Depth estimation with a practical camera

Arnav V. Bhavsar
arnav.bhavsar@gmail.com
A. N. Rajagopalan
http://www.ee.iitm.ac.in/~raju

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

Abstract

Given an off-the-shelf camera, one has the freedom to move the camera or play around with its intrinsic parameters such as zoom or aperture settings. We propose a framework for depth estimation from a set of calibrated images, captured under general camera motion and parameter variation. Our framework considers the practical trade-offs in a camera and hence essentially generalizes the more constrained areas such as lateral or axial stereo, shape from defocus/focus etc. We discuss practical issues where such an approach becomes important. We pose the problem in a MAP framework and compute the depth estimates efficiently using belief propagation (BP). We also incorporate the visibility consideration to handle occlusions. Moreover, we use the vital cue from color image segmentation to constrain the estimation process. Our results demonstrate the effectiveness of our approach to localize discontinuities and handle low-textured regions

1 Introduction

Various cues are exploited by vision systems to compute depth information. Typically, the cues may be classified as those embedded in the scene itself, such as texture, illumination, perspective etc and those induced by the imaging device, such as motion, blur/accommodation etc. We concern ourselves with the latter category i.e. the cues depending on the camera parameters, on which one typically has more control than on the scene conditions. The camera parameters include aperture radius, distance between image plane and lens etc, while camera motion involves translation and rotation. Variations in these cause effects such as focusing/defocusing, zooming, parallax, occlusions etc. The blur and motion carry important information about the shape of the 3D world.

In fact, the axial/lateral stereo [11], depth from defocus (DFD) [6] etc problems are essentially special cases of this more general scenario. In these, one assumes restrictions on either the motion and/or the internal parameters. However, due to physical limits of the camera, such assumptions may not always hold. For instance, the pin-hole model assumed for stereo, is not valid when one operates on distances beyond the Depth of field (DOF) and the images suffer from defocus blur. On the other hand, single view scenario in DFD/SFF, may not be applied in situations where one needs to have a larger field of view (FOV) and needs to move the camera. Thus, a generalization of the task of shape estimation, that respects the physical limits of cameras, is an important practical issue.

BMVC 2009 doi:10.5244/C.23.104

1.1 Relation to previous work

Few works have been reported when both blurring and motion cues are taken into account in the same framework. In [1], the authors have proposed a technique to compute affine motion and the defocus blur simultaneously in images of planar scenes. The work has been extended to estimate defocus blur and arbitrary spatial shifts in [3]. However, in these works, only the defocus blur is exploited as a cue for shape.

Some works [12] have motivated the problem of shape estimation from a cue-combination point of view. They employ a sequential process, estimating the shape using one of the DFD/SFF and stereo techniques and improving this estimate by applying the other. Some active approaches have also been reported [2] that actively control the focus and vergence. Although these techniques utilize both motion and defocus cues, they independently use stereo, DFD/SFF techniques and require an image configuration which is tailored for these specific techniques. As we argue, such configurations may not be available due to the camera limits. Moreover, the methods do not relate motion, blur and depth and use (the less reliable) local window based DFD/SFF techniques.

Some works have considered a strong coupling between blur and motion since both are related to depth [4, 5, 7]. In [4], the authors create a virtual stereo pair by varying both the size and the lateral position of the aperture and thus, giving rise to parallax and relative blur. However, this configuration has limited freedom and it also requires a special camera setup. The authors in [5] consider depth estimation in a binocular stereo scenario with lateral motion. Here, multiple (around 10) differently focused images are captured from each view and a window to constraint the disparity estimation is computed using the blur-disparity relationship. However this relationship is used locally and only to define windows for disparity estimation. The authors in [7] use this strong coupling in a MAP-MRF framework to compute blur map in a binocular stereo setting. They also capture more than one image from each view with different focus setting.

The novelties of this work as compared to the above mentioned works are 1) Our image acquisition process offers more freedom so as not to be constrained by a strict DFD, SFF or a stereo like setting. We consider a calibrated camera configuration with general camera motion and parameter variations. 2) We compute the MAP solution in an efficient manner using the popular fast-BP algorithm [13]. To our knowledge, this is first time that BP has been applied for depth estimation involving the effect of defocus blur. 3) We also address the important issue of visibility that has not been addressed in the above mentioned works. 4) We constrain the estimation process by using a cue from color image segmentation [14]; a cue that has been shown to be very successful in conventional stereo works [14].

