Focal Stack Representation and Focus Manipulation

Parikshit Sakurikar and P. J. Narayanan
Center for Visual Information Technology - Kohli Center on Intelligent Systems,
International Institute of Information Technology - Hyderabad, India
{parikshit.sakurikar@research.,pjn@}iiit.ac.in

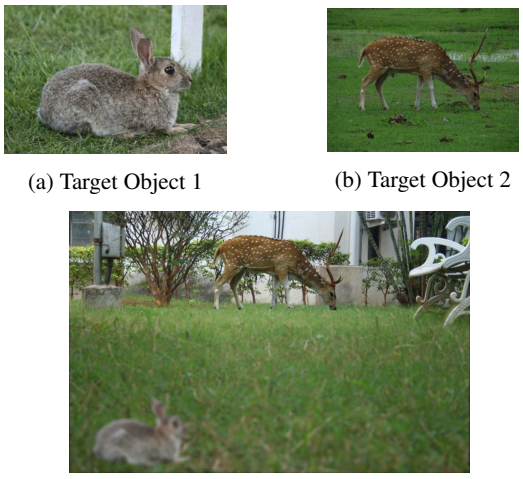
Abstract

Focus, depth-of-field, and defocus are important elements that portray the aesthetic emphasis in a good photograph. The ability to manipulate the focus after capture provides useful creative control to photographers. Capturing focal stacks - multiple images with small change in focus setting - of static scenes is relatively easy with modern cameras. We propose a compact representation for focal stacks using an all-in-focus image, a focal-slice index map and pair-wise defocus blur parameters. Using our representation, we show reconstruction of images with different focus effects including extended focus, multiple focus, and scene synthesis with natural focus effects. A user study shows high acceptability of the synthesized images compared to real ones. The compact and powerful representation of focal stacks makes them suitable for handling by image editing tools in order to provide flexible focus manipulation.

1. Introduction and Related Work

An image captured with an aperture camera is a spatio-photometric slice of the plenoptic function. The focus, aperture, and exposure settings determine the characteristics of the captured slice. The focal length governs the field-of-view, the focus distance determines the regions that appear sharp, the aperture size fixes the depth-of-field around the plane of focus as well as the light entering the camera, and the shutter speed affects the brightness by changing the integration time. An ideal pin-hole camera image of a well-lit scene captures the maximal spatio-photometric slice with every object appearing sharp and clear. Computer Vision applications typically prefer such maximal slices. Artistic photography, on the other hand, relies heavily on capturing finite spatio-photometric slices. Focus, depth-of-field and defocus blur are used by photographers for creative communication and emphasis.

Post-capture manipulation of spatio-photometric slices gives additional creative options to photographers. Bracketing in exposure has been used to control the dynamic range of captured scenes. Images from a panning camera allow expansion of the field of view. Multiview capture can lead to viewpoint interpolation. Similarly, focal stacks that capture a scene with different focus distances and a narrow depth of field (DoF) provide the basic input for post-capture manipulation of focus. Focal stacks have been used successfully to compute all-in-focus (AiF) images of the scene [7, 12], using linear filtering of focal slices, 3D filtering, focal sweep aggregation etc. Wide aperture images also improve the overall SNR of the AiF image [5]. Focal stacks have been shown to be useful for post-capture control over the focus and depth-of-field. Methods such as [6, 11, 12] describe focus sweep imaging, half sweep imaging and geometric composition models to generate composite images from the focal slices. We propose to build a computationally inexpensive representation of focal stacks that can comprehen-



(a) Target Object 1 (b) Target Object 2

(c) Synthetic Far-Focus Image

Figure 1: A naturally focused synthetic scene. Segmented target objects are added to different focal depths by pasting them into the $\mathcal{F}$ image with new index labels. Reconstruction of the focal stack gives naturally focused synthetic images.

2327-0985/17 $31.00 © 2017 IEEE
DOI 10.1109/ACPR.2017.102

sively manipulate focus aspects like focus distance, defocus blur and DoF.

