Depth-Based Image Segmentation

Nathan Loewke
Stanford University
Department of Electrical Engineering
noloewke@stanford.edu

Abstract

In this paper I investigate light field imaging as it might relate to the problem of image segmentation in cell culture, time-lapse microscopy. I discuss the current field of light field imaging, depth-based imaging segmentation, and light field microscopy. I then discuss the process of gathering data that lends itself well to this problem, calibrating depth map data with ground-truth measurements, generating heat map overlays for quick error estimation, image data segmentation performance, and depth discretization. Finally, I remark on how light field imaging might be applied to the world of microscopy, and in particular, automatic cell tracking.

1. Introduction

Automatic depth detection, segmentation, and object recognition solves a host of problems for photography. Photographers taking pictures with traditional cameras are forced to wait for the camera to determine the best focal point, sometimes after being supplied manual input, which introduces problems like identifying objects at different focal depths or moving objects. Likewise, traditional cameras throw away potentially crucial information from depths outside of the current focal plane, which is exacerbated by using small f-numbers and optical plane thickness.

Likewise, automatic depth-based segmentation could solve critical problems in biological microscopy and cell tracking. Time-lapse microscopy generates far too much data for manual observation or tracking, but automatic cell identification, segmentation, and tracking is still far from a polished realization. Different cell types can take up different shapes and thicknesses, can be difficult to identify and separate due to transparent and sharing nature, can change appearance based on environmental conditions, can be imaged in different modalities, and can occlude one another by roaming over or under one another.

Plenoptic cameras may represent an intuitive solution for depth-based focusing issues and segmentation by using arrays of entire camera devices or of microlenses in front of image plan sensors to capture 4D light field information about a scene. In doing so, image information can be focused after being captured, simulated as being captured by different optics (aperture, camera tilt or rotation, focus spread, etc.), integrated into an all-in-focus image, or used to estimate depth maps, all in a single snapshot (**Fig. 1**). As with anything in optics, these advantages don't come without space-bandwidth tradeoff: The use of microlens arrays to capture incident ray angles as well as accumulated collection means decreased spatial resolution. For example, the gen1 Lytro light field camera comes with an 11 Megaray sensor, but each refocused image is reduced to 1.1664 Megapixels [1].

In this paper I investigate light field imaging as it might relate to the problem of image segmentation in cell culture, time-lapse microscopy. I discuss the current field of light field imaging, depth-based imaging segmentation, and light field microscopy. I then discuss the process of gathering data that lends itself well to this problem, calibrating depth map data with ground-truth measurements, generating heat map overlays for quick error estimation, image data segmentation performance, and depth discretization. Finally, I remark on how light field imaging might be applied to the world of microscopy, and in particular, automatic cell tracking.

2. Related Work

2.1. Light Field Imaging

Light fields are what we refer to as the 4D spatio-angular light ray distribution incident on a 2D light sensor [2]. These light fields may be recorded by introducing some form of parallax to the sensor, either through use of lenslet arrays, single camera translation, or multi-camera gridding. Although the field as a whole has been around for quite some time [3], and is difficult to address fully in limited space, recent advances sparked by consumer products such as Microsoft's Kinect, Sony's Playstation Move, Intel's Realsense, and of course Lytro light field cameras have both increased public awareness of the technology and reduced the barrier to entry for many.

Figure 1. These three, cropped images were refocused digitally after image acquisition from the same 4D image.

In addition to lenslet array based light field imaging, there are a number of technologies used to measure depth maps, or z-position as a function of x,y-position, including stereo triangulation, sheet of light triangulation, structured illumination, time-of-flight imaging, interferometry, and coded aperture.

The field as a whole has many individual components that are not discussed at length here, and are instead, for the most part, tackled by Lytro in the form of their Gen1 product. These concepts might be divided into imaging equations, computing photographs from recorded light fields, digital refocusing, signal processing, selectable refocusing power, digital correction of lens aberrations, and RGB camera sensor sensitivity calibration [4].

In addition, specific advances in more advanced topics have recently pushed the field forward and broadened its research areas into displays [5,6], data compression and sensing [7-10], and image processing and computer vision [11-15].

2.2. Light Field Microscopy

Light field microscopy (LFM) is an even more recent spinoff of the field as a whole, and offers recent success with both biological and nonbiological samples. The application of 4D microscopy with a single sensor offers much of the same tradeoff as with imaging in general: high-speed volumetric acquisition and high temporal resolution, but with reduced spatial and axial resolution. With that said, there have been a few promising applications as of late, including two of the first LFM systems that imaged fluorescent samples including crayon wax [16] and functional neuronal activity [17].

It should be noted that most of the recent LFM work has been devoted to fluorescence imaging, rather than to techniques such as bright field, dark field, or phase contrast. These techniques represent different ways to generate contrast from optically transparent specimen such as cells. The current trend of avoidance of gathering bright field LFM data might be attributed to the difficulty in acquiring observations of transparent, refractive specimen through distortion of reference background patterns alone. One recent method to get around this involves measuring the distortion of light field background illumination [18].

