Focal Sweep Camera for Space-Time Refocusing

Changyin Zhou · Daniel Miau · Shree K. Nayar

Abstract A conventional camera has a limited depth of field (DOF), which often results in defocus blur and loss of image detail. The technique of image refocusing allows a user to interactively change the plane of focus and DOF of an image after it is captured. One way to achieve refocusing is to capture the entire light field. But this requires a significant compromise of spatial resolution. This is because of the dimensionality gap - the captured information (a light field) is 4-D, while the information required for refocusing (a focal stack) is only 3-D.

In this paper, we present an imaging system that directly captures a focal stack by physically sweeping the focal plane. We first describe how to sweep the focal plane so that the aggregate DOF of the focal stack covers the entire desired depth range without gaps or overlaps. Since the focal stack is captured in a duration of time when scene objects can move, we refer to the captured focal stack as a *duration focal stack*. We then propose an algorithm for computing a *space-time in-focus index map* from the focal stack, which represents the time at which each pixel is best focused. The algorithm is designed to enable a seamless refocusing experience, even for textureless regions and at depth discontinuities.

We have implemented two prototype focal-sweep cameras and captured several duration focal stacks. Results obtained using our method can be viewed at *www.focalsweep.com*.

Changyin Zhou
Department of Computer Science, Columbia University
E-mail: changyin.zhou@gmail.com.

Daniel Miau
Department of Computer Science, Columbia University
E-mail: dmiau@cs.columbia.edu

Shree K. Nayar
Department of Computer Science, Columbia University
E-mail: nayar@cs.columbia.edu

* Changyin Zhou is currently a software engineer at Google.

Keywords Focal Sweep · Focal Stack · Depth Recovery · Refocusing · Extended Depth of Field

1 Introduction

Traditionally, photography has required photographers to choose a wide range of parameters before taking a photo. These parameters are related to composition, depth of field, dynamic range, etc. The end result is a fixed photograph over which viewers have little control. This paradigm is changing. With the advent of computational photography and interactive displays, we are entering a new era in photography where it is possible for viewers to explore the scene after it has been captured. For example, recent commercial light field cameras, such as the Lytro™, enable a viewer to interactively control the depth of field (or focus setting) of a photograph after capture. This type of viewer control is referred to as refocusing.

The basic representation used to facilitate refocusing is a focal stack – a sequence of images of the scene that are captured at different focus settings. One way to create a focal stack is to compute it from a light field (Gortler et al., 1996; Levoy and Hanrahan, 1996). For example, a plenoptic camera (Lippmann, 1908; Ng et al., 2005) uses a lens array in the imaging pipeline to capture a 4-D light field within a single 2-D image. The captured 4-D light field can then be used to render a focal stack for image refocusing. But this results in a significant sacrifice of image resolution. This is due to the dimensionality gap between the captured and required information – while the light field is 4-D, the focal stack is only 3-D. In short, for refocusing, a light field camera captures more information than is needed and hence compromises more spatial resolution than necessary.

In this paper, we propose using an imaging system that directly captures a focal stack for refocusing by physically sweeping its focal plane across a scene. Such a system is

called a focal sweep camera. The captured stack of images is referred to as a duration focal stack – images of a possibly dynamic scene captured while the plane of focus is swept through it. We have explored the information embedded within a duration focal stack and developed an algorithm for computing a depth map which, in the case of a dynamic scene, includes depth values that correspond to different instants of time. This depth map is used as an index map to facilitate post-focusing over space and time – we refer to this as *space-time refocusing*.

The proposed space-time refocusing technique differs from the light field based refocusing techniques in two aspects. First, while the focal stack produced by a light field camera (such as a plenoptic camera) corresponds to a single instant of time, the focal stack captured by a focal sweep camera corresponds to a finite duration of time and hence includes scene motion. While capturing an instant of time can be beneficial in some situations, capturing a duration of time results in a unique and appealing user experience – viewers perceive scene dynamics along with scene depth while refocusing. Second, a focal sweep camera captures focal stacks directly. This preserves sensor spatial resolution, saves a significant amount of computation power and time for focal stack rendering, and avoids all possible focal stack rendering artifacts, which are common in most rendering techniques.

The following are the main contributions of our paper.

An efficient depth range sampling strategy. Since scene motion and camera shake can lead to motion blur in the captured images, it is important to capture the entire focal stack in the shortest possible time. We present a strategy of focal sweep imaging that samples a desired depth range in a complete and efficient manner. Completeness means the aggregate DOF of the focal stack covers the entire desired depth range, and efficiency means using the smallest number of images to cover the complete depth range. Given a desired depth range and the intrinsic parameters of a camera, the strategy can be used to compute the speed of sensor translation and the number of images needed to cover the desired depth range.

An algorithm to generate robust index-maps for refocusing. We present an algorithm to compute an in-focus index map from a duration focal stack. One major challenge of the algorithm design is to reliably measure the amount of focus even on regions of no texture (or weak texture). The fundamental difficulty is that a region of no texture appears similar across various focus levels, and thus requires a large support for focus measurement. The increase of support size not only results in an index map of lower resolution, but also causes severe scale-space effect (Perona and Malik, 1990). Scale-space effect leads to ambiguious focus measure at different support sizes, especially in textured regions and depth boundaries. Note that while existing applications such as extended depth of field are fairly tolerant to inaccurate depth in textureless regions (Agarwala et al., 2004; Kutulakos and Hasinoff, 2009), it is critical for refocusing to have a reasonably good index map even in regions of weak or no texture to enable a seamless refocusing experience.

We have built two prototypes of the focal sweep camera and used them to capture a variety of scenes. We have also implemented the imaging processing pipeline and a click-to-refocus duration focal stack viewer. Several interactive photos captured using a focal sweep camera are available at *www.focalsweep.com*.

2 Related Work

2.1 Focal Sweep and Focal Stack

A conventional lens camera has a limited depth of field. A variety of techniques have been proposed to extend depth of field in the past few decades (Castro and Ojeda-Castañeda, 2004; Cossairt et al., 2010; Dowski and Cathey, 1995; George and Chi, 2003; Guichard et al., 2009; Indebetouw and Bai, 1984; Mouroulis, 2008; Nagahara et al., 2008; Poon and Motamedi, 1987). One way to capture an extended depth of field (EDOF) image is focal sweep. A focal sweep EDOF camera captures a single image while its focus is quickly swept over a large range of depth. Hausler (1972) extended DOF of a microscope by translating the specimen along the optical axis during image exposure. Nagahara et al. (2008) extended DOF for consumer photography by sweeping the image sensor. Nagahara et al. (2008) also showed that the point-spread-function (PSF) of a focal sweep camera is nearly depth-invariant, allowing one to deconvolve a captured image with a single PSF to recover a sharp image without knowing the 3D structure of the scene. We develop a similar imaging system as in Nagahara et al. (2008), but use it to capture image stacks for image refocusing, instead of a single image for extending DOF.

Several techniques have been proposed to capture a stack of images for EDOF and 3D reconstruction. In deconvolution microscopy, for example, a stack of images of the specimen are captured at different focus settings to form a 3D image (McNally et al., 1999; Sibarita, 2005). The 3D PSF in 3D focal images is shown to be a depth-invariant double-cone. By deconvolving with the 3D PSF, a sharp 3D microscopic image can be recovered.