We model a focal stack from a generative perspective and suggest an efficient representation based on it. The model enables several post-capture focus manipulations including composition of scenes with realistic focus effects. Our model has three components: (a) an image consisting of the in-focus version of each pixel (which is essentially the AiF image), (b) a slice index map for each pixel to indicate which slice it came from, and (c) the defocus blur model parameters π_{ij} that can be used to reconstruct a pixel in slice j from its in-focus slice i. The original slices can be reconstructed with high fidelity using our model. We also show how synthesis of focus-effects like extended DoF, multifocus images, image synthesis with focus-effects etc., are easy using this representation. A user study conducted by us shows high acceptability of images synthesized using this model as compared to captured ones. Our representation is very compact and enables easy manipulation of focal stacks with 50 or more slices.

The contributions of this paper include (i) a generative model for focal stacks and an efficient representation based on it, (ii) a fast and accurate algorithm to recover the representation from a focal stack, and (iii) demonstration of post-capture focus manipulation including extended focus, multifocus photos and composition of new scenes with natural focus effects using the representation. Several researchers have computed the all-in-focus image from a focal stack by picking pixels from specific slices in the past. However, the idea of using it as the basis of a compact representation in conjunction with a generative model for the individual slices is a simple but powerful idea we introduce here. We envisage focus manipulations to be an integral part of on-device image editing toolkits in the near future. Our compact representation can be used by them natively to directly perform versatile post-capture focus manipulation. Figure 1 gives an example of scene composition with focus effects that we facilitate.

2. Focal Stack Representation

A focal stack $\mathcal{G}$ is a sequence of k images (called focal slices) $G_i, 1 \leq i \leq k$. Each slice is captured with a progressively varying focal distance but a fixed finite aperture opening. Ideally, a pixel appears in sharp focus in one and only one slice and appears blurred due to defocus in other slices. Prior researchers stored focal stacks in full as k images. This needs large storage and processing time for deep stacks.

The in-focus pixels combined with the right blur model can generate the defocused pixels in other slices. The blur parameter π_{ij} is used to generate the blurred version of a pixel in slice j given its in-focus version in slice i. Thus, any focal slice can be reconstructed synthetically given (a) the in-focus versions of all pixels, (b) the index of the slice in which each is in focus, and (c) the blur parameters. The collection of all in-focus pixels is the all-in-focus (AiF) image. This is the basis of our representation consisting of the AiF image $\mathcal{F}$, the slice index map $\mathcal{I}$, and the defocus parameters π_{ij} for pairs of focal slices.

A defocused pixel p in a target focal slice j is synthesized as

$$G_j(p) = \pi_{ij} * \mathcal{F}(p), \tag{1}$$

where $i = \mathcal{I}(p)$ is its in-focus slice, π_{ij} is the defocus parameter between the focal slices i and j and $*$ is a general operator based on the defocus model. We follow the practice of modeling defocus blur as a convolution with a zero-mean Gaussian with variance σ^2. Prior methods that estimate AiF image from a focal stack identify the in-focus pixel and its slice. However, none have explicitly modeled the focal stack using a simple but powerful model like this, which also results in significant compression. The AiF image $\mathcal{F}$ is a single image, the slice index $\mathcal{I}$ is an image with a single channel, and the blur parameters π_{ij} consist of k^2 σ values. Our representation thus needs $O(N^2 + k^2)$ space compared to $O(kN^2)$ needed to store all slices. This results in significant savings, especially as k increases. This is important to make focal stacks as native image objects.

We use an MRF labeling scheme with a robust focus measure to estimate $\mathcal{F}$ and $\mathcal{I}$. We evaluate the blur parameters empirically for each slice pair, assuming Gaussian blurring. We now describe the process of estimating the in-focus pixels and blur parameters.

