within a single shape estimation framework. Some references on passive and active methods that weakly relate blur and motion cues are provided in [1]. Here, we mention those methods which are closer to our work.

In [18], the authors estimate defocus and affine shifts, however, only for 2D scenes. The work in [5], uses variation in aperture size and position to induce parallax and defocus. However, this configuration has limited freedom and is specially manufactured. The authors in [20] consider defocus in structure from motion, although, for sparse structure computation.

Closely related to our work are [1, 13, 16], which strongly couple blur and motion. The work in [13] uses latarally translated binocular stereo while that in [16] involves axial camera translation with no intrinsic parameter variations. Thus, both these works involve restricted configurations. Moreover, they do not handle occlusions and their approaches use simulated annealing, which is quite inefficient. The method in [1], is restricted to camera translation and aperture variation. Ours is a non-trivial generalization that considers arbitrary motion, and variations in aperture as well as focusing distance. We explore the effect of general camera motion on blur variation and accommodate for the (actual/apparent) magnification due to zooming/axial motion. Our work also relaxes the constraint in [1], that all points in the reference image are more blurred than those in other images.

Importantly, none of the above mentioned works address the inpainting problem. Inpainting has been mainly reported for color and range images [2, 3], individually. However, there has been little work on inpainting both images and depth given only (damaged) color images of the scene. A recent work in a stereo setting addresses the removal of occluders, which are not stationary in the images since they are part of the scene [19]. Our approach attempts to inpaint the image artifacts such as those caused by lens/sensor defects, which are typically static in the camera reference frame. Thus, in our case the pixel mapping is considerably different than in [19]. Moreover, like traditional inpainting approaches the method in [19] does not account for defocus. Some approaches do consider defocus while inpainting [8, 21]. However, they use single views and focus only on image inpainting. In contrast, we consider both defocus and pixel motion to inpaint the image as well as depth map.

2 Joint modeling of warping and blurring

Without loss of generality, we establish the relationship between motion, blur and depth in camera-centered coordinates with the initial lens center coinciding with the world origin. The optical axis for the initial camera position, from which the reference image is captured, coincides with the z-axis, and the x- and y-axis are parallel to the image plane axes.

Denoting the (hypothetical) all-focused image from the reference view as f, the observed images g_is are modeled to be warped and blurred manifestations of f as

$$g_i(n_1,n_2) = \sum_{l_1,l_2} h_i(n_1,n_2,\sigma_i,\theta_{1i},\theta_{2i}) \cdot f(\theta_{1i},\theta_{2i}) + \eta_i(n_1,n_2) \tag{1}$$

Here, $(\theta_{1i}, \theta_{2i})$ denote the transformed pixel coordinates while $h_i(n_1, n_2, \sigma_i, \theta_{1i}, \theta_{2i})$ signifies the defocus kernel that blurs the pixel for the i^{th} image $f(\theta_{1i}, \theta_{2i})$. η_i represents the noise in the i^{th} observation. Fig. 1 shows perspective projection and blurring for 2 lenses in flatland. The reference lens is shown centered at the origin of the $x-z$ coordinate system with its associated image plane π_1. The i^{th} lens is rotated and translated with respect to the reference lens and has different internal parameters.

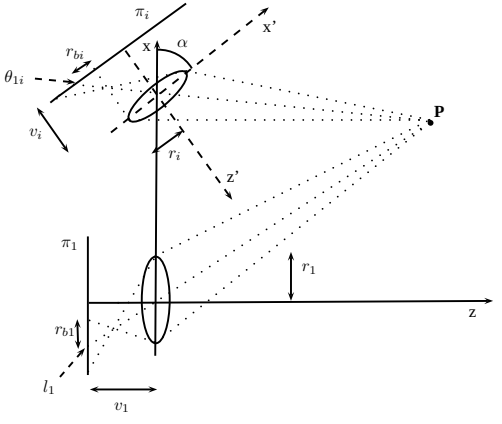


Figure 1: Image formation under general camera motion and parameter variations

2.1 Relating motion with depth

We denote the coordinates of a 3D point P as (X,Y,Z), with respect to the reference camera. The pixel coordinates (l_1, l_2) in the reference view corresponding to P are expressed as

$$l_1 = \frac{v_1 X}{Z} \qquad\qquad l_2 = \frac{v_1 Y}{Z} \tag{2}$$

where v_1 is the distance between the lens and image plane.