Before getting into the details of our approach, we discuss some practical examples of camera parameters where a general approach for shape estimation will indeed be important; a discussion which may be important to consider domains for use of practical cameras. For the intrinsic parameters, we consider variations in the aperture and lens to image plane distance viz. the two commonly controlled parameters. Our framework also accommodates the zooming process caused by variation in the latter. However, we assume (as is common in traditional DFD) that the scene completely lies in the forward or the reverse blur cone, i.e. the point of focus is beyond the farthest point or closer than the nearest point in the scene. Also, for this work, the camera motion is restricted to only translation along all the 3 axes. However, our framework can be extended to camera rotation in a straightforward way.

2 Practical camera limits

Typically, for modern cameras the internal parameters that a user can control are the aperture size, the effective focal length and the lens to image plane distance. These parameters control factors such as DOF and FOV which affect the image quality, resolution and spatial content.

The DOF is defined as the range of the distances in the real world where the object appears to be focused in its image. The DOF can be expressed as [10]

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

Here, f_n is the f-number of the lens (inversely proportional to the aperture radius r), f denotes the focal length, u stands for the working distance, m expresses the magnification and c denotes the acceptable circle of confusion which typically is fixed for a sensor size and an image format. It can be shown that the DOF is inversely related to r, u and f. The angle of view, that also depends on f, u and the sensor dimension d is given by

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

Note that α is also inversely related to f. The FOV is directly related to α and r.

By closely scrutinizing the above equations, we can deduce the following:

- Moving the camera closer to the objects to acquire enough resolution will reduce the FOV. Also, the objects at close distances can lie beyond nearer limit of DOF and resulting in a defocused image.
- Another way to acquire sufficient resolution is to increase the magnification. However, this means increasing f, which will also cause reduction in the FOV. Also, this will decrease the nearer DOF limit, increasing the chance of inducing defocus blur.
- To increase FOV, one can either increase r or reduce f. The latter will reduce magnification and hence the resolution. Increasing r but will decrease the DOF resulting in increased defocus blur.

Thus, there are trade-offs between acquiring sufficient detail, content and image quality. In Table 1 we provide some data for an Olympus digital camera in this respect. [8, 9].

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

Table 1: DOF variation with f_n, u for Olympus C-5050 (for $f = 14.2mm$)

Note that at $u = 20cm$, the object closer than $16cm$ or farther than $22cm$, will result in a defocused image. Moreover, in the super-macro mode of the camera (typically used between $u = 3cm$ to $20cm$), the DOF is even smaller with the FOV being about $5cm$ of the real world.

The above discussion point towards the fact that in some very practical situations, one must compromise the image quality if one wishes the desirable resolution, no matter how much one plays around with the camera parameters.

3 Image generation: Coupling motion, blur and depth

We now discuss how we relate motion, blur and depth. We follow the convention of the z-axis being parallel to the optical axis direction, and the x- and y-axis being parallel to the image plane axes. The observed images g_is are modeled to be warped and blurred manifestations of an ideally non-blurred image f as

$$g_i(n_1,n_2) = \sum_{l_1,l_2} h_i(n_1,n_2,\sigma_i,\theta_{1i}(l_1),\theta_{2i}(l_2)) \cdot f(\theta_{1i}(l_1),\theta_{2i}(l_2)) + \eta_i(n_1,n_2) \tag{3}$$

In the above equation, the geometric transformation of pixels is denoted by $\theta_{1i}(l_1)$ and $\theta_{2i}(l_2)$ while $h_i(n_1,n_2,\sigma_i,\theta_{1i}(l_1),\theta_{2i}(l_2))$ signifies the kernel that blurs a pixel in the i^{th} image $f(\theta_{1i}(l_1),\theta_{2i}(l_2))$.

In the following discussion we refer to Fig. 1 which shows perspective projection and blurring for 2 lenses. Let us denote the coordinates of a 3D point in space as (X,Y,Z), with