To produce an all-in-focus image from a focal stack, Kuthirummal et al. (2011) average all images in a focal stack and then recover an all-in-focus images by deconvolution. The average image is the same as that captured by an EDOF focal sweep camera. Guichard et al. (2009) and Cossairt and Nayar (2010) make use of chromatic aberration by capturing images corresponding to different foci in the three color channels within a single shot, and then combining the sharpness from all color channels to produce an all-in-focus im-

age. Agarwala et al. (2004) propose using a global maximum contrast image objective to merge a focal stack into a single all-in-focus image.

Hasinoff et al. (2009) compare various capture strategies for reducing defocus blur in a comprehensive framework where both sensor noise and deblurring error are taken into account. Their analysis and subsequent analysis in (Kutulakos and Hasinoff, 2009) show that for extending DOF, focal stack photography has two performance advantages over one-shot photography: 1) it allows one to capture a given DOF faster; and 2) it achieves higher signal-to-noise ratio (SNR) for a given exposure time.

Hasinoff and Kutulakos (2009) consider the problem of minimizing the time to capture a scene with a given DOF and a given exposure level. This is related to the optimization that we use to minimize the time required to capture a focal stack. While Hasinoff and Kutulakos (2009) are interested in determining the F-numbers and the number of images for spanning a given DOF, we are interested in finding the optimal translation speed of the sensor.

Kutulakos and Hasinoff (2009) use a similar algorithm as in Agarwala et al. (2004) to compute EDOF images from focal stacks by assuming that scenes are static. Their algorithms are optimized to produce artifact-free images with minimal blur. Our goal is diffeent – our algorithm is designed to faciliate a seamless refocusing experience.

2.2 Light Field Camera and Image Refocusing

The concept of light field has a long history. In the early 20^{th} century, Lippmann (1908) and Ives (1930) proposed plenoptic camera designs to capture light fields. The idea of light field resurfaced in the fields of computer vision and graphic in the late 1990s when Levoy and Hanrahan (1996) and Gortler et al. (1996) described the 4D parameterization of light fields and showed how new views can be rendered by using light field data.

A number of light field cameras have been implemented in recent years. Levoy et al. (2006) used a plenoptic camera to capture the light field of specimens and proposed algorithms to compute a focal stack from a single light field image, which can be processed as in deconvolution microscopy to produce a sharp 3D volume. Ng et al. (2005) and Ng (2006) used the same plenoptic camera design to demonstrate its application in digital refocusing. Georgeiv et al. (2006) and Georgiev and Intwala (2006) showed a number of variants of light field camera designs with different tradeoffs between spatial and angular resolution. Light field cameras have also been built using camera arrays (Wilburn et al., 2005).

Light fields can be used for various applications, including 3D display (Javidi and Okano, 2002; Matusik and Pfister, 2004), synthetic aperture photography (Isaksen et al., 2000), glare reduction (Raskar et al., 2008), etc. Light fields can also be used to render a focal stack for image refocusing. However, light field cameras like plenoptic cameras capture 4D light fields, while image refocusing only requires 3D focal stacks. For this dimensionality gap, light field cameras capture more information than is needed and significantly sacrifice image spatial resolution.

2.3 Depth from Focus or Defocus

Two other techniques closely related to our proposed technique are depth from focus and depth from defocus. Instead of capturing a complete focal stack, depth from focus or defocus captures only a few images with varying focus settings or aperture patterns and uses focus as a cue to compute depth maps of scenes (Chaudhuri and Rajagopalan, 1999; Nayar et al., 1996; Pentland, 1987; Rajagopalan and Chaudhuri, 1997; Zhou et al., 2009).

One key component of depth from focus is to compute focus measure reliably. Focus measure is a measurement of the sharpness of an image patch. Numerous methods such as the Laplacian of the Gaussian, the difference of the Gaussian have been proposed to compute focus measure. There are several difficulties associated with computing focus measure reliablely. The scale-space effect (Perona and Malik, 1990) presents a major difficulty since it can lead to ambiguities of focus measure at different scales. In addition, image patches at depth discontinuities may cross multiple depth layers and make focus measure inaccurate. Furthermore, since focus measure does not reveal anything about depth in non-textured regions, techniques such as plane fitting (Tao et al., 2001), graph-cut (Boykov and Kolmogorov, 2004), and belief prorogation (Yedidia et al., 2001)) have been employed to fill the resulting holes in depth maps.

3 3D Focal Volume and the Sampling Strategy for Refocusing

A 3D (XYT) focal volume, which is the concatenation of all the images in a focal stack along the temporal dimension, encodes more visual information about a scene than does a single image. Figure 1 (a) shows a 3D focal volume of a synthetic scene of colored balls in motion and (b) shows one 2D *XT* slice of the 3D focal volume. In focal sweep, axial object motion (along the optical axis) is usually negligible compared with the motion of the focal plane. As a result, in the 2D *XT* slice each ball appears as a double-cone, and balls with lateral motion (perpendicular to the optical axis) result in titled double-cones. By finding the apex of every double-cone in the 3D focal volume (shown as a yellow band in Figure 1 (b)), we can obtain an all-in-focus image. For a static scene, an all-in-focus image is an extended depth of

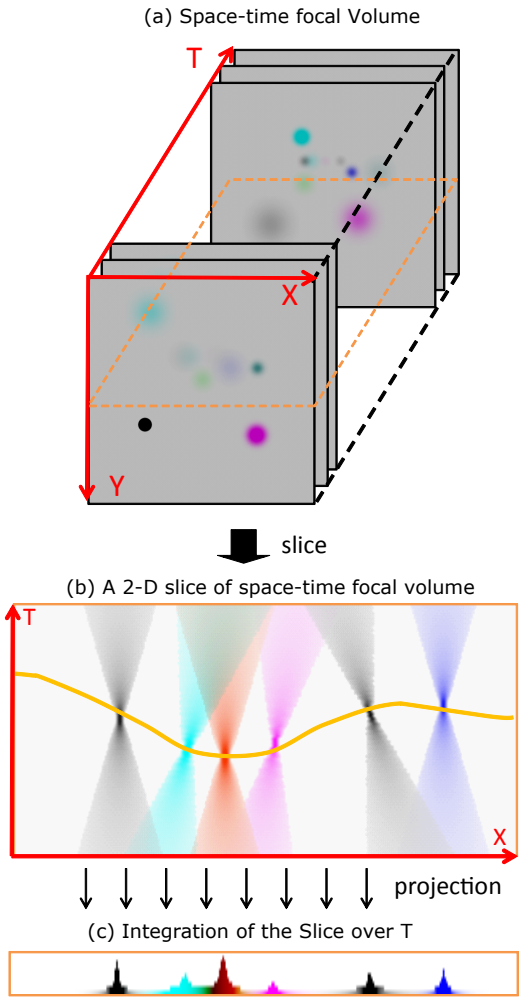


Fig. 1 (a) A space-time focus volume of a synthetic scene with color balls with motion. The balls move as the focus changes with time (in the T dimension). (b) A 2D XT slice of the 3D volume, in which each small ball appears as a double-cone. The double-cones corresponding to moving balls are tilted. (c) Integrating the volume along the T dimension produces an image like the one captured by a focal-sweep EDOF camera (prior to processing). Each object appears sharp in the EDOF image regardless of its depth.

field image. For a dynamic scene, an all-in-focus image may consist of scene points corresponding to different instants of time, thus we refer to it as a *space-time in-focus image*.