Denoting the lens-image plane distance in the i^{th} view by v_i, the camera translation along the 3 axes by a 3×1 vector $\underline{t} = [t_{xi}\ t_{yi}\ t_{zi}]^T$, and the camera rotation by a 3×3 matrix $R = [a_{pq}]$, where $1 \le p,q \le 3$, the 2D projection of P in the i^{th} view can be expressed as

$$\theta_{1i} = \frac{v_i a_{i11} X + v_i a_{i12} Y + v_i a_{i13} Z + v_i t_{xi}}{a_{i31} X + a_{i32} Y + a_{i33} Z + v_2 t_{zi}} \qquad \theta_{2i} = \frac{v_i a_{i21} X + v_i a_{i22} Y + v_i a_{i23} Z + v_i t_{yi}}{a_{i31} X + a_{i32} Y + a_{i33} Z + v_2 t_{zi}} \tag{3}$$

Eliminating X and Y, we relate pixel coordinates in two views in terms of the camera parameters and depth Z as

$$\theta_{1i} = \frac{v_i a_{i11} l_1 + v_i a_{i12} l_2 + v_1 v_i a_{i13} + v_1 v_i \frac{t_{xi}}{Z}}{a_{i31} l_1 + a_{i32} l_2 + v_1 a_{i33} + v_1 \frac{t_{zi}}{Z}} \qquad \theta_{2i} = \frac{v_i a_{i21} l_1 + v_i a_{i22} l_2 + v_1 v_i a_{i23} + v_1 v_i \frac{t_{yi}}{Z}}{a_{i31} l_1 + a_{i32} l_2 + v_1 a_{i33} + v_1 \frac{t_{zi}}{Z}} \tag{4}$$

2.2 Relating blur with depth

The Gaussian function is a popular approximation to the blur kernel owing to the effect of the central limit theorem on various optical aberrations [11, 12]. Thus, the blur kernel can

be expressed as

$$h_i(n_1,n_2,\sigma_i,\theta_{1i},\theta_{2i}) = \frac{1}{2\pi\sigma_i^2}\exp\left(-\frac{(n_1-\theta_{1i})^2+(n_2-\theta_{2i})^2}{2\sigma_i^2}\right) \tag{5}$$

It is well-known in DFD that for a real aperture camera, the blur parameter $\sigma_i = \rho r_{bi}$ in the i^{th} for a 3D point P, depends on the $Z-$coordinate of the 3D point as

$$\sigma_i = \rho r_i v_i \left(\frac{1}{f}-\frac{1}{v_i}-\frac{1}{Z_i}\right) \tag{6}$$

where r_{bi} is the blur radius, r_i is the aperture radius in the i^{th} view and Z_i is the depth of point P *with respect to the* i^{th} *camera.*

The equality $Z_1 = Z_i \forall i$ that is used in DFD, is not satisfied in our case due to general camera motion. Thus, the depth that decides the blur parameter for $\underline{X}$ in reference frame of the i^{th} camera is $Z_i \neq Z_1$. Since we are interested in the depth from the reference camera, we must express Z_i in terms of Z. Here, we invoke a relation in multi-view geometry that expresses the change in the coordinate system so that a 3D point $\underline{X_i}$ with respect to the reference camera can be written as $\underline{X_i} = R\underline{X} + \underline{t}$ with respect to a rotated and translated i^{th} camera. Thus, the depth of the point from the i^{th} camera can be expressed as $Z_i = a_{i31}X + a_{i32}Y + a_{i33}Z + t_{zi}$. Hence, using equations 2 and 6 we have

$$\sigma_i = \rho r_i v_i \left(\frac{1}{f}-\frac{1}{v_i}-\frac{1}{Z(\frac{a_{i31}l_1}{v_1}+\frac{a_{i32}l_2}{v_1}+a_{i33})+t_{zi}}\right) \tag{7}$$

The overall transformation of a particular point on f can be described by warping of a point to a new position followed by blurring at that new position. This order of transformations is also geometrically correct for a thin lens model as seen from Fig. 1. The blur kernel is formed around the point of projection of the principal ray on the image plane. Hence, the position of the blur kernel is also warped in the i^{th} image.

3 Depth estimation with belief propagation

The efficient-BP algorithm involves computing messages and beliefs on an image-sized grid. These are expressed as data and prior costs. For details, please refer to [7]. We next describe the cost computation for our problem.

