

Variable Focus Video: Reconstructing Depth and Video for Dynamic Scenes

Nitesh Shroff¹ Ashok Veeraraghavan² Yuichi Taguchi³ Oncel Tuzel³ Amit Agrawal³ Rama Chellappa¹

¹University of Maryland, College Park ²Rice University ³Mitsubishi Electric Research Labs

{nshroff, rama}@umiacs.umd.edu vashok@rice.edu {taguchi, oncel, agrawal}@merl.com

Abstract

Traditional depth from defocus (DFD) algorithms assume that the camera and the scene are static during acquisition time. In this paper, we examine the effects of camera and scene motion on DFD algorithms. We show that, given accurate estimates of optical flow (OF), one can robustly warp the focal stack (FS) images to obtain a virtual static FS and apply traditional DFD algorithms on the static FS. Acquiring accurate OF in the presence of varying focal blur is a challenging task. We show how defocus blur variations cause inherent biases in the estimates of optical flow. We then show how to robustly handle these biases and compute accurate OF estimates in the presence of varying focal blur. This leads to an architecture and an algorithm that converts a traditional 30 fps video camera into a co-located 30 fps image and a range sensor. Further, the ability to extract image and range information allows us to render images with artistic depth-of field effects, both extending and reducing the depth of field of the captured images. We demonstrate experimental results on challenging scenes captured using a camera prototype.

1. Introduction

Cameras have become ubiquitous with billions of them present in various forms including cell-phone cameras, surveillance cameras, cameras installed on cars, at home, etc. In most cases, these cameras are passive devices just recording and saving video streams. This leaves most of the acquired data unprocessed. A major bottleneck towards automating the visual scene interpretation is the lack of 3D information that is crucial for scene understanding. The goal of this paper is to make traditional video cameras be able to extract meaningful 3D information by varying the focal distance during the acquisition of the video.

Variable focus makes cameras 3D: Most cameras are already equipped with features such as auto-focus, variable focal length and zoom, all of which require the focal distance to change. Unfortunately, this ability of the camera is significantly under-utilized. The ability to auto-focus is

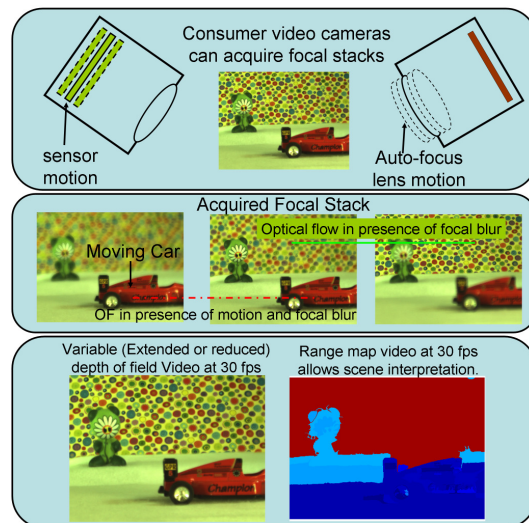


Figure 1. Consumer video cameras have the functionality to automatically focus. We can exploit this feature to acquire a focal stack and perform depth from defocus. In this paper, we show how this can be achieved in the presence of scene and/or camera motion, thereby turning an off-the shelf 30 fps video camera into a variable depth of field 30 fps camera and a 30 fps range sensor.

only used to obtain an image in which the subject of interest is in-focus. The fact that different subjects in the scene were in-focus at different focal distances enables the extraction of 3D information from these devices using depth from focus/defocus methods. We exploit this variable focus to simultaneously obtain an all-focus video and a depth map video.

Traditional depth from focus/defocus approaches require the scene to be static during the acquisition of multiple images. In this paper, we show how one can use optical flow information in order to remove this restriction and obtain 3D information in the presence of camera or scene motion. As a direct consequence of our ability to align multiple frames accurately, we are able to extract depth information and texture map at the native frame-rate of the camera, thereby converting a traditional 30 fps camera into a 30 fps extended depth-of-field (EDOF) camera and a 30 fps range sensor simultaneously.

1.1. Contributions

- DFD and EDOF imaging is extended to dynamic scenes by explicitly accounting for scene and camera motion. This enables a traditional video camera to acquire a 30 fps variable depth-of-field (DOF) video and act as a range sensor.
- The effects of (a) varying focal blur and (b) occlusions on the motion compensation problem are analyzed and an iterative refinement algorithm is designed that explicitly tackles such errors via reblurring.
- The design of a prototype is presented along with experimental results on several challenging scenes.

1.2. Prior Work

Depth From Defocus (DFD): Focus and defocus analysis for depth estimation has significant advantages over stereo and structure from motion as shown by [19, 14, 4], since they circumvent the correspondence problem. Another advantage of DFD over stereo is that only a single camera is required in DFD. Several algorithms for tackling DFD have been proposed [15, 7]. They minimize a cost function consisting of a data term and a spatial regularization term. The data term constrains how the texture blurs as a function of the unknown depth and the known focal distances. The regularization terms model spatial smoothness constraints within the depth map of the scene, typically by penalizing the L_2 cost of the depth difference of neighboring pixels. However, none of these methods can handle the DFD problem in the presence of scene or camera motion. In this paper, we present a framework to extend the DFD problem to scenarios with scene or camera motion.

Active Range Sensors: These sensors use an active pulse (e.g., laser, ultrasound) and either the direct time-of-flight or the phase difference between the emitted and received pulses to infer the scene depth. Structured light systems [17] use a pair of camera and projector to make the point correspondence problem in stereo easier. While the quality of the depth maps produced by such devices is usually high, they are expensive and require additional devices.