2.1. Estimating In-Focus Pixels

Several methods exist to estimate the in-focus pixels from the focal slices. Usually, the response of a focus measure at each pixel is used to identify its in-focus slice. There are various sharpness based focus measures which can be used [1, 9, 13]. However, segmenting in-focus pixels from different slices of a focal stack based only on a focus measure lead to noisy estimates [16], which results in sub-optimal reconstruction. Smoothness enforcing optimization such as Cost-Volume Filtering [14] or MRF labeling can be used to improve in-focus segmentation. Iterative optimization methods such as [8, 15] estimate in-focus pixels by solving a large set of equations using Gauss-Seidel methods. We found from experimentation that iterative methods do not scale well to deep focal stacks with large k.

We use a 2D MRF framework [2] to label each pixel to its in-focus slice using a structure tensor based focus measure (recently shown by Boshtayeva *et al*. [1] to be a superior focus measure). Our experiments showed that 3D MRF methods do not scale to many layered stacks. We also experiment with Cost-Volume filtering, which outperformed 2D MRF for stereo labeling [14]. We find that MRF and CVF performed equally well on using a variance-based focus mea-

(a) Halo Artifacts

(b) Halo Suppressed

Figure 2: Halo artifacts can be suppressed by updating the index map $\mathcal{I}$ such that there is no double counting of rays.

sure. However, our choice of MRF with the structure-tensor focus measure produces best in-focus slice labeling across multiple focal stacks.

In-focus slice labeling We build a standard MRF over each pixel across the focal stack. Each node (pixel) is connected to its 4-neighbors. A measure based on the trace of the structure tensor is used as the unary potential at each pixel.

The structure tensor for a pixel p in the focal slice i is defined as $J_i(p) = \sum_r w_r J_i^0[p-r]$, where w_r is a normalized weight and r denotes a radius around the pixel, with

$$J_i^0(p) = \begin{bmatrix} G_i^x(p).G_i^x(p) & G_i^x(p).G_i^y(p) \\ G_i^x(p).G_i^y(p) & G_i^y(p).G_i^y(p) \end{bmatrix}. \quad (2)$$

Here G_i^x, indicates the horizontal gradient of G_i and G_i^y is the vertical gradient. The data term $D_i(p)$ for pixel p and label i is inversely proportional to the trace of the structure tensor at p. We use $D_i(p) = -trace(J_i(p))$ as the unary potential.

A Potts model is used for pair-wise potentials. The MRF is optimized using an alpha-expansion algorithm. The optimal label indicates the slice at which each pixel is in focus. The all-in-focus image $\mathcal{F}$ and the slice index $\mathcal{I}$ are generated directly from this label (Figure 3).

Halo artifacts Segmentation of focal slices to in-focus pixels based on pixel sharpness usually results in Halo artifacts that adversely affect slice reconstruction. Halo is the result of double counting of rays at different sensor locations [6]. The same object contributes as sharp in both its focused as well as nearby defocused slices due to intensity spreading near the depth edges. Figure 2a shows the undesirable impact of haloing.

A precise knowledge of the lens geometry and sensor positions are required to eliminate halos. Since we use Magic Lantern [10] to capture focal stacks, we have an approximate yet adequate description of sensor positions for each focal slice. We implement de-haloing as a post-processing step over the estimated in-focus pixels using the method by Jacobs *et al*. [6]. Results of de-haloing can be seen in Figure 2b. De-haloing removes the edgy artifacts caused by halos, but causes blurring near depth edges [6] (Figure 2b). This is more significant with narrower focus zones. This occurs because at the pixels where the halo is removed, there is no focal slice which captures the true scene content without the contribution of haloing. Thus the slice closest to the true depth distribution is chosen in which the pixel is slightly defocused, instead of being perfectly in-focus.

2.2. Estimating Defocus Model Parameters

The defocus parameters π_{ij}, $i, j \in [1, k]\}$ are directly related to the geometry of the lens and the sensor. There are $k(k-1)$ possible combinations of defocus parameters for a focal stack with k slices. We use a zero-mean Gaussian with standard deviation σ to model the defocus blur, with σ increasing with the distance from the in-focus slice. The Pill Box model has also been analyzed earlier [4, 9].