3.1 Cost computation

Without loss of generality, we consider the first image as reference. At this point, for ease of explanation, we make an assumption (which we will soon relax) that the reference image is more blurred than the i^{th} image at all points. The reference image is modeled as a shifted and blurred version of the i^{th} image. This can be expressed by local convolutions between the warped i^{th} relative blur kernel h_{ri} and the warped i^{th} image as

$$g_1(n_1,n_2) = h_{ri}(\sigma_i,n_1,n_2) * g_i(n_1,n_2) = \sum_{l_1,l_2} h_{ri}(\sigma_i,n_1-\theta_{1i},n_2-\theta_{2i})\cdot g_i(\theta_{1i},\theta_{2i}) \tag{8}$$

where h_{ri} signifies the relative blur kernel corresponding to blur parameter $\sqrt{\sigma_1^2 - \sigma_i^2}$ which is a function of Z. The data cost for a particular node for a particular view is then defined as

$$E_{di}(n_1,n_2) = |g_1(n_1,n_2) - h_{ri}(\sigma_i,n_1,n_2) * g_i(n_1,n_2)| \tag{9}$$

Due to general camera parameter variations, g_1 need not be more blurred than g_i at all pixels. Hence, we now relax the assumption about the reference image being always more blurred than the others. The data cost in equation 9, will not be valid at those points where g_i is more blurred than g_1. In such cases, only the magnitude of $\sigma_1^2 - \sigma_i^2$ is not sufficient for labeling, 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 equations 9 to define the data cost. If for a depth label, we have $\sigma_1^2 - \sigma_i^2 < 0$, we define the data cost as

$$E_{di}(n_1,n_2) = |g_i(\theta_{1i},\theta_{2i}) - h_{ri}(\sigma_i,n_1,n_2) * g_1(n_1,n_2)| \tag{10}$$

The convolution on the right hand side of 10 is defined as

$$h_{r1}(\sigma_i,n_1,n_2) * g_1(n_1,n_2) = \sum_{l_1,l_2} h_{ri}(\sigma_i,n_1-l_1,n_2-l_2) \cdot g_1(l_1,l_2) \tag{11}$$

In this case, since g_i is more defocused than g_1, we blur g_1 to yield an estimate of g_i. Note that such a data cost computation will automatically handle the simpler case where g_1 is more blurred than g_i at all points (e.g. in scenarios with only variations in r).

At this point, we note that unlike equation 8, equation 1 does not involve a convolution since it models space-variant blurring. Hence, equation 8 is an approximation, which we follow to make our data cost amenable to the efficient-BP algorithm [11]. Indeed, this approximation only assumes that the scene is locally planar; a practical assumption. Similar approximations are reported in traditional and contemporary DFD works [1, 6].

To handle occlusions, we introduce a binary visibility function $V_i(n_1,n_2)$ which switches on/off depending on whether a pixel is visible or occluded in the i^{th} image. We thus have

$$E_{di}(n_1,n_2) = V_i(n_1,n_2) \cdot |g_1(n_1,n_2) - h_{ri}(\sigma_i,n_1,n_2) * g_i(n_1,n_2)| \tag{12}$$

The data cost in equation 10 can be modified similarly when considering visibility. For the first iteration, all pixels are considered visible. In subsequent iterations, V_i is computed by warping the current depth estimate to the i^{th} view. We also use geo-consistency to update visibility temporally, to mitigate the pathological errors due to incorrect labeling of an invisible pixel as visible and to yield a convergent solution [4, 9].

The total data cost for a particular node considering for all views is then computed as $E_d = \frac{1}{N_i}\sum_i E_{di}$, where $i > 1$ and N_i is the total number of images excluding the reference image where the pixel is visible.

To regularize the estimation, we use the MRF prior to enforce smoothness between neighbouring nodes. To avoid over-smoothing of prominent discontinuities, we define the smoothness prior cost as a truncated absolute function stated as

$$E_p(n_1,n_2,m_1,m_2) = \min(|Z(n_1,n_2) - Z(m_1,m_2)|, T) \tag{13}$$

where, (n_1,n_2) and (m_1,m_2) are neighbouring nodes and T is the threshold for truncation.

3.2 Segmentation constraint

The local convolution approximation, image noise and errors in visibility can result in somewhat incorrect depth estimates. We incorporate the image-segmentation cue to improve our estimation. The segmentation cue exploits a natural pattern of depth discontinuities coinciding with image discontinuities. Moreover, given a sufficiently over segmented image, each image segment can be assumed to have a planar depth variation.