Capturing a stack of well-lit images is difficult because of possible camera or scene motion. Large scene motions make it difficult to process and analyze the captured 3D focal volumes. Therefore, it is better to minimize the total capture time so as to minimize the amount of motion during capture. To this end, we devise an efficient focal sweep strategy to sample a desired depth range.

3.1 Sampling depth range

Within a given time budget, the number of images that can be captured is limited by sensor frame-rate and signal-to-noise (SNR) considerations. In addition, to ensure that all scene points are focused in at least one of the images in a focal stack, one must capture a focal stack such that the combined depth of field of a focal stack covers the entire desired depth range. In essence, the constraints above forms a sampling problem of a desired depth range. We argue that an ideal sampling strategy should satisfy the following two conditions:

- Completeness: the aggregated DOFs of all captured images should cover the entire desired depth range. If the desired depth range is DOF^*, we have

$$DOF_1 \bigcup DOF_2 \bigcup DOF_3 ... \bigcup DOF_n \supseteq DOF^*, \quad (1)$$

  where $\bigcup$ denotes a union operation and $\supseteq$ denotes superset.
- Efficiency: No two DOFs should overlap, so that the number of required images is minimized.

$$DOF_1 \bigcap DOF_2 \bigcap DOF_3 ... \bigcap DOF_n = \varnothing, \quad (2)$$

  where $\bigcap$ denotes intersection.

Hasinoff and Kutulakos (2009) refers to an image sequence as *Sequence with Sequential DOFs*, if the end-point of one image's DOF is the start-point of the next image's DOF. The ideal capture sequence described above is a sequence with sequential DOFs that covers the entire desired depth range.

We start our DOF analysis from the Thin Lens Law, $1/f = 1/u + 1/z$, where f is the focal length of the lens, u is the sensor-lens distance, and z is the object distance. As in common practice, we transform the equation to the reciprocal domain $\hat{f} = \hat{u} + \hat{z}$, where $\hat{x} = 1/x$. In its reciprocal form, the Thin Lens Law becomes a linear equation. The reciprocal of object distance, $\hat{z}$, is often expressed in the unit of diopter $(1/m)$. Depth of field, $[z_1, z_2]$, is the depth range in which the blur radius is less than the circle of confusion, c. In this paper, we use the pixel size as the diameter of circle of confusion (common practice in imaging). For a given sensor-lens distance $\hat{u}$, the DOF in the reciprocal domain, $\hat{z}_1$ and $\hat{z}_2$, can be expressed as:

$$\hat{z}_1 = \hat{z} + \hat{u} \cdot c/A \quad (3)$$

$$\hat{z}_2 = \hat{z} - \hat{u} \cdot c/A \quad (4)$$

where A is the diameter of the lens aperture. Both the position and the range of DOF change with the sensor position. Figure 2 (a) shows the geometry of DOF for sensor positions u on the left and illustrates the DOF in the reciprocal domain on the right. The yellow line in the figure represents

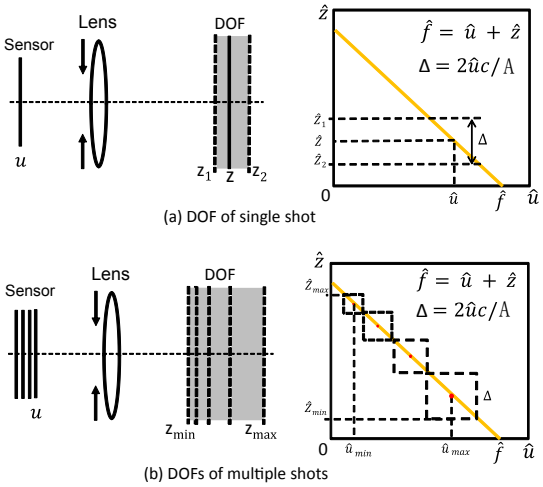


Fig. 2 Efficient and complete focus sampling. (a) Left: A geometrical illustration of depth of field. Objects in the range $[Z_1, Z_2]$ will appear focused when u and z satisfy the Thin Lens Law. Right: The Thin Lens Law is shown as an yellow line in the reciprocal domain. Z_1 and Z_2 can be easily located in the reciprocal domain (or in diopter) by Eqn 3 and Eqn 4. (b) In order to have an efficient and complete focus sampling, the DOFs of consecutive sensor positions (e.g., $\hat{v}_{i-1}, \hat{v}_i, \hat{v}_{i+1}$) must have no gap or overlap.

the Thin Lens Law. According to Equation 3 and Equation 4, for an arbitrary sensor position $\hat{u}$, the size of DOF in the reciprocal domain is $\Delta = 2 \cdot \hat{u} \cdot c/A$.

For an efficient and complete sampling, we require that each pair of consecutive DOFs have no overlap and no gap as shown in Figure 2 (b). According to the Thin Lens Law, we have

$$|\hat{u}_i - \hat{u}_{i+1}| = |(\hat{f} - \hat{z}_i) - (\hat{f} - \hat{z}_{i+1})| \tag{5}$$
$$|\hat{u}_i - \hat{u}_{i+1}| = |\hat{z}_i - \hat{z}_{i+1}|, \tag{6}$$

And then for Eqn 3 and Eqn 4, we derive:

$$|\hat{u}_i - \hat{u}_{i+1}| = (\hat{u}_i + \hat{u}_{i+1}) \cdot c/A, \tag{7}$$

where $\hat{u}_i$ and $\hat{u}_{i+1}$ are the sensor positions of two consecutive DOFs.

In consumer photography, we have $z \gg u$ and so $\hat{u}_i \approx \hat{f}$. By approximating Equation 7 we have:

$$\hat{u}_i \cdot \hat{u}_{i+1} \cdot |\hat{u}_i - \hat{u}_{i+1}| = \hat{u}_i \cdot \hat{u}_{i+1} \cdot (\hat{u}_i + \hat{u}_{i+1}) \cdot c/A \tag{8}$$
$$|u_{i+1} - u_i| = (u_i + u_{i+1}) \cdot c/A \tag{9}$$
$$\delta u \approx 2 \cdot f \cdot c/A \tag{10}$$
$$\delta u \approx 2 \cdot c \cdot N, \tag{11}$$

where $N = f/A$ is the f-number of the lens. Equation 11 suggests that an efficient and complete sampling strategy should move the sensor by a constant distance δu between every consecutive image captures. The moving distance is determined by the pixel size c and f-number N. Notice that this is a constant step in the normal domain, no longer in the reciprocal domain. Hence, if a camera operates at a constant frame-rate P, the ideal strategy to sample the desired depth range is to sweep the sensor at a constant speed:

$$s = \frac{\delta u}{\delta t} = 2 \cdot c \cdot N \cdot P. \tag{12}$$

3.2 Refocusing and In-focus Index Map

Let $F_1, F_2, ..., F_t$ be the t images of resolution $M \times N$ in a focal stack. Let $i = refocus(p)$ be a function which takes a pixel location $p = (x, y)$ as input, and returns an index $i \in 1, 2, \ldots, t$, such that p is best focused in F_i. An index map $IMap(p)$ is a 2D $M \times N$ matrix which stores the pre-computed result of $refocus(p)$ for each p. When an interactive display receives a refocusing request at p, the display shows $F_{\{IMap(p)\}}$.