We compute π_{ij} empirically by analyzing the in-focus pixels in slice i with corresponding pixels in a captured slice j, given $\mathcal{F}$ and $\mathcal{I}$. We first identify the regions of in-focus pixels in each slice from $\mathcal{I}$. We eliminate small regions from the computations.

A binary search in blur-radius space is performed to identify the σ that best blurs the in-focus pixels to its defocused versions for each pair i, j. For each pair, we search σ values from 0.01 to a high value of 25% of the image size by recursively halving the interval based on the errors. The in-focus regions are blurred using the current σ value and compared with its defocused counterparts. The σ value that provides the best MSE between the two regions is chosen as σ_{ij}.

This method is general and works for all scenarios, even when no estimates of focal distance are available for each slice of the stack. This method is superior to geometric models as the real defocus quality of a lens - which might show amplified or attenuated blurring compared to geometric blur radii - can be captured by empirical blur estimation.

3. Evaluation of the Model

The true strength of the compact representation is its ability to create all focus effects that can be generated using a full focal stack with high quality. The goal of our method is to show that such a simple and compute inexpensive representation is more than sufficient to perform a variety of post-capture focus manipulations. Thus, we perform quantitative and qualitative evaluation of reconstructing each slice of the focal stack using our model against the ground truth captured focal slices.

Quantitative evaluation The difference between the model-based reconstruction and captured images gives the quality of reconstruction. Given the AiF image $\mathcal{F}$, the index

(a) Grass (b) Leaning Tower (c) Soldiers (d) Textbook

Figure 3: All-in-Focus Images generated by cumulating the in-focus pixels from all focal slices.

map $\mathcal{I}$ and the set of defocus parameters π_{ij}, we reconstruct the full focal stack by applying the blur model to $\mathcal{F}$. The reconstruction PSNR for each focal slice is an indication of its quality.

To generate a focal slice G_j, we create blur map for it. The pixels which are labeled j in the index map $\mathcal{I}$ are given a blur radius of 0 in the blur map. This results in picking these pixel values directly from $\mathcal{F}$. Other pixels are assigned defocus radii based on their index map value. A pixel p of the blur map gets a value π_{ij} where $i = \mathcal{I}(p)$ gives the slice in which it is in focus.

We smooth the per-pixel blur radius map to ensure that pixels at differing depth-edges do not get drastically differing blur radii, to avoid artifacts in the reconstructed image. The smoothed blur map is applied on $\mathcal{F}$ to synthesize G_j. Each pixel of $\mathcal{F}$ is blurred using a Gaussian with its σ from the blur map to get its value in G_j. We compute the PSNR between reconstructed and captured focal slices. Figure 4 shows the reconstruction PSNR for 5 focal stacks. The average PSNR across slices and its range can be seen in the figure. The average PSNR ranges from $37dB$ to $42dB$ for these datasets, indicating high quality reconstruction.

We also evaluate the PSNR limited to the depth-boundaries of the scene by looking at the errors in reconstructed focal slices near depth edges, where adjacent pixels have been labeled to different focal slices. Figure 5 shows the depth-edge reconstruction PSNR for different slices of the Grass dataset. The average PSNR is about $45dB$, again indicating very good reconstruction at depth boundaries. For reconstruction PSNR limited to depth boundaries, we get an average PSNR $> 40dB$ across all our focal stacks.

Qualitative evaluation We also perform a subjective user evaluation in which users are shown side-by-side images of a focal slice where one is captured by the camera and the other is reconstructed at the same focus distance using our method. The users were asked to select either one of the two as the real image or state that both images look real. We received over 150 evaluations, each containing five image pairs, thereby accumulating a total of 750 responses. Users were differentiated into three categories of Novice, Basic and Expert based on their familiarity with SLR/DSLR cameras, with experts having used DSLRs for over 1 year.