Initially, we color-segment the reference image and classify the pixels as reliable or unreliable. The first BP iteration is run without using the segmentation cue. We then compute a plane-fitted depth map using the current depth estimate, segmented image and reliable pixels [1]. We feed the plane-fitted depth back to the iteration process to regularize the data term as

$$E_{ds}(n_1,n_2) = E_d(n_1,n_2) + w(n_1,n_2)\cdot|Z(n_1,n_2) - Z_p(n_1,n_2)| \tag{14}$$

where $E_d(n_1,n_2)$ is the data cost of equation 9 or equation 10. Z_p denotes the plane-fitted depth map and the weight w is 0/1 if the pixel is reliable/unreliable. The second term in equation 14 regularizes the unreliable estimates so that their labels are close to that of the plane-fitted depth map. We use this data term in subsequent iterations.

4 Extension to inpainting

We now discuss the adaptation of the above approach for inpainting. Given images from a real aperture camera with some areas marked as missing, we wish to estimate the depth and the image (from the reference view) with correct/plausible values assigned in the areas with missing observation. Importantly, the intensity assignment for the missing pixels in the image should also satisfy the defocus level to maintain visual coherence.

4.1 Depth inpainting

The location of the missing pixels, in case of sensor/lens damage, are constant in all the images. However, due to camera motion, the locations of pixels corresponding to scene points do vary. Hence, pixels missing in the reference image may be observed in other images. Thus, even if correspondences with the reference image cannot be found, they may yet be established between other images. We now formalize this idea in our cost computation.

We denote the set of missing pixels as M. We begin with arranging the images in an (arbitrary) order $(g_1, g_2, ..., g_N)$ with g_1 being the reference image. For pixels $\notin M$ in all images, the depth estimation approach is the one described in section 3.1. If a pixel $g_1(l_1,l_2) \notin M$ and $g_j(\theta_{1i},\theta_{2i}) \in M$ for some $i > 1$ then the data cost between the reference view and the i^{th} view is not computed. In case, $g_j(\theta_{1i},\theta_{2i}) \in M\ \forall i$, then the pixel is left unlabeled.

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

$$E_{di}(n_1,n_2) = V_{ij}(n_1,n_2)\cdot|g_i(\theta_{1i},\theta_{2i}) - h_{rij}(\sigma_{ij},n_1,n_2) * g_j(n_1,n_2)| \tag{15}$$

where $1 < i < j$, and

$$h_{rij}(\sigma_i,n_1,n_2) * g_j(n_1,n_2) = \sum_{l_1,l_2} h_{rij}(\sigma_i,n_1-\theta_{1j},n_2-\theta_{1j})\cdot g_j(\theta_{1j},\theta_{2j}) \tag{16}$$

$$V_{ij}(n_1,n_2) = V_i(n_1,n_2)V_j(n_1,n_2) \tag{17}$$

Here, h_{rij} denotes the blur kernel corresponding to $\sqrt{\sigma_i^2 - \sigma_j^2}$. 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. The corresponding data cost for $g_1(l_1, l_2)$ for all views for a particular depth label is then computed by summing the matching costs as $E_d = \frac{1}{N_{ij}} \sum_i E_{di}$, where N_{ij} are the number of pairs of images g_i and g_j such that $g_i(\theta_{1i}, \theta_{2i}) \notin M$ and $g_j(\theta_{1j}, \theta_{2j}) \notin M$ 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 pixel is visible.

Equation 15 implicitly assumes that $g_j(\theta_{1j}, \theta_{2j})$ is blurred and compared with $g_i(\theta_{1i}, \theta_{2i})$. However, like in section 3.1, this is assumed only for ease of explanation. The vice-versa case can be handled easily in a similar way as discussed in section 3.1.

The above process can yield pixels which are not labeled (for which no correspondences are found). Moreover, as discussed in section 3.2, some pixels can also be labeled incorrectly. We invoke the segmentation cue similar to to that explained in section 3.2 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. The *closeness* is determined by searching in eight directions around the pixel. The idea is that, if the missing regions did not exist, most pixels in S_m would actually belong to the segments to which they are now assigned.