If the scene is static, each pixel location p corresponds to a single scene point. In this case, there is a unique image index i such that $F_i(p)$ is the best focused, and the function $refocus(p)$ is well-defined. However, if the scene is changing, a p may correspond to multiple scene points in the duration focal stack. In this case, there may be multiple image indices for which p is in focus.

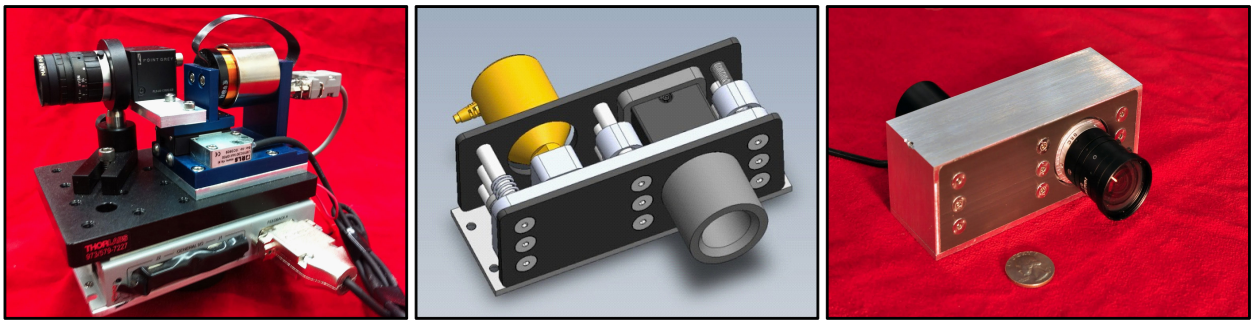
One way to resolve the ambiguity mentioned above is by explicitly considering the motion of points in a scene. In order to do this, when the user clicks on p on image number j, the scene point S present at 3D location (x, y, j) is tracked across time and the image that contains the sharpest image of S is displayed. In this case, refocus is defined as $i = refocus(x, y, j)$ and thus the resulting index map is 3D. Computing this index map accurately relies on accurate motion estimation.

In this paper, for simplicity, we choose the 2D definition of the index map, without explicitly considering the motion. If a pixel location p is in focus at two or more image indices, we design a $refocus(p)$ such that the index which yields the seamless refocusing experience is selected. Note that different designs of refocus(p) may yield different refocusing experience. Ideally, the choice should be made based on user intention or preference when they click on a pixel in the viewer. We leave the study of the other definitions and their impacts on user experience to future work.

4 Focal Sweep Camera

4.1 Prototypes

Focal sweep can be implemented in multiple ways. One way is to directly sweep the image sensor or camera lens along the optical axis. A variety of actuators such as voice coil motors, piezoelectric motors, ultrasonic transducers, and DC motors can be used to translate the sensor at the optimized



(a) A side view of Prototype 1 (b) CAD rendering of Prototype 2 (c) A side view of Prototype 2

Fig. 3 Two focal sweep camera prototypes. (a) In Prototype 1, a sensor sweep is driven by a voice coil actuator; (b) In Prototype 2, a lens sweep is driven by a linear actuator.

speed. Many commercial lenses have built-in auto-focus mechanism, which may be programmed to perform focal sweep. A liquid lens (Ren and Wu, 2007; Ren et al., 2006) which focuses at different distances when different voltages are applied is another way to implement focal sweep.

We built two prototype focal sweep cameras as shown in Figure 3. Prototype 1, as shown in Figure 3 (a), uses a Fujinon HF9HA-1B, 9mm, F/1.4, C-mount lens, and a Pointgrey Flea 3 camera with a max resolution of 1328×1048 and a frame rate of 120fps. The sensor is driven by a voice coil actuator (BEI LA15-16-024). This setting is similar to the one used in (Nagahara et al., 2008) for capturing extended depth of field. The sensor is tethered to a laptop via a USB 3.0 cable and synced with the motor start/stop signal. The voice coil motor and the motor controller are able to translate the sensor at the speed of $1.47mm/s$. In almost all scenes that we have experimented with, the sensor motion is less than $0.3mm$ to cover the entire desired depth range. With our 120fps camera, this can be completed in less than 0.21 second. The major advantage of this implementation is that all of the parts are off-the-shelf components. This first prototype demonstrates that a focal sweep camera can be built with minimal effort.

Prototype 2, as shown in Figure 3 (b), is a more compact design, in which a sensor secured on a structure and the lens can be translated during the sensor's integration time. In this prototype we use a compact linear actuator instead of a voice coil motor, allowing us to reduce the camera's overall size. The same lens and camera as in Prototype 1 are used. During the integration time, the sensor is translated from the near focus position to the far focus position. With this prototype, we are able to translate the sensor at a top speed of $0.9mm/s$. The major advantage of this implementation is its compactness and its close resemblance to existing consumer point-and-shoot camera designs.

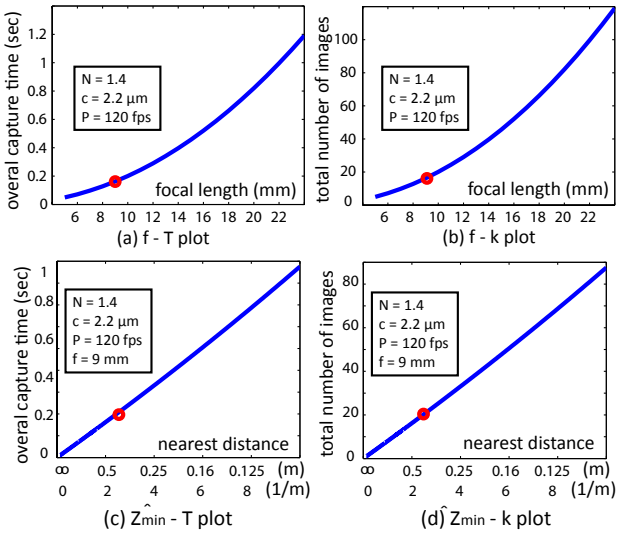


Fig. 4 For a given pixel size, frame rate, and f-number, the overall capture time and total image count are highly related to focal length and scene distance range. (a) shows the $f-T$ plot of the overall capture time T with respect to focal length f to cover a wide depth range from $0.4m$ to infinity. (b) shows the $f-k$ plot of the total image number k with respect to focal length f to cover a wide depth range from $0.4m$ to infinity. (c) and (d) show the plots of overall time T and total image number k with respect to the depth range (in both diopter and meter), respectively (f = 9mm). In each plot, the red spot indicates the most typical setting in our implementation.

4.2 Speed and Range of Focal Sweep

As in conventional photography, users first determine the frame rate P and f-number N according to the speed of object motion, the lighting condition, and the desired amount of defocus in the captured images for each scene. Then, the ideal speed of sensor sweep s can be computed using Equation 12. Note that s is independent of camera focus and depth range of scenes.

Although the sweep speed is independent of the focal length of a lens and the depth range of a scene, the overall capture time of a focal stack, T (or the number of images, k) depend on the focal length and depth range. Consider a

depth range from $0.4m$ to infinity, Figure 4 (a) shows how the overall capture time T varies with focal length f in a camera with $N = 1.4$, $c = 2.2\mu m$, and $P = 120$ fps.