Variable Depth-of-Field (DOF) Imaging: The DOF of an imaging system can be extended by reducing the aperture. This however reduces the amount of light received by the camera, leading to low signal to noise ratio. On the other hand, if we increase the aperture, the sensor noise is reduced but at the cost of the decrease in DOF. We would ideally want a large DOF but with reduced sensor noise. Recently, several approaches have been proposed [5, 21, 11, 13, 9] to overcome this fundamental trade-off between the sensor noise and DOF. Veeraraghavan *et al.* [21] and Levin *et al.* [11] use a broadband mask at the aperture making

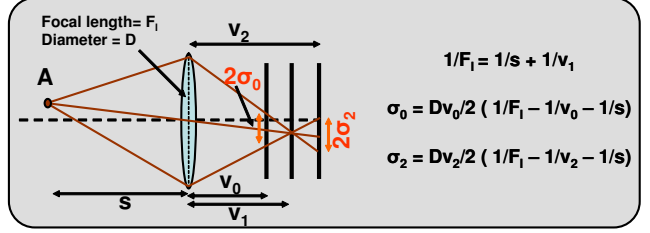


Figure 2. Depth from Defocus: The thin lens law states that when a scene point is out of focus, the blur radius of the out-of-focus scene point depends on the depth of the point apart from camera dependent parameters such as the focal length, aperture diameter and the sensor position. Given the camera parameters one can estimate depth from the blur radius.

the point spread function of blur better behaved. This allows computational deblurring extending the DOF. Dowski and Cathey [5] increase the DOF by inserting a cubic phase plate near the lens, while Nagahara *et al.* [13] increase the DOF by moving the sensor during the exposure duration. In both these methods the captured image is blurred, but the blur kernel is independent of depth and therefore can be deblurred using non-blind deblurring algorithms.

2. Basics and Limitations of DFD

A camera captures light from a scene and projects it on a sensor. Parts of the scene that are in focus are at depth s_0 given by the thin lens law:

$$\frac{1}{F_l} = \frac{1}{v} + \frac{1}{s_0}. \quad (1)$$

Here, F_l is the focal length of the lens and v is the distance between the camera lens and the sensor. Scene points that are at distance $s \neq s_0$ have a circle of confusion (blur) in the image plane. The distribution of light energy within this blur circle is referred to as the Point Spread Function (PSF). This PSF is a disc with its radius depending on the depth s of scene point:

$$\sigma = \frac{Dv}{2} \left(\frac{1}{F_l} - \frac{1}{v} - \frac{1}{s} \right), \quad (2)$$

where D is the lens aperture and σ is the radius of blur circle in the sensor plane. Figure 2 reviews this basic image formation in a camera. The scene point A with its depth s is in focus when the sensor is at distance v_1 from the lens. When the sensor is moved either towards the lens (v_0) or away from the lens (v_2), the rays from A form a circle of confusion with their radius given by equation 2. This dependence of blur on the depth of the scene point can be used as a cue to identify the depth of the scene, which is known as Depth from Defocus (DFD).

DFD in Static Scenes: Typical DFD methods capture a focal stack $\mathbb{F} = \{F_1, F_2, \dots, F_M\}$, consisting of a sequence of M