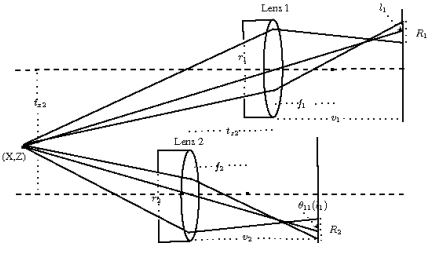


Figure 1: Camera parameters

respect to the reference camera position. Denoting the camera translation by t_{xi}, t_{yi} and t_{zi} along the x-, y- and z-axes respectively, the projection of this ray in the reference view and i^{th} view can be expressed as

$$l_1 = \frac{v_1 X}{Z} \quad l_2 = \frac{v_1 Y}{Z} \quad \text{and} \quad \theta_{1i}(l_1) = \frac{v_i(X+t_{xi})}{z+t_{zi}} \quad \theta_{2i}(l_2) = \frac{v_i(Y+t_{yi})}{z+t_{zi}} \tag{4}$$

where v_1 and v_i are the lens to image plane distances in the reference and the i^{th} view, respectively. Eliminating X and Y and denoting the ratio v_i/v_1 as v_r, we can relate pixel coordinates in two views in terms of the camera parameters and depth Z as

$$\theta_{1i}(l_1) = \frac{v_r l_1 Z + v_i t_{xi}}{Z+t_{zi}} \text{ and } \theta_{1i}(l_2) = \frac{v_r l_2 Z + v_i t_{yi}}{Z+t_{zi}} \tag{5}$$

Having described the pixel motion, we now focus our attention on blurring. The Gaussian function is a popular model to the blur kernel owing to the effect of the central limit theorem on various optical aberrations [4]. This Gaussian blur kernel can be expressed as

$$h_i(n_1,n_2,\sigma_i,\theta_{1i}(l_1),\theta_{2i}(l_2)) = \frac{1}{2\pi\sigma_i^2} \exp\left(-\frac{(n_1-\theta_{1i}(l_1))^2+(n_2-\theta_{2i}(l_2))^2}{2\sigma_i^2}\right) \tag{6}$$

The blur parameter σ in the above equation is related to the absolute depth Z from the lens. As the camera translates along the optical axis by t_{zi} for the i^{th} image, the depth of the point is now $Z \pm t_{zi}$. The blur parameter σ_i, in the i^{th} camera position is related to this depth

through the aperture r_i, lens to image plane distance v_i, working distance u_i and focal length f as

$$\sigma_i = \rho r_i v_i \left(\frac{1}{f} - \frac{1}{v_i} - \frac{1}{Z \pm t_{zi}} \right) \tag{7}$$

Note that in equation (6) the blur kernel is centered on the warped pixel $(\theta_{1i}(l_1), \theta_{2i}(l_2))$. Thus, according to the above model and Fig. 1 the blur kernel is formed around the point where the central ray projects on to the image plane. Hence, the position of the blur kernel is also warped in the i^{th} image.

From a physical point of view, aperture variation will cause variation in the blur parameter. The change in the lens to image plane distance causes blur variation as well as global scaling, whereas camera translation is responsible for depth-dependent pixel motion as well as blur variation for t_z translation. Thus, pixel warping, blur warping and blur magnitude all are related to the depth. Hence, in equation (3), the parameters σ_i, $\theta_{1i}(l_1)$ and $\theta_{2i}(l_2)$ are strongly coupled through a common unknown Z, which we wish to estimate.

4 Energy minimization using belief propagation

We now move on to describe our estimation approach. We formulate the depth estimation problem in a MAP framework which we solve using the fast-BP algorithm [13]. The max-product BP computes the MAP estimates over a graph [13]. For images, the graph is usually a grid-graph, with graph nodes as pixel locations. (For the following discussion on BP, we denote pixel locations as p, q and s, for conciseness). The max-product rule works by passing messages $m^t_{pq}(f_q)$ at time t to a node q from its neighbouring node p of the graph as follows

$$m^t_{pq}(f_q) = \min_{f_p} \left(V(f_p, f_q) + D_p(f_p) + \sum_{s \in N(p)|q} m^{t-1}_{sp}(f_p) \right) \tag{8}$$

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

$$b_q(f_q) = D_q(f_q) + \sum_{p \in N(q)} m_{pq}(f_q) \tag{9}$$

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

In recent years, belief propagation has been demonstrated to be one of the better performing algorithms in stereo vision domain [14, 17, 18]. As mentioned earlier, our work can be looked at as generalization to the multi-view stereo problem that also handles the defocus blur caused by the physical limits of a camera. Hence, in our case, the data cost that must be computed at each node in the BP algorithm is considerably different than that used in the traditional stereo depth estimation. Our cost computation also takes into account the defocus blur at each node along with its depth-dependent motion (as explained in the next subsection). Moreover, the pixel-motion can also be due to the zoom variation in the camera, which is also not considered in traditional BP based stereo approaches.