From the Thin Lens Law, the distance D required to sweep the focal plane over the depth range $[z_1, z_2]$ can be expressed as:

$$D = f^2 \frac{z_2 - z_1}{(z_2 - f)(z_1 - f)} \tag{13}$$

When $z_1 \gg f$ and $z_2 \gg f$, D is approximately proportional to f^2. As a result, when the sensor/lens is translated at the constant speed as indicated by Equation 12, T increases proportionally to f^2. Figure 4 (b) plots the total number of captured images, k, with respect to focal length f. Again, k is proportional to f^2.

In many scenarios, the desired depth ranges from a certain distance, Z_{min}, to the infinity. Figure 4 (c) and (d) plot the capture time T and the total image count k with respect to $\hat{Z}_{min} = 1/Z_{min}$. We can see that both are linear. The closer the foreground is to the camera the longer the capture time becomes. The x axis is labeled in the unit of both diopter $(1/m)$ and distance (m) for easy reference.

It can be noticed from Figure 4 that the required capture time and image counts have a huge range at different settings. The red dot in each plot indicates a typical setting in our implementation. Most of our scenes have depth that ranges from $0.4m$ to infinity. So we need to capture about 20 images to sample the entire depth range, which requires about $0.2sec$ (with a $9m$ lens and a $120p$ sensor). Our prototypes are also able to capture scenes with smaller Z_{min}, but it will take longer time to complete the sweep as shown in Figure 4 (c). Figure 5 illustrates an example of space-time focal stack that was captured using our prototype camera.

5 Algorithm

Figure 6 shows an overview of the proposed image processing algoritm. After a stack of images, $\{F_i^0\}$, are captured, we first apply a typical multi-scale optical flow algorithm to estimate frame-to-frame global transformations to account for hand-shake and correct magnification changes. The stabilized focal stack $\{F_i\}$ is then used to compute a space-time in-focus image (Section 5.1) and space-time in-focus index maps at various scales (Section 5.2). In Section 5.3, we describe a new approach to merge multi-scale space-time in-focus index maps into one high-quality space-time in-focus index map. There are two key ideas in the algorithm:

– At any given scale, we propose a novel approach to compute an index map. In literature, it is common to first estimate an index map (or depth map) using focus measure, and then use the index map to produce an all-in-focus image (Agarwala et al., 2004; Hasinoff and Kutulakos, 2009). In this paper, however, we take a different strategy. We first compute an all-in-focus image without knowing the index map, and then use the all-in-focus image to help estimate the index map. We will show the advantage of using this new strategy.
– We use a pyramid strategy to handle regions with no or weak texture. For each pixel, we estimate its index (the frame where it is best focused) at multiple scales. Due to the scale-space effect (Perona and Malik, 1990), the index may not be consistent at different scales, especially in regions with weak or no texture, depth discontinuities, or object motions. This inconsistency is one of the fundamental difficulties in the algorithm design, and we show a simple yet effective solution.

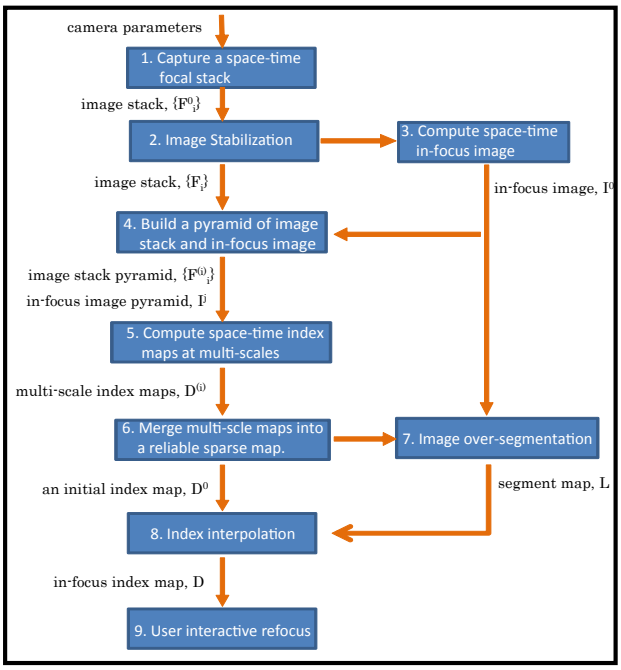
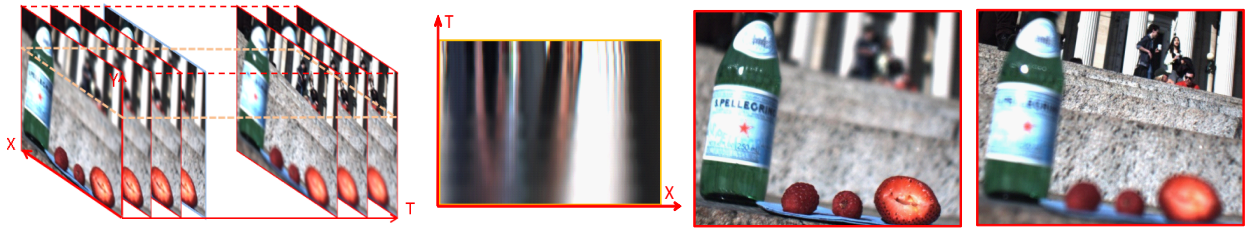


Fig. 6 A diagram illustrating the process from capturing a space-time focal stack, to generating an in-focus index map, and to interactive image refocusing.

5.1 Space-time In-Focus Image

Given a focal stack, we first compute a space-time in-focus image without the knowledge of an index map. The idea is inspired by the focal sweep EDOF technique (Kuthirummal et al., 2011). Kuthirummal et al. (2011) show that the mean of a focal stack preserves image details, and they deconvolve the averaged image with a $(1/x)$-shape integrated point-spread-function (IPSF) to recover an all-in-focus EDOF image without knowing the depth map. This approach is further shown to be robust in regions of depth edges, occlusions, and even object motion. In Figure 7, we show the mean image of a space-time focal stack (a) and the EDOF image after deconvolution (b).



(a) A Space-time focal stack (b) 2D slice of the stack (c) The first frame (d) The last frame

Fig. 5 A sample space-time focal stack captured using our focal sweep camera prototype 1. (a) A space-time focal stack of 25 images; (b) A 2D slice of the 3D stack; (c) The first frame of the stack (focused on the foreground); (d) The last frame of the stack (focused on the background). The frame rate of capturing was $120fps$ and it took the focal sweep camera about $0.2sec$ to complete the focal sweep.

(a) Mean image (b) Mean image after deblurring (c) Weighted average image

Fig. 7 Space-time in-focus images computed using different approaches and their close-ups. (a) The mean of all images in the stack; (b) The mean image deconvolved using an integrated PSF; (c) Weighted average of all images in the stack; (d) The best focused patches in the captured focal stack.