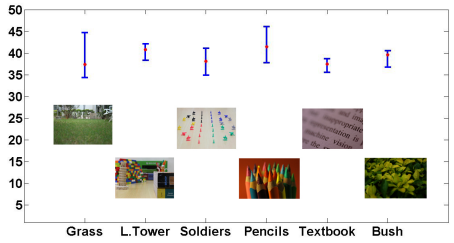


Figure 4: PSNR for focal slice Reconstruction. For each focal stack, the range of PSNR values for all focal slices is shown in blue and the average reconstruction PSNR is shown as the red mark.

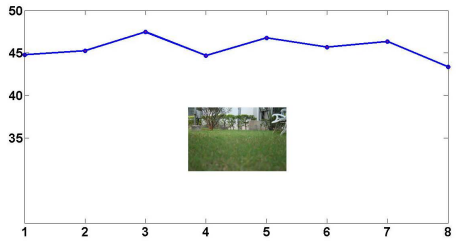


Figure 5: Reconstruction PSNR near depth-edges for the Grass focal stack. Average PSNR = $45.54dB$

Over 70% of the responses either considered the synthesized image to be real or thought both were real. This suggests that in a majority of instances, our image is either preferred or indistinguishable from the captured image for most subjects. There was a small but surprising spread across expertise categories: 68% of the Novice users, 67% of basic users, and 72% of expert users favored our reconstructed version or had no choice.

4. Applications and Results

We demonstrate segmentation of in-focus pixels, estimation of blur kernels and focus and scene manipulation on a variety of focal stacks ranging from shallow to deep, and of different types of scenes with differing complexities. We capture several focal stacks using Magic Lantern on a Canon EOS 70D and a Canon EOS 1100D. All operations

are performed on RAW images. We first align each focal stack in order to eliminate pixel misalignment arising from magnification due to focus/defocus. We use the enhanced correlation coefficient maximization approach [3] to align the focal layers. We estimate the blur radius for each pair of sensor positions empirically using the $\mathcal{F}$ image and the index map $\mathcal{I}$. We plan to release several focal stacks that we have captured for others to use.

4.1. In-focus Pixel Segmentation

We extract the in-focus pixels for various focal stacks using our multilabel MRF method. Fig. 6 shows the in-focus pixels for the Grass stack with simple near to far progression, and the Bush and Flowers stacks having complicated pixel distribution pixels in consequent focal layers.

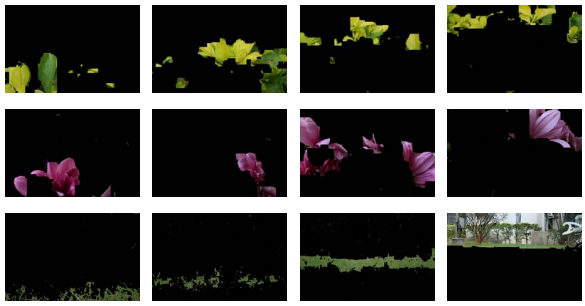


Figure 6: In-focus pixels corresponding to labels $(2, 4, 7, 9)$ from the Bush (top), Flower $(3, 5, 8, 11)$ (middle), and Grass $(1, 4, 6, 8)$ (bottom) focal stacks.

4.2. Extended Focus Images

An extended focus image is a generalization with slices a to $a + b$ in focus and others blurred in a photorealistic manner. Picking the in-focus pixels from slices a to $a + b$ and blurring the rest with suitable π_{ij} kernels is a naive way to generate the extended focus image. It will, however, show artifacts at slice boundaries in the generated image. We use a two-step approach for better quality.

In the first step, a blur radius map is created with a value for each pixel (x, y) of the target image, based on its label $\mathcal{I}(x, y)$ in the index map. Labels in the range $[a, a+b]$ get a radius of 0. For the remaining pixels having labels i, a blur radii r is assigned given by

$$r = \min(\pi_{(a,i)}, \pi_{(a+b,i)}). \tag{3}$$

thereby blurring from a or $a + b$, whichever is closer.