The plane-fitted depth map is then computed using the reliable pixels in these extended segments, (including reliable pixels from segments which were earlier in S_m). This plane-fitted depth map is fed back in 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 help to 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 $g_i(\theta_{1i}, \theta_{2i})$, $i > 1$ with an intensity label, if $g_i(\theta_{1i}, \theta_{2i}) \notin M$. This data cost is defined as

$$E_d(n_1, n_2) = V(n_1, n_2) \cdot |L - h_{ri}^p(\sigma_i, n_1, n_2) * g_i(n_1, n_2)| \tag{18}$$

where L is an intensity label, and the convolution is defined as in 8. We note that the image inpainting accounts for the blurring process so that the inpainted image intensity is assigned according to the defocusing that takes g_i to g_1. The kernel superscript p denotes that h_{ri}^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 $i > 1$ at the pixels to be inpainted. The above data cost is computed only for the g_is satisfying this condition, which is easily met, given sufficient images. The prior cost for image is defined similar to the one used for depth estimation, except it operates on intensity labels.

Lastly, there may be missing pixels in g_1 pixels for which $g_i(\theta_{1i}, \theta_{2i}) \in M \forall i$. Such pixels are left unlabeled. The extent of such unlabeled regions depends on the original extent of the missing region and pixel motion. In our experiments, for most cases, the pixel motion

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, the exemplar-based inpainting method [3])

5 Experimental results

We validate our approach on various real images. The observations were captured in our laboratory with the Olympus C-5050 camera. The internal parameters for this camera are known from the exif data stored in the captured images. The focal length of the camera is of the order of 1-2 cm. The distance range of the scene is 20-50 cm, within which we vary the focusing distance. The camera translations and rotations are about 5-15 mm and 5-10 degrees respectively, which were measured while capturing the images. The f-number varies between F/8 - F/4. In all experiments, T in the prior cost is chosen as half of the maximum depth label for depth estimation and 50 gray levels for image inpainting. We use depth labels in steps of 0.5. In the following examples, the first image is the reference observation.

5.1 Results for depth estimation

We begin with a real result involving translation, rotation and aperture variation in Fig. 2. Figs. 2(a-c) show three of the five observations. Our depth map (Fig. 2(d)) shows a very plausible depth variation with well-defined discontinuities. Fig. 2(e) shows a novel view.

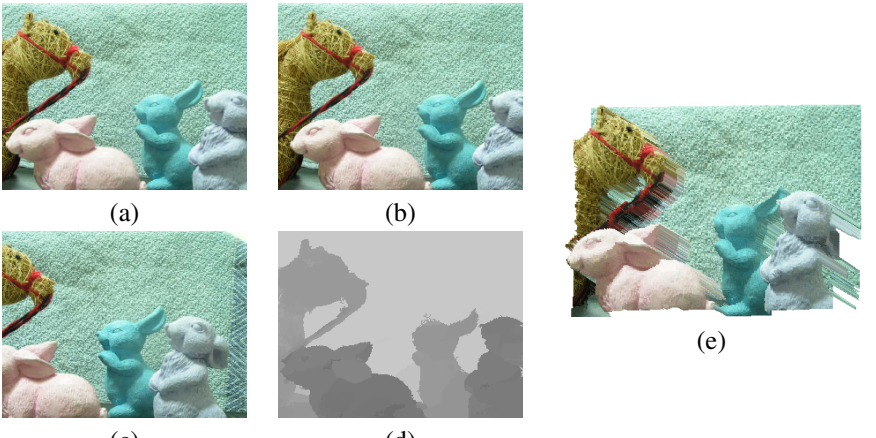


Figure 2: Real experiment: (a,b,c) Observations with translation, rotation and aperture variation. (d) Estimated depth map. (e) Novel view Rendering

In our next real experiment the camera, focused on the Ashoka pillar model (leftmost), is translated and varied in aperture. Three of the four observations are shown in Figs. 3(a-c). Note the heavy blurring over the Ganesha idol and the background, and observe the low texture on the wooden objects. The estimated depth map (Fig. 3(d)) again captures the variations and is well localized; Fig. 3(e) shows a novel view of the scene. We also evaluate our approach in this experiment, by comparing our depth estimates with some actual measured distances from the camera to some regions on the objects. We find that our estimated distances agree well with the measurements (Table 1).