Although both (a) and (b) preserve most high frequency information, the average image yields a low contrast (especially when the number of images increases), and the deconvolved EDOF image (b) is prone to image artifacts. In addition, deconvolution is computationally expensive, especially for mobile devices.In this paper, we compute a space-time in-focus image $I(x,y)$ as a weighted sum of all images:

$$I(x,y) = \frac{\Sigma_i W_i(x,y) \cdot F_i(x,y)}{\Sigma_i W_i(x,y) + \epsilon}, \tag{14}$$

where the weight $W_i(x)$ is defined as the variance of the Laplacian patch:

$$W_i(x,y) = \mathcal{V}(\triangle \mathbf{P}_i(x,y,d)). \tag{15}$$

$\mathbf{P}_i(x,y,d)$ here represents a patch of size d centered at (x,y) in the i^{th} frame. With this strategy, severely blurred patches will have much less weight than sharper patches do, reducing the hazy effects that one often sees in the averaged image from Figure 7(a). As shown in (c), the weighted sum is sharp and has high contrast even without deconvolution. Although the weighted sum (c) is sometimes not as sharp as the deblurred image (b), it avoids the risk of producing deconvolution artifacts and reduces the halo effects introduced by object motions. Note that our final goal is not to produce an all-in-focus image, but to use an all-in-focus image to compute the in-focus index map. For this purpose, it is important to have an all-in-focus image that is free of high-frequency artifacts.

5.2 Space-time In-focus Index Maps at Various Scales

We use the computed all-in-focus image $I(x,y)$ to help estimate in-focus index map. For each pixel location (x,y), we look for the frame where its surrounding patch is most similar in high frequencies to that in $I(x,y)$. Then, the in-focus index map $IMap(x)$ is estimated as:

$$IMap(x,y) = \arg\min_i S(F_i(x,y), I(x,y)), \tag{16}$$

where S measures the high frequency similarity between F_i and I at each pixel and is defined as

$$S(P,Q) = |\triangle(\mathbf{P} - \mathbf{Q})| \otimes \sqcap(r), \tag{17}$$

where the bold $\mathbf{P}$ and $\mathbf{Q}$ denote the patches at P and Q, respectively, $\otimes$ is convolution, and $\sqcap(r)$ is a pillbox function

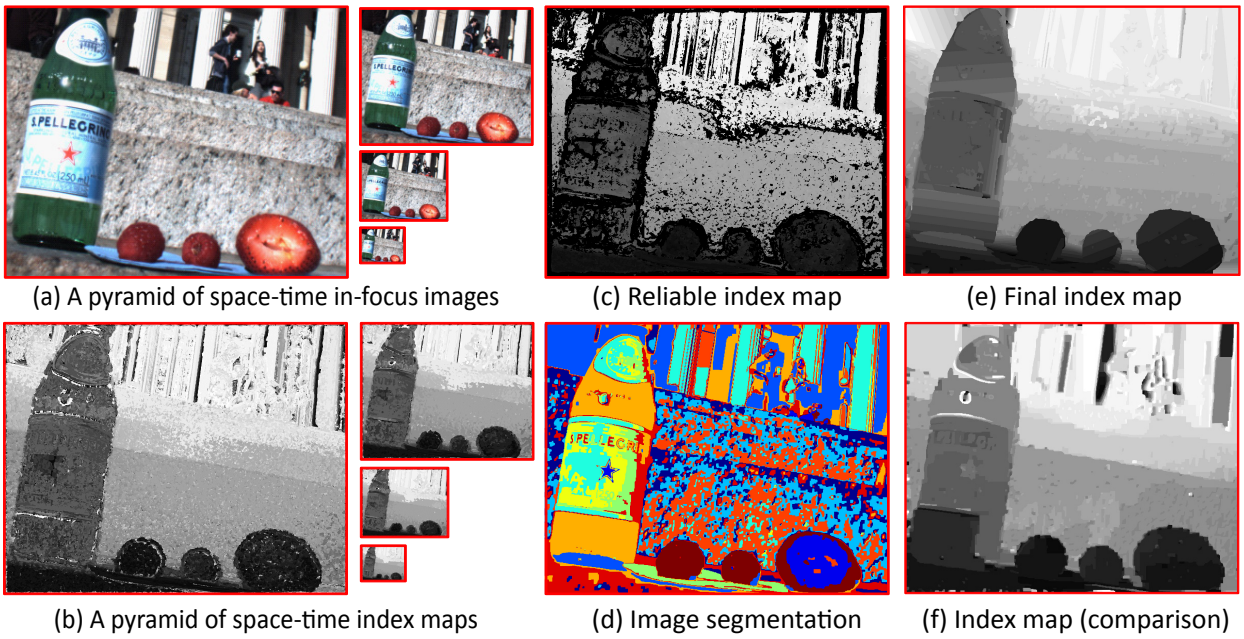


Fig. 8 (a) A pyramid of space-time in-focus images; (b) A pyramid of space-time index maps; (c) A reliable index map that is computed from (b) using index consistence; (d) An over-segmentation of the full-resolution in-focus image; (e) Our final depth map computed from (c) and (d) by hole-filling; (f) An index map computed using a traditional algorithm which uses difference-of-Gaussians as focus measure (Agarwala et al., 2004; Hasinoff and Kutulakos, 2009) and Graph-cut for global optimization(Boykov and Kolmogorov, 2004).

of radius r. The key idea here is to measure the similarity in high frequencies. The weighted mean image preserves good high frequencies as in the best focused layer even at depth discontinuities and for moving objects. By convolving with $\sqcap(r)$, we consider a neighborhood in processing each pixel.

5.3 Merging and Interpolating Index Maps

Due to the scale-space effect and depth discontinuities, the index maps computed at different scales (or different neighborhood size r) can be significantly different. Figure 8 (b) shows the index map pyramid $IMap^{(i)}, i = 1, 2, \ldots, k$, where k is the total number of the levels of the pyramid. At each level, the focal stack reduces its spatial resolution by 2×2 from its upper level. The index maps at different scales are significantly different, especially at depth boundaries. It is a challenging problem to pick the right scale for each pixel.

We propose a novel multi-scale technique to solve this problem in four steps.

1. Compute one index map $IMap^{(i)}$ at each level i, $i = 1, 2, \ldots, k$.

2. Construct a reliable but sparse index map $IMap^0$ by only accepting indices that are consistent in all levels:

$$IMap^0(x,y) = \begin{cases} \overline{IMap^{(i)}(x,y)}, & \text{if } \max[IMap^{(i)}(x,y)] \\ & - \min[IMap^{(i)}(x,y)] \\ & < \tau \\ \emptyset, & \text{otherwise} \end{cases} \tag{18}$$

   τ is set to a small number to enforce consistency. One sample is shown in Figure 8 (c). (We use $d = 7, k = 7, r = 5$ in our implementation.) The pixels with no index assignment are shown in black. The observation is that the index map is dense in regions of rich texture, and sparse in non-textured regions and depth boundaries.

3. Over-segment the in-focus image $I(x, y)$. An image segmentation algorithm like graph-cut assigns a color or number to each pixel, as shown in Figure 8 (d). Each connected region with the same color assignment is defined as a *segment*.

4. In each segment, fill the holes in $IMap^0$ by interpolation according to the following two simple rules:
   - If a segment has m or more valid (and reliable) indices, do interpolation by fitting a plane to the valid indices.

– If the number of valid indices is less than m, do nearest neighbor interpolation.

This gives us the final index map $IMap(x, y)$. Our observation is that segments in a textured region have many reliable indices in $IMap^0$, which yield a reliable plane fitting; segments in a texture-less region or depth discontinuities have few indices and so the nearest neighbor interpolation propagates index information from the region boundary.