In our framework, the data cost is formulated from the image generation process described in the previous section. The prior cost is chosen to be a smoothness penalty that constrains neighbouring nodes to accept similar labels. Moreover, we modulate the data cost with a visibility term which is updated at each iteration. Furthermore, to improve the depth estimate, we use a cue from color-image segmentation and plane-fitting, inspired from recent works in the stereo vision [14, 17].

4.1 Data cost, prior cost and visibility consideration

To define the data cost we relate the image in the i^{th} view with that in the reference view. The reference image is modeled as the shifted and blurred version of the i^{th} image. Thus, the relationship between the reference image and the i^{th} image is given as

$$g_1(n_1,n_2) = h_{ri}(\sigma_i,n_1,n_2) * g_i(n_1,n_2) \tag{10}$$

where

$$h_{ri}(\sigma_i,n_1,n_2) * g_i(n_1,n_2) = \sum_{l_1,l_2} h_i(\sigma_i,n_1-\theta_{1i}(l_1),n_2-\theta_{2i}(l_2)) \cdot g_i(\theta_{1i}(l_1),\theta_{2i}(l_2)) + \eta_i(n_1,n_2) \tag{11}$$

Here, h_{ri} signifies the relative blur kernel which corresponds to blur parameter $\sqrt{\sigma_1^2-\sigma_i^2}$. The symbol $*$ denotes convolution. This blur parameter can be related to Z simply by substituting the equation (7) for σ_1 and σ_i. The data cost is then defined as

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

At this point, we note that equation (11) is a convolution of a shifted blur kernel and a shifted reference image. However, equation (3) does not involve a convolution since it models the image generation under space-variant blur. Hence, equation (10) is an approximation to the actual image generation model. We follow this approximation to make our data cost amenable to the BP algorithm. The data cost in the BP algorithm, for a particular label at a node, is defined as *the cost incurred by that node for accepting a particular label*. For applications such as de-noising or stereo disparity estimation, the data cost at a particular node involves only the label at that node and hence can be easily computed. However, for applications involving space-variant blur, the data cost at a node also involves labels in the neighbouring nodes. Since BP does not entertain a notion of current label estimates, such a data cost which depends on neighbouring node labels cannot be defined. Defining the data cost at a node as in equation (12), through a convolution approximation, makes it dependent only on the label at that node. The convolution $h_{ri}(\sigma_i,n_1,n_2) * g_i(n_1,n_2)$ computes an estimate $\hat{g}_1(n_1,n_2)$ of the intensity value for a particular node (n_1,n_2). Since this estimate is computed for a single node (n_1,n_2), it can be compared with the observation at that node $g_1(n_1,n_2)$. We can hence define the data cost at a single node thus allowing the cost to be minimized in an efficient BP framework.

The prior cost enforces a smooth solution that constrains the neighbouring nodes to have similar labels. At a more fundamental level, a smoothness constraint on depth actually manifests from modeling the depth as a Markov random field having a joint Gibbs distribution. However, an elaborate discussion on MRFs and Gibbs distribution is out of the scope of this paper. We define the smoothness prior as a truncated absolute function which is stated as

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

where, (n_1,n_2) and (m_1,m_2) are neighbouring nodes in a 4-connected neighbourhood. The truncation is carried out to avoid over-smoothing and allow discontinuities in the solution.

We incorporate the notion of visibility in the above data term. We introduce a binary visibility function V. For a particular site (n_1,n_2) on the reference grid, $V_i(n_1,n_2)$ is 1 if the pixel at that site is visible in the i^{th} image and 0 if the pixel is occluded. The visibility function V_i modulates the datacost as

$$D(n_1,n_2) = V_i(n_1,n_2) \cdot |g_1(n_1,n_2) - h_{ri}(\sigma_i,n_1,n_2) * g_i(n_1,n_2)| \tag{14}$$

In the beginning, all pixels are considered visible. After the first iteration, the visibility is computed by warping the current estimate of the depth in the i^{th} view. As observed in [15, 16] for the stereo problem, computing visibility in each iteration independently may not yield a convergent solution [16]. Hence, we use the geo-consistency definition [15] to update visibility temporally as

$$V_i(n_1,n_2,t) = V_i^{\text{new}}(n_1,n_2,t) \cdot V_i(n_1,n_2,t-1) \tag{15}$$

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

4.2 Segmentation cue

Because we use the convolution approximation in defining our data cost, the estimation process does not exactly follow the generation model of equation (3). This effect along with the presence of occlusions and other noise-induced outliers can make the depth estimation process error prone. Recently, the cue from image segmentation has shown great success in improving stereo disparity estimation [14]. Inspired by [14], we also incorporate the segmentation cue in our estimation process.