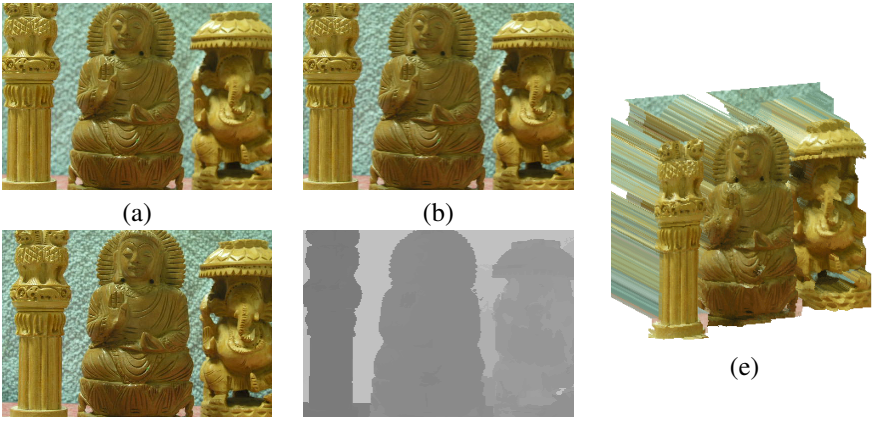


Figure 3: Real experiment: (a,b,c) Translated observations with aperture variation. (d) Estimated depth map (e) Novel view Rendering.

Table 1: Comparison with ground-truth distances for the scene in Fig. 3

Objects/regions	Measured distance (cm)	Estimated distance (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

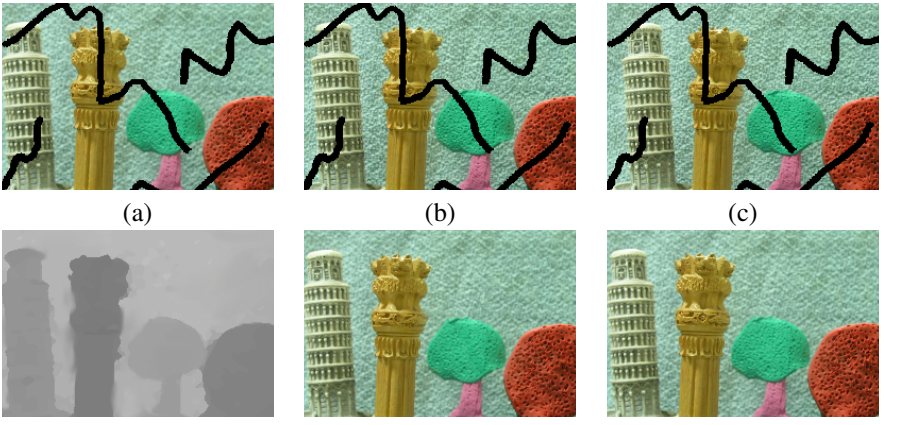
5.2 Results for inpainting

We now show results for image and depth inpainting where we have introduced about 20 pixel wide scratches in the images to emulate the damaged observations. Figs. 4 shows a real result involving translation, rotation and aperture variation. In the three of the four observations (Figs. 4(a-c)), observe the blur variations in the background, green tree and Pisa tower. Our depth map output (4(d)) is cleanly inpainted even at the discontinuities. Comparing the original unscratched image (Fig. 4(f)) and the inpainted image (Fig. 4(e)), we find that the scratches are filled without hampering the details and texture on the objects and the defocus in the inpainted areas is coherent with the neighbourhood.

Our final real result involves camera translation, and variation in aperture and focusing distance. Observe the near-far focusing and heavy blurring in the three of the four observations (Figs. 5(a-c)). Again, our depth estimate (Fig. 5(d)) shows good depth variations and localization. In addition, the depth inpainting is quite flawless in the damaged regions. With very differently blurred observations, the image inpainting (Fig. 5(e)) further validates our claim of accounting for blur while inpainting. The visually correct defocus, can be observed when the inpainted images is compared with the actual undamaged observation (Fig. 5(f)).