It is important to note that a smoothed index map at depth boundary must be avoided, because it would cause the image to be refocused to a middle depth where neither foreground nor background is correctly focused. Our observation is that $IMap^0$ is very sparse along depth boundaries. By doing nearest neighbor interpolation, we avoid smoothing out the index map in these regions. Nearest neighbor interpolation may not be able to produce an accurate spatial boundary between foreground and background, but fortunately, users are much more tolerant to this spatial inaccuracy. This is because user input itself (e.g., a finger tapping on a touch screen) has much lower precision than image resolution.

Figure 8 (d) shows a result of image over-segmentation using Graph-cut (d), and Figure 8 (e) shows the index map after interpolation. We can see that the index map is sharp at depth boundaries, and smooth in non-textured regions.

6 Experiments

We captured a variety of indoor and outdoor scenes to demonstrate that the proposed camera is well-suited to photograph common daily scenes. Shown in Figure 9 are 4 duration focal stacks selected from a larger collection. In each dataset we show 2 images from the focal stack as well as the computed in-focus image and the associated index map. Note that none of these index maps have obvious holes or artifacts, even in regions of weak or no texture. The index maps are also sharper at depth boundaries.

We qualitatively evaluated the index maps generated by our proposed algorithm with the ones generated using the traditional focus measure maximization algorithm. Various definitions of focus measure exist in literature(Nayar and Nakagawa, 1990; Nayar et al., 1996; Subbarao and Tyan, 1998; Xiong and Shafer, 1993). We adopted the one used in the photomontage paper (Agarwala et al., 2004), in which the focus measure is defined as a simple local contrast according to the Difference-of-Gaussians filter. For the index maps generated by Difference-of-Gaussians filter, we further polished the results using Graph-cut (Boykov and Kolmogorov, 2004). Graph-cut as a global optimization technique helps fill out the holes and smooths the index map, as shown in Figure 8 (f). 8 (e) shows that the proposed algorithm produces better results in non-textured or specular regions and depth discontinuities, which are important for image refocusing.

A refocusing viewer has been implemented to allow interaction with the captured duration focal stacks. In the refocusing viewer, for any pixel (x, y) that a user clicks, the displayed image transitions from the present image to the image indexed by $IMap(x, y)$. The transition is made smooth by sequentially displaying the images between the present index to $IMap(x, y)$. We have made our refocusing viewer available online at *www.focalsweep.com*.

7 Summary and Discussion

In this paper, we have presented a focal sweep imaging system to capture duration focal stacks for refocusing. The proposed camera sweeps its focal plane at a sufficiently high speed so that the aggregated DOF of the captured focal stack efficiently and completely covers the desired depth range. The major benefit of focal sweep imaging systems lies in the fact that the camera directly captures all the images required for refocusing. While light field cameras, which are commonly used for refocusing, require significant sacrifice of image resolution, our system produces high-quality, artifact-free, full-resolution images at every focus with minimal computation cost.

Due to object motion, each pixel that a user clicks on might correspond to different objects at different focus layers (or time points). For example, a defocused object in motion often appears blended with its background object; there could be cases when it is preferred to estimate object motion and perform refocusing along the estimated motion trajectory. Solving these ambiguities often requires a deeper understanding of user intentions. In this paper, however, we choose a simple design by not explicitly considering object motion in the algorithm design. There are other possible refocusing choices as discussed in Section 3.2, which deal with the ambiguities in different manners. We decide to leave them as future work.

The focal sweep functionality could potentially be added to many of the existing cameras with minimal modifications. The modification could be as simple as a firmware update. Auto-focus has become a standard feature in cameras across all categories, from cell phone cameras to professional SLRs. An image is typically captured *after* the camera adjusts its focus–either by moving the sensor (cell phone) or changing the relative distance between the optical components within a lens (SLR). In comparison, a focal stack is captured *while* the focal plane of the imaging system is swept through a predefined range. By proper control of synchronization between image captures and focus adjustment, many of the existing cameras can be used to capture focal sweep photographs.

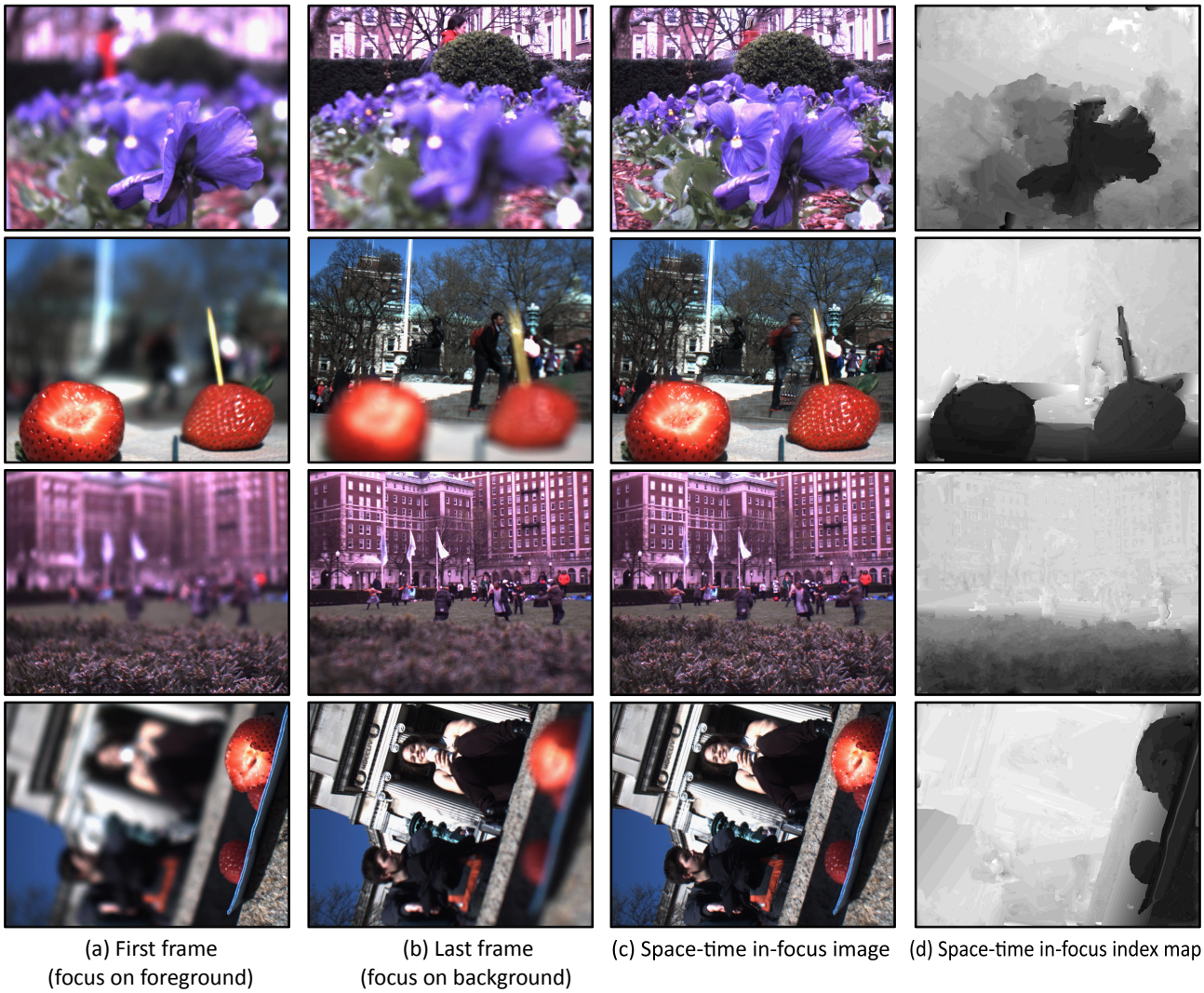


Fig. 9 More experimental results. Each row corresponds to a scene. From left to right, (a) and (b) are the first and last frames captured with focal sweep, (c) are the computed space-time in-focus images, and (d) are the estimated space-time in-focus index maps. The resulting index maps are used for image refocusing, as demonstrated on our website *www.focalsweep.com*.