The second step smoothes the per-pixel blur map to avoid visible boundaries where layers meet. A small Gaussian smoothing filter is applied to the blur map of each pixel to soften the edges. The smoothed blur map is applied to $\mathcal{F}$ to yield the synthesize extended DoF image. Figs. 7e, 7f show examples of extended focus images.

4.3. Multifocus Image Generation

Our model allows arbitrary slices to be in focus. This can be used for creative composition of images with a focus distribution which cannot be directly captured by an aperture camera. Consider the case in which a user indicates an arbitrary subset S of the slices to be in focus.

We construct the blur radius map for the target image as before. A pixel p with a slice index $\mathcal{I}(p) \in S$ gets a radius of 0. Pixels with other index values are assigned a blur radius corresponding to the closest slice in S. That is, the blur radius of a pixel p with $\mathcal{I}(p) = j$ is given by $\min_{i \in S}(\sigma_{ij})$. Each slice is now assigned blur radius based on its distance from the closest focused slice. The blur map is then smoothed using a small Gaussian to avoid edge artifacts. The blur map is then applied to AiF image $\mathcal{F}$ to yield the the multifocus image. Figs. 7g, 7h show examples of multifocus images.

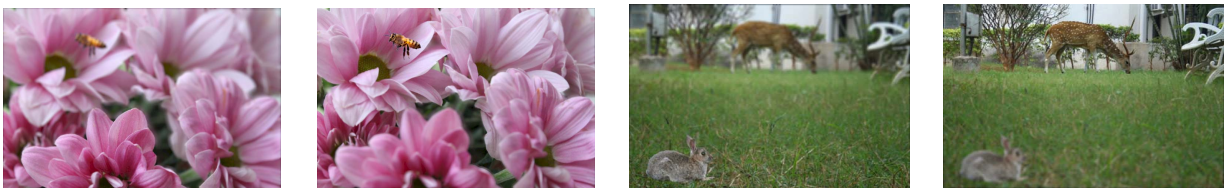
4.4. Scene Manipulation with Natural Focus

Our focal stack representation allows other direct modifications, including introducing a new object into the scene with photorealistic focus effects. Such effects are novel to our representation of segmented in-focus pixels as they involve addition of objects and updation of individual pixel labels.

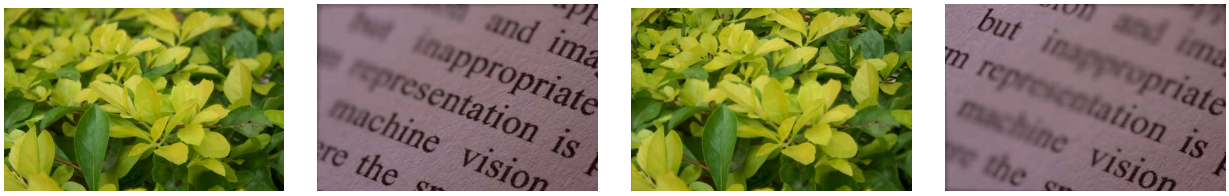
A cleanly segmented image of an object in sharp focus can be added to a focal stack at a slice indicated by the user. To do this, the image is first introduced into the AiF image $\mathcal{F}$ at a user selected location and scale using a standard image compositing technique. The index map $\mathcal{I}$ of pixels that are overwritten by the new objects are set to the user-specified slice number of the object. This results in a modified focal stack representation. All previously explained focus synthesis operations – such as reconstrucing a specific slice, extended or multifocus images, etc. – can be performed on the modified representation to get high quality images. Figures 7a, 7b, 7c, 7d are examples of scene manipulation with natural focus. The bee, the bunny, and the deer in those images are introduced into the Flower and Grass datasets using this method.