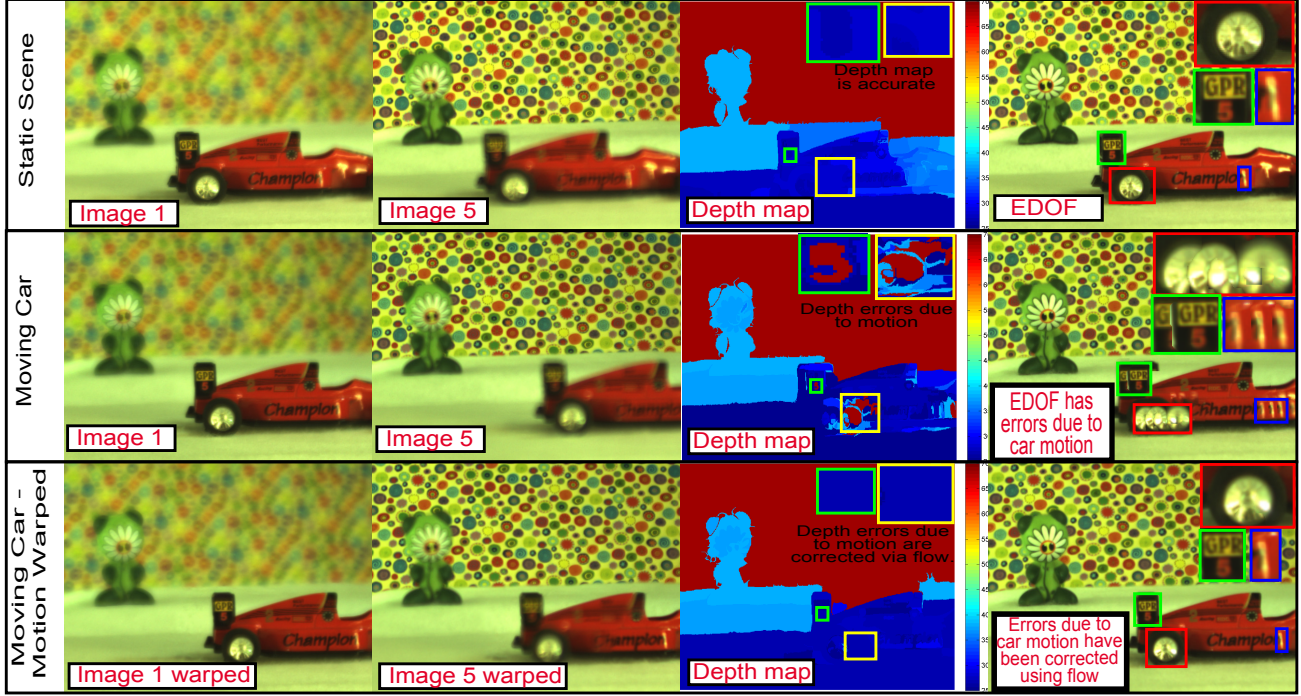


Figure 3. Effect of motion on DFD: Top row shows a static scene. A traditional DFD algorithm produces accurate depth and texture. Middle row shows a dynamic scene where the red car is moving to the right. This motion causes correspondence errors in DFD, resulting in depth and texture errors. Last row shows our algorithm applied to the scene in the middle row. Images shown here correspond to the virtual focal stack $\mathbb{Z}_3$ i.e., image 1 and image 5 have been warped to image 3 of the captured focal stack using optical flow. This allows the estimation of accurate depth and texture by correcting the correspondence errors. Depth values are in inches.

images F_j captured at various focus settings. Consider the focal stack of a static scene shown in the top row of Figure 3. First (F_1) and last (F_5) images of a focal stack of 5 images for a static scene are shown. In this static case, traditional DFD methods can produce an accurate depth map. The depth map can be used to obtain an extended depth-of-field (EDOF) image (also known as all-focus image) by combining the images in the focal stack.

Impact of Scene/Camera Motion: The basic assumption in traditional DFD methods is that the scene and camera are static. Consider the second row of Figure 3, where a focal stack is obtained as the red car moves to the right. The scene motion leads to correspondence errors in DFD, resulting in depth and texture errors. In EDOF images, the error appears as multiple copies of the moving object, while in the depth map, spurious depth edges are present on and around the moving object.

3. DFD in Dynamic Scenes

In this section, we describe how to adapt DFD methods to handle dynamic scenes. Let us assume that we have the motion information between a frame F_i and all the other frames within the focal stack $\mathbb{F}$. Intuitively, this information can be used to warp $\{F_j\}_{j=1}^{j=M}$ to F_i . This creates a “virtual”

focal stack $\mathbb{Z}_i$ that corresponds to the time instant i and has the properties of a static focal stack. The virtual focal stack corresponding to F_3 is shown in the bottom row (Moving Car - Motion Warped) of Figure 3. Since the motion has been compensated for in this focal stack $\mathbb{Z}_3$, the scene points are in correspondence. Depth and texture maps for time instant i can then be obtained from this virtual focal stack.

The central problem in motion estimation is the presence of varying defocus blur across the frames. Standard motion estimation techniques such as optical flow rely on the assumption of brightness constancy [10]:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \quad (3)$$

where (x, y) are the pixel grid points and t is the time instant. In Figure 4, we analyze the effect of varying defocus blur on the flow estimation. The two frames are focused at different depths, and the pixel marked as A has different intensities. This is because varying focal setting blurs A with different kernels, leading to the violation of brightness constancy. This violation induces spurious optical flow as shown in the bottom row of Figure 4— Flow before reblurring. The magnitude of optical flow estimated in the static regions of the scene is around 2.5 pixels.

As shown in the example, in the absence of any further in-

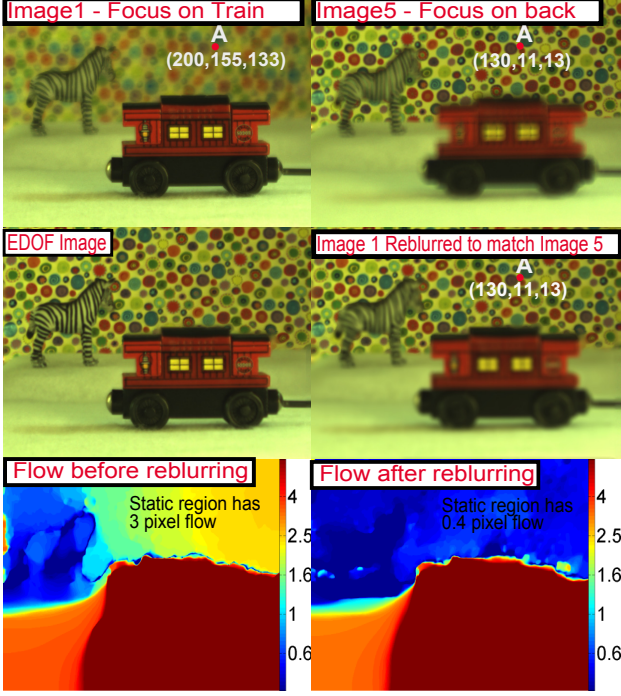


Figure 4. Flow estimation with varying blur. Top row shows two images of a focal stack with different blur levels. Observed intensities of pixel A violate the brightness constancy constraint for flow estimation. This violation introduces error in flow estimation. Texture and depth map (of image1) can be used to reblur image1 to match the blur level of image5. Third row shows the improvement in the estimated flow. This improvement is easily seen in static parts of the scene.