Acknowledgments

This research was funded in part by ONR Grant No. N00014-11-1-0285, ONR Grant No. N00014-08-1-0929, and DARPA Grant No. W911NF-10-1-0214.

References

Agarwala A, Dontcheva M, Agrawala M, Drucker S, Colburn A, Curless B, Salesin D, Cohen M (2004) Interactive digital photomontage. In: ACM Transactions on Graphics (TOG), ACM, vol 23, pp 294–302

Boykov Y, Kolmogorov V (2004) An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(9):1124–1137

Castro A, Ojeda-Castañeda J (2004) Asymmetric phase masks for extended depth of field. Applied Optics 43(17):3474–3479

Chaudhuri S, Rajagopalan A (1999) Depth from defocus: a real aperture imaging approach. Springer Verlag

Cossairt O, Nayar S (2010) Spectral Focal Sweep: Extended depth of field from chromatic aberrations. In: IEEE Conference on Computational Photography, pp 1–8

Cossairt O, Zhou C, Nayar S (2010) Diffusion coded photography for extended depth of field. In: SIGGRAPH, ACM, pp 1–10

Dowski E, Cathey W (1995) Extended depth of field through wave-front coding. Applied Optics 34(11):1859–1866

George N, Chi W (2003) Extended depth of field using a logarithmic asphere. Journal of Optics A: Pure and Applied Optics 5:S157

Georgeiv T, Zheng K, Curless B, Salesin D, Nayar S, Intwala C (2006) Spatio-Angular Resolution Tradeoff in Integral Photography. In: In Eurographics Symposium on Rendering

Georgiev T, Intwala C (2006) Light field camera design for integral view photography. Tech. rep., Adobe

Gortler S, Grzeszczuk R, Szeliski R, Cohen M (1996) The lumigraph. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, ACM, pp 43–54

Guichard F, Nguyen H, Tessières R, Pyanet M, Tarchouna I, Cao F (2009) Extended depth-of-field using sharpness transport across color channels. Technical Paper, DXO Labs

Hasinoff S, Kutulakos K (2009) Light-efficient photography. IEEE Transactions on Pattern Analysis and Machine Intelligence 1(1):1

Hasinoff S, Kutulakos K, Durand F, Freeman W (2009) Time-constrained photography. In: IEEE International Conference on Computer Vision, pp 333–340

Hausler G (1972) A method to increase the depth of focus by two step image processing. Optics Communications 6(1):38–42

Indebetouw G, Bai H (1984) Imaging with Fresnel zone pupil masks: extended depth of field. Applied Optics 23(23):4299–4302

Isaksen A, McMillan L, Gortler S (2000) Dynamically reparameterized light fields. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co., pp 297–306

Ives H (1930) Parallax panoramagrams made with a large diameter lens. Journal of the Optical Society of America A 20(6):332–340

Javidi B, Okano F (2002) Three-dimensional television, video, and display technologies. Springer

Kuthirummal S, Nagahara H, Zhou C, Nayar S (2011) Flexible depth of field photography. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(1):58–71

Kutulakos K, Hasinoff S (2009) Focal stack photography: High-performance photography with a conventional camera. Proc 11th IAPR Conference on Machine Vision Applications pp 332–337

Levoy M, Hanrahan P (1996) Light field rendering. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, ACM, pp 31–42

Levoy M, Ng R, Adams A, Footer M, Horowitz M (2006) Light field microscopy. In: ACM Transactions on Graphics (TOG), ACM, vol 25, pp 924–934

Lippmann G (1908) La photographie integrale. Comptes-Rendus, Academie des Sciences (146):446–551

Matusik W, Pfister H (2004) 3d tv: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. In: ACM Transactions on Graphics (TOG), ACM, vol 23, pp 814–824

McNally J, Karpova T, Cooper J, Conchello J (1999) Three-dimensional imaging by deconvolution microscopy. Methods 19(3):373–385

Mouroulis P (2008) Depth of field extension with spherical optics. Optics Express 16(17):12,995–13,004

Nagahara H, Kuthirummal S, Zhou C, Nayar S (2008) Flexible depth of field photography. In: European Conference on Computer Vision

Nayar S, Nakagawa Y (1990) Shape from focus: An effective approach for rough surfaces. In: IEEE International Conference on Robotics and Automation, pp 218–225

Nayar S, Watanabe M, Noguchi M (1996) Real-time focus range sensor. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(12):1186–1198

Ng R (2006) Digital light field photography

Ng R, Levoy M, Brédif M, Duval G, Horowitz M, Hanrahan P (2005) Light field photography with a hand-held plenoptic camera. Stanford Computer Science Technical Report 2

Pentland A (1987) A new sense for depth of field. IEEE Transactions on Pattern Analysis and Machine Intelligence (4):523–531

Perona P, Malik J (1990) Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(7):629–639

Poon T, Motamedi M (1987) Optical/digital incoherent image processing for extended depth of field. Applied Optics 26(21):4612–4615

Rajagopalan A, Chaudhuri S (1997) Optimal selection of camera parameters for recovery of depth from defocused images. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 219–224

Raskar R, Agrawal A, Wilson C, Veeraraghavan A (2008) Glare aware photography: 4d ray sampling for reducing glare effects of camera lenses. In: ACM Transactions on Graphics (TOG), ACM, vol 27, p 56

Ren H, Wu S (2007) Variable-focus liquid lens. Opt Express 15(10):5931–5936

Ren H, Fox D, Anderson P, Wu B, Wu S (2006) Tunable-focus liquid lens controlled using a servo motor. Opt Express 14(18):8031–8036

Sibarita J (2005) Deconvolution microscopy. Microscopy Techniques pp 1288–1291

Subbarao M, Tyan J (1998) Selecting the optimal focus measure for autofocusing and depth-from-focus. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8):864–870

Tao H, Sawhney H, Kumar R (2001) A global matching framework for stereo computation. In: IEEE International Conference on Computer Vision, vol 1, pp 532–539

Wilburn B, Joshi N, Vaish V, Talvala E, Antunez E, Barth A, Adams A, Horowitz M, Levoy M (2005) High performance imaging using large camera arrays. ACM Transactions on Graphics (TOG) 24(3):765–776

Xiong Y, Shafer S (1993) Depth from focusing and defocusing. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 68–73

Yedidia J, Freeman W, Weiss Y (2001) Generalized belief propagation. Advances in neural information processing systems pp 689–695

Zhou C, Lin S, Nayar S (2009) Coded aperture pairs for depth from defocus. In: IEEE International Conference on Computer Vision