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

$$C = \frac{|C_1 - C_2|}{C_2} \tag{16}$$

This measure is similar to that defined in [14]. However, we note that in [14] the costs C_1 and C_2 are computed using a correlation volume. However, to also account for the blurring effect while computing the confidence, we use the data cost itself. If the confidence measure is above a particular threshold c_f then we define that pixel to be a reliable pixel. After this initial processing, we move on to the actual estimation process.

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

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

$$D(n_1,n_2) = V(n_1,n_2)\cdot|g_1(n_1,n_2) - h_{ri}(\sigma_i,n_1,n_2) * g_i(n_1,n_2)| + w\cdot|Z(n_1,n_2) - Z_p(n_1,n_2)| \quad (17)$$

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

5 Results

We classify our experiments into synthetic, semi-synthetic and real. The synthetic and the semi-synthetic experiments were carried out on Middlebury stereo database [□]. For the synthetic experiments, we created the warped and blurred observations from a focused image and depth map. For the semi-synthetic case, we used the independent focused stereo observations and blurred them synthetically using the depth map, while maintaining the blur, motion and depth relationships. In both these cases we blur each image in a space-variant way using (3), while during estimation we use the data cost of (12). In real experiments, we used the Olympus C-5050 camera. In all experiments T in the prior cost is chosen as half of the maximum depth label. The values of c_f, s_f and r_f for segmentation, are chosen as 0.2, 300 and 0.7, respectively. We use depth labels in steps of 0.5.

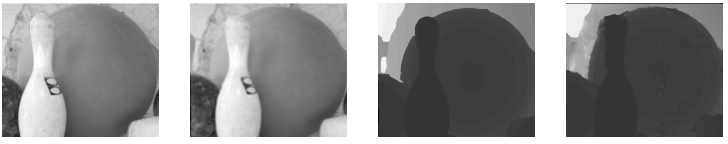
(a) (b) (c)

Figure 2: Synthetic experiment: Translation and aperture variation. (a) The reference blurred observation (c,d) Ground-truth and estimated depth maps, respectively.

We start with a synthetic experiment with t_x translation and aperture variation. The parameters were chosen such that the maximum disparity was 20 pixels and the relative blur parameter ranged from 0.8 to 2.2. Fig. 2(a) shows the reference image. Figs. 2(b) and (c), respectively, show the ground truth and the estimated depth map. Note that the edges are well localized and the gradual disparity variation is also captured.

We next show another synthetic experiment that involves the effects of zooming (change in v), variation in aperture, and t_x translation. We show two images in Figs. 3(a) and (b) to highlight the zooming and the translation effect. The scale between the images is 0.9 and the translation and the blurring is similar to that in the above example. The ground truth and the estimated depth are shown in Figs. 3(c) and (d), respectively. Again, note that the estimated depth map both in terms of localization and depth variations is very close to the ground-truth. This demonstrates that our approach can handle zooming effects induced while varying the defocus. This example also shows the ability of our algorithm to work with low-textured scenes (an ability uncommon to algorithms that compute depth from defocus blur).

We now show a semi-synthetic result in 4. This involves variation in aperture and t_x translation, wherein the focussed images themselves involve the translation. The pair that



(a) (b) (c) (d)

Figure 3: Synthetic experiment: (a,b) Observations with tranlsation, blur and zoom. (c,d) Ground-truth and estimated depth maps, respectively.

we chose from the dataset had a maximum disparity of 36 pixels. The hypothetical camera parameters are chosen such that the maximum relative blur is about 2.5. Fig. 4(a) shows the blurred reference image and Figs. 4(b) and (c) show the ground truth and the estimated depth map, respectively. In the background window region, we miss some very fine details. However, at other places (e.g. the bottle, the cloth at the bottom and the pillow at the right), the discontinuities and gradual depth variations are recovered quite well.