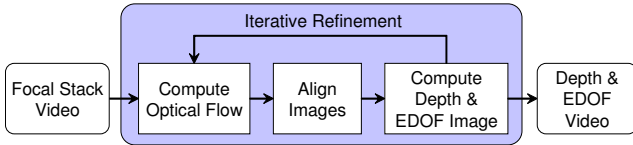


Figure 5. Block diagram of our reconstruction algorithm.

formation, computing optical flow with changing blur levels is a challenging task. However, given the depth map and the texture map of the scene at time instant i , the flow can be solved accurately using reblurring (Figure 4– Flow after reblurring). Likewise, given the flow information the depth and the texture map can be recovered using DFD. This leads to an iterative refinement algorithm for estimating the depth and the texture of a dynamic scene via stage-wise optimization, which is presented in the next section.

4. Iterative Reconstruction of Depth and Flow

The overview of our reconstruction algorithm is given in Figure 5. Given the focal stack video, we initially compute the optical flow between the frames. We then warp

the frames to generate a virtual focal stack for each time instant. The virtual focal stack is used to compute the depth and the texture maps using DFD. The results are refined by alternating between flow estimation and DFD. The following subsections describe each step of the algorithm.

4.1. Optical Flow

In the first iteration, the depth and the texture maps are unavailable. Hence, flow estimation can be approached by first computing a coarse optical flow and then refining via reblurring during the subsequent iterations.

Initial Flow: Several approaches to compute the coarse initial flow can be adopted each with its own pros and cons. Spurious optical flow between frames i and $i + 1$ could be computed in the first iteration and improved in subsequent iterations. This would have errors caused by the blur change as discussed earlier. Instead, we recover the initial flow by making an additional assumption — the flow has constant velocity within a period M of a focal stack. Frames from two consecutive focal stacks, (say i^{th} and $(M + i)^{th}$ frames of a video, have the same blur levels, hence satisfy brightness constancy. We compute the optical flow between i^{th} and $(M + i)^{th}$ frames and linearly interpolate the flow for the in-between frames. Although the initial flow is coarse due to the constant velocity assumption, it is refined via reblurring during the following iterations. Along with the constant velocity assumption, this approach also restricts the allowable motion between i^{th} and $(M + i)^{th}$ frames. In scenarios where this motion exceeds the allowable motion, the previous approach could be adopted.

Flow Given Depth and Texture: After the initial iteration of the DFD, the algorithm recovers a coarse estimation of the depth and the texture maps of the scene. Let D_i and T_i be the depth map and the texture map of the scene at time instant i respectively. This depth map D_i allows us to blur T_i with depth dependent kernels. This approach of matching the blur level of different images has earlier been utilized in shape from defocus [7, 18, 6]. Once the blur level of the two images are matched, brightness constancy is satisfied and hence optical flow can be computed with higher accuracy. Figure 4 shows the reblurred image F_5 and the computed optical flow between this reblurred image and F_1 .

Occlusion: Here, we discuss the occlusion problem in motion compensation. Consider two frames F_1 and F_4 of a focal stack $\mathbb{F}$ shown in the top row of Figure 6. In this dataset, red and blue markers rest on a turning table. Zoom in of the red marker shows it moving to the right from F_1 to F_4 . This motion occludes the pixels in the background region marked in blue. In F_1 , this occluded region is blurred while it is occluded in other images of this focal stack. This implies that the information regarding the focused background for this occluded region is unavailable.

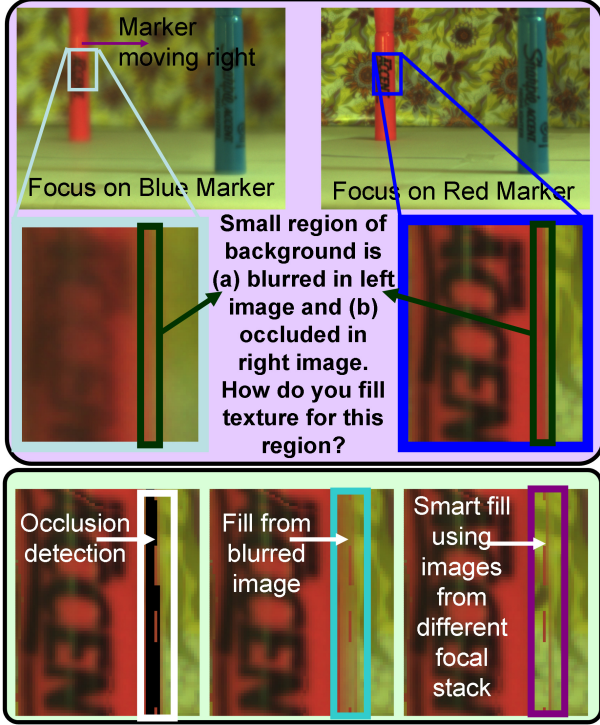


Figure 6. Occlusion handling. We detect occluded regions by checking consistency between the forward and backward optical flows between frames. If the occluded regions are static, we can fill them by using images from different focal stacks; otherwise we fill them using original (blurred) image in the current focal stack.

While warping F_4 to F_1 to compensate for the motion, this occlusion region needs to be detected and filled in. The occlusion region is detected by the inconsistency between the forward and backward optical flows. The pixel is assumed to be occluded if the forward-backward tracking results in a disparity greater than 2 pixels. The detected occlusion region is shown in the third row of Figure 6.

The information required to fill in the occluded region while warping F_4 to F_1 , can be retrieved by looking at other frames with the same blur level i.e., F_{4+Mj} where M is the length of a focal stack and j is any integer. In other words, the values for this region can be identified by looking at other frames with same blur level which have pixels from the background in the occluded region. This is shown in the bottom right of Figure 6. However, this might not be possible when the occluded pixels itself are moving. For instance, in the case of camera motion, even the pixels corresponding to the background move. In such cases, this region is filled up by copying corresponding pixels from the source frame, F_1 in this case which leads to visible artifacts on the object boundaries.