4.5. Practical Details

We use a medium range desktop computer for our experiments and use code written in Matlab except for MRF labeling which is performed using C++ and OpenCV. Our overall pipeline from capturing a focal stack to computing the ($\mathcal{F}$,$\mathcal{I}$,π_{ij}) representation takes about 40-50 seconds for a 15 sliced high resolution focal stack. Computing any novel refocused image 2-3 seconds per image.



(a) Synthetic Near Focus (b) Synthetic Far Focus (c) Synthetic Near Focus (d) Synthetic Far Focus



(e) Focal layers 1 to 5 in focus (f) Focal layers 1 to 8 in focus (g) Layers (1,2,10,11) in focus (h) Layers (1,2,3,17,18) in focus

Figure 7: Comprehensive Focus Manipulation that we faciliate using $\mathcal{F}$, $\mathcal{I}$ and π. First Row: Scene manipulation with natural focus. In the first two images, an image of a bee has been added to focal layer 9 in the stack and naturally defocused near and far focused images are shown. Second Row: Extended Focus Images and Multifocus Images

5. Conclusion and Future Work

We presented an effective model and a compact representation of focal stacks in this paper. The representation is easy to compute and provides high quality focal stack reconstruction and manipulation. While more sophisticated models for focal stack representation can be built to compensate for lens defocus effects, bokeh effects, multiple in-focus slices for each pixel etc, ours is the first study which shows that a simple and generic model is quite robust and useful for good quality focal stack manipulation. A compact and efficient representation such as ours will go a long way in facilitating the manipulation of focal stacks on standard image editing tools and applications built for portable mobile devices.

References

[1] M. Boshtayeva, D. Hafner, and J. Weickert. A focus fusion framework with anisotropic depth map smoothing. *Pattern Recognition*, 48(11):3310 – 3323, 2015. 2

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23, 2001. 2

[3] G. Evangelidis and E. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865, 2008. 5

[4] S. Hasinoff. *Variable-Aperture Photography*. PhD thesis, University of Toronto, 2008. 3

[5] S. W. Hasinoff, K. N. Kutulakos, F. Durand, and W. T. Freeman. Time-constrained photography. In *IEEE International Conference on Computer Vision*, pages 333–340, 2009. 1

[6] D. E. Jacobs, J. Baek, and M. Levoy. Focal stack compositing for depth of field control. *Stanford Computer Graphics Laboratory Technical Report*, 1, 2012. 1, 3

[7] A. Kubota, K. Aizawa, and T. Chen. Reconstructing dense light field from array of multifocus images for novel view synthesis. *IEEE Transactions on Image Processing*, 16(1):269–279, 2007. 1

[8] A. Kubota, K. Takahashi, K. Aizawa, and T. Chen. All-focused light field rendering. In *Proceedings of the Fifteenth Eurographics conference on Rendering Techniques*, EGSR'04, pages 235–242, 2004. 2

[9] A. Kumar and N. Ahuja. A generative focus measure with application to omnifocus imaging. In *IEEE International Conference on Computational Photography*, 2013. 2, 3

[10] Magic lantern. http://magiclantern.fm/. 3

[11] S. Matsui, H. Nagahara, and R. I. Taniguchi. Half-sweep imaging for depth from defocus. In *Advances in Image and Video Technology*, pages 335–347. Springer, 2012. 1

[12] H. Nagahara, S. Kuthirummal, C. Zhou, and S. K. Nayar. Flexible depth of field photography. In *European Conference on Computer Vision*, pages 60–73. 2008. 1

[13] S. Pertuz, D. Puig, and M. A. Garcia. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, 46(5):1415 – 1432, 2013. 2

[14] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *2011 IEEE Conference onComputer Vision and Pattern Recognition (CVPR)*. 2

[15] P. Sakurikar and P. Narayanan. Dense view interpolation on mobile devices using focal stacks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 138–143, 2014. 2

[16] C. Zhou, D. Miau, and S. K. Nayar. Focal sweep camera for space-time refocusing. *Technical Report, Department of Computer Science, Columbia University*, CUCS-021-12, 2012. 2