(a) (b) (c)

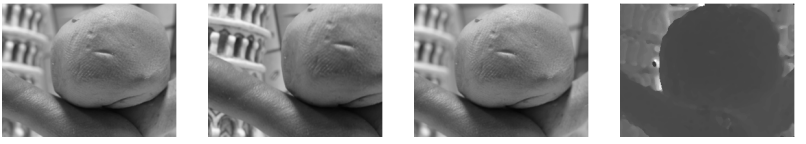
Figure 4: Semi synthetic experiment: Translation and aperture variation. (a) The reference blurred observation (b) Ground-truth depth map. (c) Estimated depth maps

We next demonstrate results on some real data that we captured using the Olympus C-5050 camera. We used the camera in macro mode, with the scene distance range being 3 to 15 cm. Figs. 5(a,b) show two images which involved t_x translation and variation in the aperture setting. The recovered depth map is shown in 5(c) for the reference frame of Fig. 5. Note that the discontinuity localization is accurate and fine variations (e.g. on the Pisa tower-model and on the circuit board components) are also captured.

(a) (b) (c)

Figure 5: Real experiment. (a,b) t_x translated mages with with and aperture variation. (c) Estimated depth map.

The next example (Fig. 6) demonstrates the efficacy of the framework to handle low-textured surfaces, as well as detailed surfaces in the same image. For this example, we have a multiple-image configuration involving t_x and t_z translations and variation r. In Figs. 6(a-c) we show 3 of the 4 observations used, with Fig. 6(c) as the reference image. Note that the objects in Fig. 6(b), appear closer due to the t_z translation. One can appreciate the low texture on the clay ball and the details on Pisa tower model. The estimated depth map (Fig. 6(d)) faithfully captures the gradual depth variations and the discontinuities of the low-textured object as well as the fine variations on the Pisa tower model.



(a) (b) (c) (d)

Figure 6: Real experiment. (a-c) Images with t_x and t_z translation with change in aperture. (d) Estimated depth map.

6 Conclusion

We proposed a depth estimation framework that relates the camera parameters, camera motion and depth cues. We outlined practical cases where such a framework will become important. We formulated the estimation in a BP framework that handles both motion and defocus blur. Moreover, we incorporated the notion of visibility and used the segmentation cue in the estimation. Our results show the effectiveness of the framework to handle discontinuities, low-textured regions and fine depth variations. The work can be further extended so as to include aspects such as camera rotations, auto-calibration, image restoration etc.

References

[1] Z. Myles and N. V. Lobo. Recovering affine motion and defocus blur simultaneously. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(6):652–658, 1998.

[2] N. Ahuja and A. L. Abbot. Active stereo: Integrating disparity, vergence, focus, aperture and calibration for surface estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(10):1007–1029, 1993.

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

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

[5] J. Kim and T. Sikora. Confocal disparity estimation and recovery of pinhole image for real aperture stereo camera systems. In *IEEE International Conference Image Processing (ICIP 2007)*, volume 5.

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

[7] 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.

[8] A. Wrotniak. Depth of field tables for the Olympus C-5050/4040/3040Z. Retrieved July 4, 2009 from *http://www.digitaldiver.net/lib_docs/oly_dof.html*, Last updated June 15, 2003.

[9] A. Wrotniak. Depth of field tables for the Olympus C-30x0Z, C-40x0Z, and C-5050Z cameras. Retrieved July 4, 2009 from *http://www.wrotniak.net/photo/tech/dof-c5050.html*, Last updated October 7, 2006.

[10] J. Conrad. Depth of field in depth (2006). Retrieved July 4, 2009 from *http://www.largeformatphotography.info*, Last updated August 4, 2007.

[11] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1/2/3):7–42, 2002.

[12] M. Subbarao, T. Yuan, and J. Tyan. Integration of defocus and focus analysis with stereo for 3d shape recovery. In *Proceedings of SPIE*, volume 3204, pages 11–23, 1997.

[13] P. Felzenszwalb, D. Huttenlocher. Efficient belief propagation for early vision. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 1996)*, 1:261–268, 2004.

[14] Q. Yang, L. Wang, R, Yang, H. Stewenius and D. Nister. Stereo matching with color-weighted correlation, heiarchical belief propagation, and occlusion handling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(3):492–504, 2009.

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

[16] S. Kang, R. Szeliski and J. Chai. Handling occlusions in dense multi-view stereo. *Microsoft Technical Report MSR-TR-2001-80*, 2001.

[17] A. Klaus, M. Sormann and K. Karnar. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. *Proc. IEEE International Conference on Pattern Recognition (ICPR 2006)*, 3:15–18, 2006.

[18] J. Sun, N. Zheng and H. Shum. Stereo matching using belief propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(7):787–800, 2003.