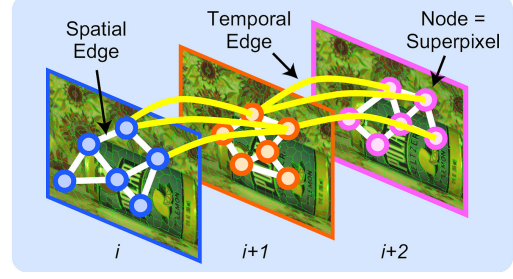


Figure 7. Pairwise MRF defined on superpixels. A spatial edge connects two nodes within an image if the superpixels share the boundary. A temporal edge connects two node across images if the superpixels have overlapped region after the alignment using optical flow.

4.2. Depth and Texture Given Optical Flow

After aligning the images in a focal stack using optical flow, we estimate depth maps $\mathbb{D} = \{D_1, D_2, \dots, D_M\}$ and textures $\mathbb{T} = \{T_1, T_2, \dots, T_M\}$ corresponding to each image F_i in the focal stack. We formulate the problem of depth estimation using a spatio-temporal Markov Random Field (MRF). As shown in Figure 7, we define an MRF using superpixels of the images as nodes and assume that each superpixel is represented by a front-parallel plane having a single depth value. These superpixels are perceptually meaningful over-segmentation of an image which conserves most of the structure in terms of color and shape [16]. It provides advantages in terms of reduced complexity by moving from pixel grid to superpixel map and is also representationally efficient as it can model smoothness between all pixels in a superpixel. Similar to [22, 20], superpixel segmentation for each frame is obtained using an iterative algorithm where we initialize superpixels as a regular grid and update their shapes based on the current estimate of shape and color distribution of each segment. This produces superpixels which are regularly shaped.

Given a set of superpixels $\mathbb{P}$ and a finite set of depth labels $\mathbb{S}$, the objective is to assign a depth label $s \in \mathbb{S}$ to each $p \in \mathbb{P}$. The energy function of the MRF is represented as

$$E(s) = \sum_{p \in \mathbb{P}} D_p(s_p) + \alpha \sum_{\{p,q\}} V_{pq}(s_p, s_q), \quad (4)$$

where α controls the degree of regularization.

To compute the data term $D_p(s_p)$, we assume that textures T_i are available for each focal stack image F_i . Initially these textures are obtained by applying photomontage algorithm [2] on the motion compensated focal stack $\mathbb{Z}_i$. After the first iteration we use textures computed using the previous estimate of the depth maps. Given the textures, the data term is computed by the sum of the squared difference between the observed superpixel and the reblurred superpixel

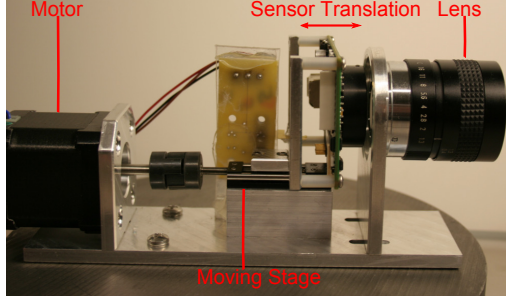


Figure 8. Our camera prototype.

for each depth level s_p . The PSF is assumed to be a disk kernel for reblurring the textures.

In our MRF formulation, we consider both spatial and temporal smoothness. Spatial smoothness penalizes changes in depth maps between two neighboring regions. We adopt the Potts model for spatial smoothness. Temporal smoothness, on the other hand, penalizes large change in depth between two consecutive frames. The smoothness term is computed as

$$V_{pq}(s_p, s_q) = w_{pq} T(s_p \neq s_q), \quad (5)$$

here, $T(\cdot)$ is the indicator function with value 1 if its argument is true and 0 if it is false. This term penalizes depth discontinuities between the neighboring superpixels p and q , and w_{pq} is a spatial/temporal weighting factor. The weight w_{pq} between two spatially neighboring superpixels p and q is determined by the similarity in the average color of the two superpixels: $w_{pq} = \exp\left(-\frac{\|I_p - I_q\|^2}{\tau}\right)$. Weights for the temporally neighboring superpixels are determined in the following way: Consider frames A and B . Let u be the optical flow between these two frames. Superpixel $p \in A$ is warped to frame B using the optical flow u . The overlap of p with the superpixels of frame B is then used as weights between temporal neighbors. We use graph cut algorithm [3] to minimize the energy function.

5. Experiments

In this section we discuss our prototype design and results on few challenging scenes.

5.1. Camera Prototype

Figure 8 shows our camera prototype. It translates a 1/3 inch Sony progressive scan CCD with a stepper motor which drives the National Aperture Inc., MicroMini Stage *MM-3-X*. The sensor moves around $2\mu m$ in each motor step. Thus, the distance between two consecutive sensor positions to capture the image can be varied at multiples of $2\mu m$. This can be controlled according to the DOF of the

scene to be captured and the number of images required per stack. In our experiments, we used 67 steps ($= 134\mu m$) between two consecutive sensor positions. A C-mount lens with a focal length of $12.5mm$ is attached to the fixed optical stage. While capturing the video, we move the sensor continuously and typically keep our exposure time to be $10ms$. A very small translation of the camera sensor covers a large amount of focal depths [13].

The camera translates along the optical axis of the lens with a constant speed. When it reaches the pre-specified extremum in one direction, the camera translates in the opposite direction. Hence it is capable of continuously capturing images at 30 fps. In most of our experiments, during one half period of sensor motion, the camera captures 5 images, which is the size of the focal stack.