6 Conclusion

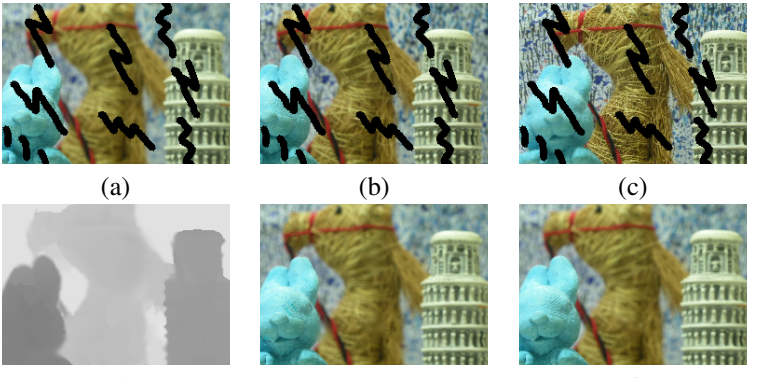
We proposed a depth estimation framework that considers general motion and camera parameter variations. It can account for various effects such as motion parallax, occlusions, zooming, blur variation under general camera motion, arbitrary focusing etc. We further extended our approach for image and depth inpainting from damaged observations. In future, it would be interesting to extend the inpainting method to handle non-stationary occluders.



(a) (b) (c)

(d) (e) (f)

Figure 4: Real result: (a,b,c) Observations with translation, rotation and variations in aperture. (d) Estimated depth map. (e) Inpainted image. (f) Actual unscratched image.



(a) (b) (c)

(d) (e) (f)

Figure 5: Real result: (a,b,c) Observations with translation, and variations in aperture and focusing distance. (d) Estimated depth. (e,f) Inpainted and undamaged image, respectively.

References

[1] A. V. Bhavsar and A. N. Rajagopalan. Depth estimation with a practical camera. *British Machine Vision Conference (BMVC 2009)*, 2009.

[2] A. V. Bhavsar and A. N. Rajagopalan. Range map with missing data - joint resolution enhancement and inpainting. *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP 2008)*, pages 359–365, 2008.

[3] A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based inpainting. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, pages 721–728, 2003.

[4] M. Drouin, M. Trudeau, and S. Roy. Geo-consistency for wide multi-camera stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 1: 351–358, 2005.

[5] Q. Duo and P. Favaro. Off-axis aperture camera: 3d shape reconstruction and image restoration. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pages 1–7, 2008.

[6] P. Favaro, S. Soatto, M. Burger, and S. Osher. Shape from defocus via diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(3):406–417, 2005.

[7] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, pages 1: 261–268, 2004.

[8] J. Gu, R. Ramamoorthi, P. Belhumeur, and S. Nayar. Removing image artifacts due to dirty camera lenses and thin occluders. *SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers*, pages 1–10, 2009.

[9] Y. Nakamura, T. Matsura, K. Satoh, and Y. Ohta. Occlusion detectable stereo - occlusion patterns in camera matrix. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 1996)*, pages 371–378, 1996.

[10] S. K. Nayar and Y. Nakagawa. Shape from focus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(8):824–831, 1994.

[11] A. Pentland. A new sense for depth of field. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(4):523–531, 1987.

[12] A. N. Rajagopalan and S. Chaudhuri. *Depth from defocus: A real aperture imaging approach*. Springer-Verlag New York, Inc., New York, 1999.

[13] A. N. Rajagopalan, S. Chaudhuri, and U. Mudenagudi. Depth estimation and image restoration using defocused stereo pairs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26 (11):1521–1525, 2004.

[14] R. Sahay and A. N. Rajagopalan. Inpainting in shape from focus: Taking a cue from motion parallax. *British Machine Vision Conference (BMVC 2009)*, 2009.

[15] R. R. Sahay and A. N. Rajagopalan. A model-based approach to shape from focus. In *Proceedings of the 3rd International Conference on Computer Vision Theory and Applications (VISAPP 2008)*, pages 1: 243–250, 2008.

[16] R. R. Sahay and A. N. Rajagopalan. Real aperture axial stereo: Solving for correspondences in blur. *DAGM-Symposium 2009*, pages 362–371, 2009.

[17] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1).

[18] S. Seitz and S. Baker. Filter flow. *Proc. International Conference on Computer Vision (ICCV 2009)*, 2009.

[19] L. Wang, H. Jin, R. Yang, and M. Gong. Stereoscopic inpainting: Joint color and depth completion from stereo images. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pages 1–8, 2008.

[20] C. Wohler, P. d'Angelo, L. Kruger, A. Kuhl, and H. M. Grob. Monocular 3d scene reconstruction at absolute scale. doi:10.1016/j.isprsjprs.2009.03.004.

[21] C. Zhou and S. Lin. Removal of image artifacts due to sensor dust. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, pages 1–8, 2007.