Calibration: The magnification of the imaging system changes due to the translation of the sensor. In addition, the direction of the translation is not perfectly aligned with the central axis of the camera. These effects induce a planar projective transformation between the focal stack images. During calibration, we place a calibration grid in the scene and estimate the planar homographies between the focal stack images. These transformations are used to warp the images and processing is done in a canonical space (coordinate frame of F_3).

The relation between the scene depth and the blur kernel size is necessary for reblurring. During calibration, we place the calibration grid at multiple depths. At each depth, the focus is on the grid in one of the focal stack images, and the blur kernel sizes for all the other focal stack images are computed. The blur kernel size for a given depth is then estimated via linear interpolation.

5.2. Results

DFD/EDOF for Dynamic Scenes: We collected several challenging datasets of dynamic scenes with our prototype. Shown in Figure 9 are the results on two different datasets where we compare the DFD and EDof images before and after explicit motion compensation using our algorithm. The dataset on the left consists of two markers, with the orange marker moving to the right while the blue marker is moving to the left. The dataset on the right consists of a can rolling towards the camera. The first and third columns show the EDof images and the depth maps before motion compensation. Notice the errors in the depth map due to object motion. The second and fourth column shows the EDof images and depth map after motion compensation. Notice the significant increase in depth accuracy in the marker dataset and the reduction of motion artifacts in the EDof images of the can dataset. This estimated depth and texture maps enable us to render novel views. Figure 11 shows frames from marker dataset and the toy dataset ren-

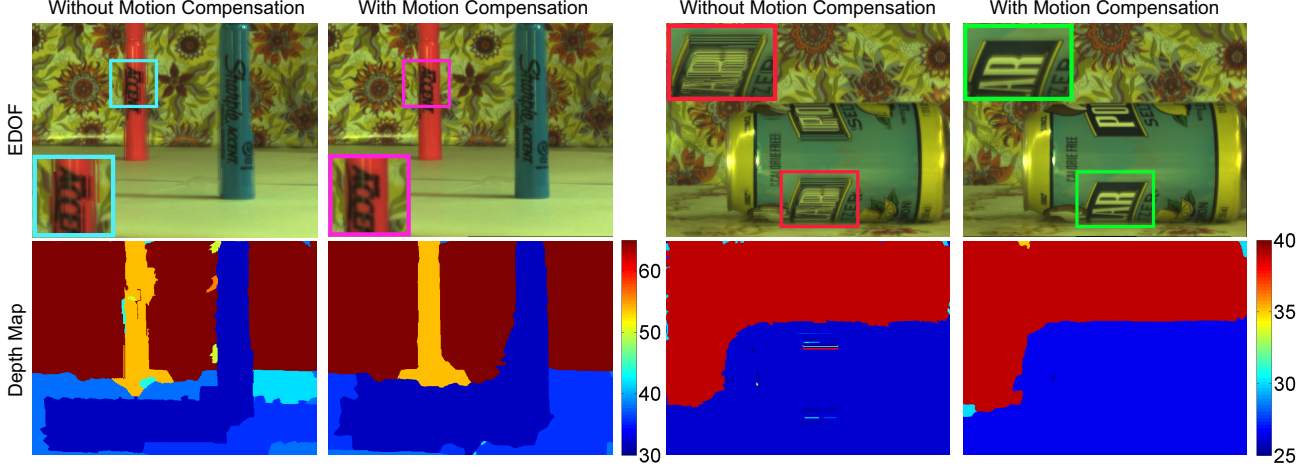


Figure 9. Comparison between EDOF images and depth maps generated with or without motion compensation. Traditional DFD without motion compensation fails to handle moving objects and produces artifacts at boundary regions. In the ‘rolling can’ example on the right, depth map artifacts can be seen corresponding to the texture map artifact highlighted for the result without motion compensation. Our algorithm correctly compensates for the object motion and generates high-quality depth maps and EDOF images. Depth values are in inches.

dered from 3 novel views. Accuracy of the retrieved boundaries for the objects can easily be visualized in these novel views. As the bottom surface is texture-less, no depth information is recovered on the surface.

Figure 10 shows a dataset of desktop where the coffee mug is being picked up. This dataset has 4 captured images in each focal stack. Two images from this dataset has been shown in the top row of the Figure. Left column corresponds to the near-focus image while the right column corresponds to the far-focus image. EDOF images and depth maps have been shown in the second and third rows respectively. It can be noted that EDOF images have all parts of the image as sharp. Depth map recovered for the textured regions are close to the ground truth values while few errors can be observed at the texture-less regions.

Effect of Refinement: Figure 12 shows the importance of the iterative optimization described in Section 4. This dataset consists of a toy scene in which the red car is moving to the right. Notice that both the depth map and the EDOF images have flow artifacts after the initial iteration (constant velocity flow assumption); whereas these artifacts are significantly reduced after second iteration via reblurring, leading to depth and texture maps as shown in the right column of Figure 12.

Reduced Depth of Field: Since our method computes both depth and texture simultaneously for dynamic scenes, this enables us to synthetically reduce the depth of field (DOF) of the acquired images. The depth quantization of the DFD algorithm is much finer than the DOF of each of the captured focal stack images. Shown in Figure 13 is a dataset

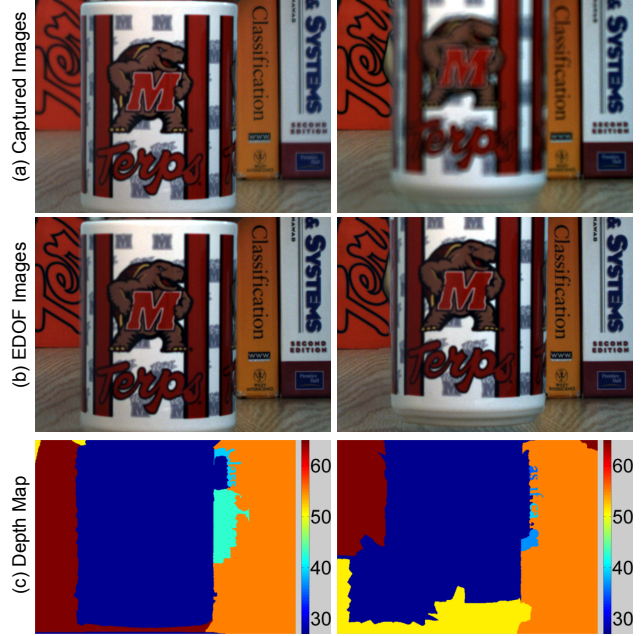


Figure 10. (a) Two images from a desktop scene with coffee mug being picked up with left one near-focused and right one far focused. (b) Corresponding EDOF images. (c) Corresponding Depth maps. Depth values shown here are in inches.

including a book on magic. This book was translated to the left and towards the camera. Shown in the first row are two of the captured images showing the translation and the DOF of the captured images. The corresponding EDOF images estimated by our algorithm are shown in the second row. The depth map at time t_1 is shown in the left of third

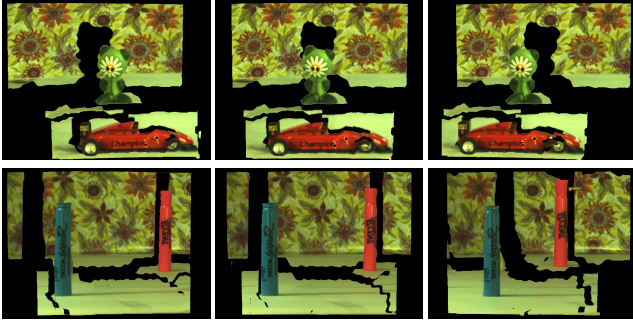


Figure 11. Novel views rendered using the depth and texture maps from different viewpoints.

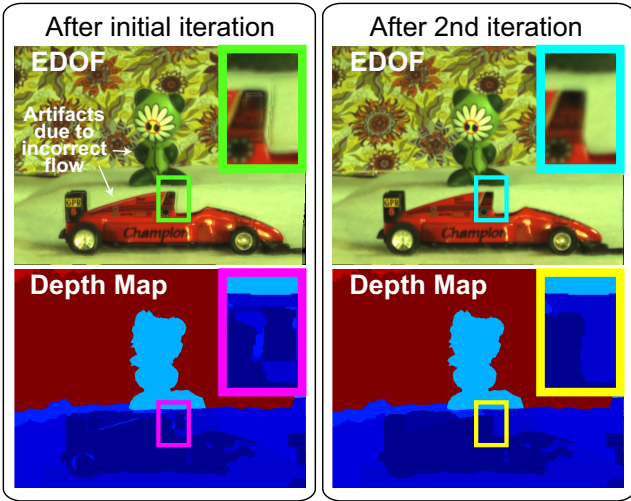


Figure 12. EDOF images and depth maps (left) after initial iteration and (right) after second iteration. Zoom-ups from each image has been shown. Note the boundary artifacts are reduced after the second iteration.

row. Finally, the depth map and the EDOF image are used in a depth dependent reblurring framework to produce the reduced DOF image shown in the right of third row. Notice that the DOF of this rendering is about 3 times smaller than the original DOF of the captured focal stack. Thus our algorithm can be used either to extend or reduce the DOF. Reducing the DOF is important for inexpensive and small cameras (like cellphone cameras) that have a very large DOF because of their small aperture size.

Camera Motion and Video results: We collected datasets with camera translation. EDOF and depth map videos were reconstructed. The effect of the camera motion is much better seen as video. This experimental result along with other video results and can be found at the project webpage ¹.

¹<http://www.umiacs.umd.edu/users/nshroff/MotionDFD.html>

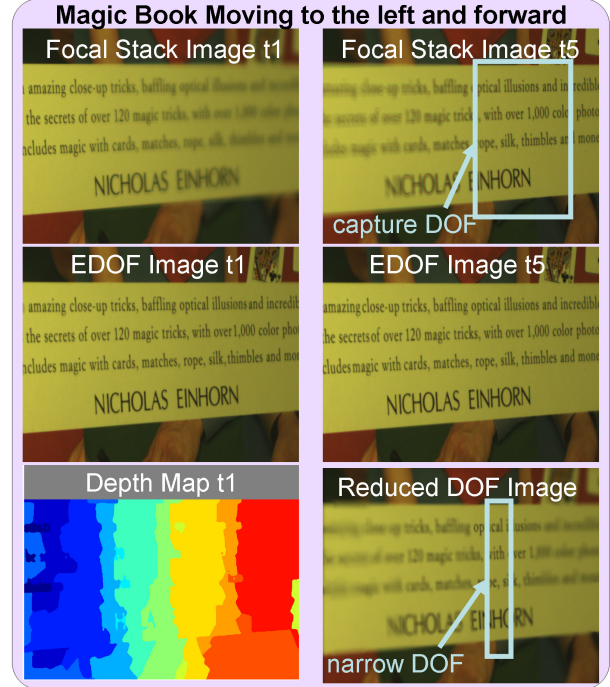


Figure 13. Variable depth-of-field (DOF) imaging. The estimated depth map enables us to compute extended DOF images and reduced DOF images.

6. Conclusion

In this paper, we have shown how advances in optical flow and motion estimation can be exploited towards extending the problem of DFD and EDOF imaging to dynamic scenes. The idea of performing explicit motion compensation is general and can be used with any of the available state-of-the-art DFD algorithms. Though in this paper, our implementation is based on a graph cut formulation, the ideas proposed here can be easily extended for other DFD approaches. Our analysis shows the importance of accurate flow estimation, describes some of the fundamental challenges in this problem due to focal blur, and suggests methods to handle these challenges.

Limitations: Our method suffers from the disadvantages of traditional DFD, such as the need for scene texture and a lens with large enough aperture to cause depth of field effects. It further inherits the limitation of dense motion estimation algorithms i.e., motion between successive frames of the video should not exceed the limitations of optical flow estimation algorithms, and motion component parallel to the axis of the camera cannot be estimated reliably. Further, the depth range in the scene should be small enough so that (a) the number of images in a focal stack is kept reasonable, and (b) the maximum blur of any observed scene point is manageable. The choice of the initialization scheme for the optical flow can pose limitation in terms of the maxi-

mum motion possible between two images with the same focal depth. This will be addressed in future work where initial optical flow could be solved by estimating motion for multiple frames together while compensating for blur in the same framework.

The main advantage of our approach is that this turns video cameras into high resolution variable depth of field and range sensors. Since most video cameras today have a built-in auto-focus feature that allows us to vary the focal distance during video acquisition, our approach could convert an off-the-shelf video camera into a range sensor also. Unfortunately, the firmwares available with cameras today do not provide external access to the auto-focus mechanism. The FrankenCamera [1] is an indication that the field is poised to change this lack of ability to control the hardware features available on the camera platform. We hope that this and several similar research directions [13, 12, 8] have the positive effect of convincing camera and lens manufacturers to allow external access to some of the in-built functionalities. Such external access, we believe, would turn traditional video cameras into powerful devices that are capable of improving our ability to do image understanding from the acquired videos.

Acknowledgments: We thank Jay Thornton and John Barnwell for their help and support. This work was mainly done while Nitesh Shroff and Ashok Veeraraghavan were at MERL with support from MERL. Nitesh Shroff and Rama Chellappa were partially supported by a MURI from the Army Research Office under the Grant W911NF-09-1-0383. Ashok Veeraraghavan was partially supported by NSF Grant NSF-IIS-1116718.

References

- [1] A. Adams, D. Jacobs, J. Dolson, M. Tico, K. Pulli, E. Talvala, B. Ajdin, D. Vaquero, H. Lensch, M. Horowitz, et al. The Frankencamera: an experimental platform for computational photography. In *SIGGRAPH*, 2010.
- [2] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *SIGGRAPH*, 2004.
- [3] Y. Boykov, O. Veksler, and R. Zabhi. Efficient Approximate Energy Minimization via Graph Cuts. *IEEE transactions on PAMI*, 20(12):1222–1239, Nov. 2001.
- [4] S. Chaudhuri and A. Rajagopalan. *Depth from defocus: a real aperture imaging approach*. Springer Verlag, 1999.
- [5] E. Dowski and W. Cathey. Extended depth of field through wave-front coding. *Applied Optics*, 34:1859–1866, 1995.
- [6] P. Favaro. Recovering thin structures via nonlocal-means regularization with application to depth from defocus. In *CVPR*, 2010.
- [7] P. Favaro and S. Soatto. A geometric approach to shape from defocus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):406–417, 2005.
- [8] S. Hasinoff and K. Kutulakos. Confocal stereo. *International journal of computer vision*, 81(1):82–104, 2009.
- [9] S. W. Hasinoff, K. N. Kutulakos, F. Durand, and W. T. Freeman. Time-constrained photography. In *ICCV*, 2009.
- [10] B. Horn and B. Schunck. Determining optical flow. *Artificial intelligence*, 17:185–203, 1981.
- [11] A. Levin, R. Fergus, F. Durand, and B. Freeman. Image and depth from a conventional camera with a coded aperture. *SIGGRAPH*, 2007.
- [12] A. Levin, P. Sand, T. Cho, F. Durand, and W. Freeman. Motion-invariant photography. In *SIGGRAPH*, 2008.
- [13] H. Nagahara, S. Kuthirummel, C. Zhou, and S. K. Nayar. Flexible Depth of Field Photography. In *ECCV*, 2008.
- [14] A. Pentland, S. Scherrock, T. Darrell, and B. Girod. Simple range cameras based on focal error. *Journal of the Optical Society of America A*, 11(11):2925–2934, 1994.
- [15] A. N. Rajagopalan and S. Chaudhuri. Optimal recovery of depth from defocused images using an MRF model. In *ICCV*, 1998.
- [16] X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, 2003.
- [17] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *CVPR*, 2003.
- [18] M. Subbarao and G. Surya. Application of spatial-domain convolution/deconvolution transform for determining distance from image defocus. In *SPIE*, 1992.
- [19] M. Subbarao and G. Surya. Depth from defocus: a spatial domain approach. *International Journal of Computer Vision*, 13(3):271–294, 1994.
- [20] Y. Taguchi, B. Wilburn, and L. Zitnick. Stereo reconstruction with mixed pixels using adaptive over-segmentation. In *CVPR*, 2008.
- [21] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *SIGGRAPH*, 2007.
- [22] C. Zitnick, N. Jojic, and S. Kang. Consistent segmentation for optical flow estimation. In *ICCV*, 2005.