2.3. Depth-Based Image Segmentation

Image segmentation is a challenging and classic problem that has been subject to a huge amount of research activity. Classes of methods can be organized into segmentation problems, clustering algorithms, region merging, level sets, watershed transformations, spectral methods, and texture measurement, among others. One of the reasons this problem can be so difficult is that information content is not always sufficient to recognize an object given its framing. For example, objects of the same color with the same background or occlusion often give robust methods grief.

However, depth data can be segmented easier than can color images and can allow us to discern objects of similar color. There have been many recent, successful approaches that deal with RGB data plus depth information from a few different sources. These span such

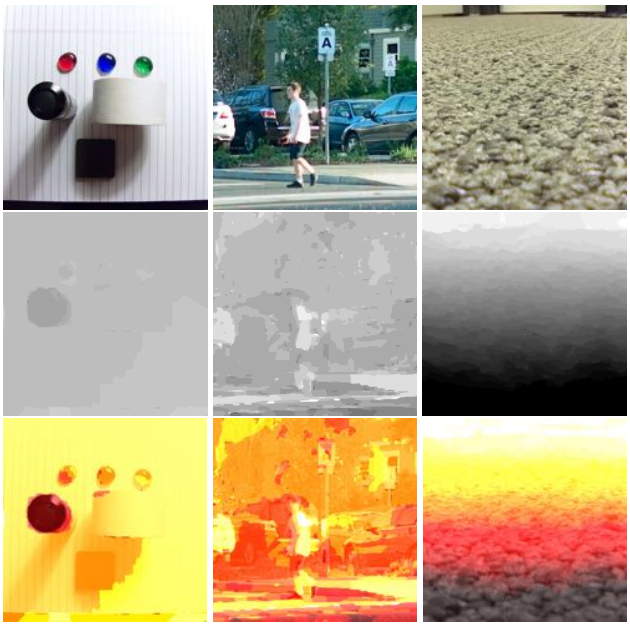

Figure 2. Visualizing the trials and errors of using the Lytro. Converting the depth map into a heatmap overlay allows for easy evaluation of data quality with easy registration. Left column: Relying on image height doesn't work well with the Lytro's depth sensitivity. Middle column: Far-field images don't work particularly well, either. Right column: Relying on high contrast and macro-style shooting produces the best results.

Figure 3. Segmentation results from using Otsu's method (left), local adaptive thresholding (center), and k-means clustering (right) for the initial background separation step. Results show similar results due to careful image staging.

segmentation approaches as geometric segmentation, depth discontinuity, saliency maps, and motion on depth data from a variety of possible sources already mentioned [19-22].

3. Approach

3.1. Acquiring Data

A Gen1 Lytro light field camera was chosen for this project because of its wide availability, economic pricing (I obtained a factory refurbished unit from eBay for about $100), inherent alignment of RGB and depth information, and of course ease of use (as compared to a custom-built setup). I am admittedly not an avid photographer (I deal more with confocal and quantitative phase microscopy), and this was my first approach at using a light field camera, so there was a bit of a learning curve. In fact, I took over 200 random images while walking campus, at my desk, and sitting outside before I was able to achieve consistent results.

It's important to carefully consider what this device is capable and incapable of before discussing data. Unlike the Gen2 Illum or a more scientific device such as a Raytrix R-series camera, this camera forced me into one shooting mode, dubbed "creative mode" which allows the user to tell the camera (via touch on a 1.52 touchscreen LCD) what to focus on. It then performs autofocus on that spot, much as a typical consumer-grade camera would, then chooses a range of focal lengths for everything else in frame. Thus, the extent of the user's control in a controlled setting is choosing what objects are in or out of the FOV. It's also important to note that while in this setting, gain controls are all automatically chosen. Additionally, the camera cannot be programmed to take bursts or timed shots, and cannot be used while plugged into a computer. Once a shot is taken, the camera must be moved to a computer and plugged in, data must be moved to the computer and calculated, and then the depth information may be viewed. This can be particularly troublesome while trying to shoot outside.

Because of the Lytro's relative lack of far-field sensitivity, I was forced to resort to a trick to obtain better segmentation results and a larger spread of depth data. Rather than imaging top-down onto a flat surface and having objects translate in *x*-*y*, I angled the camera slightly down from horizontal and placed images at different distances from the camera. This avoids what I had initially intended, which was to rely on an object's physical height or thickness to generate depth information, but works better with the device that I had.

After experimentation, I settled on two sets of data on which to experiment: black marbles on flat carpet and wooden chess pieces on a game board. The marble dataset consists of five black, glass marbles and was designed to (1) have significant contrast everywhere in the FOV, (2) to be difficult to segment via simple RGB processes alone, (3) to avoid reflections and transparent objects, (4) to include occlusions, (5) to have significant color differences as compared to the rest of the FOV, and (6) to have discrete depths at which to place the objects being segmented. The chess dataset is designed to be similar to the first dataset, but more difficult due to (1) the number of objects of interest, (2) the similarity of color between the pieces and board, (3) the more depth-varying and thus difficult to measure individual pieces.

3.2. Depth Map Quality

As already discussed, under certain circumstances it could be somewhat difficult to determine if a scene was staged correctly, had a proper range of depths in view, had enough discrete depths identified, and had relatively few computational errors. To aid in this, a simple script was

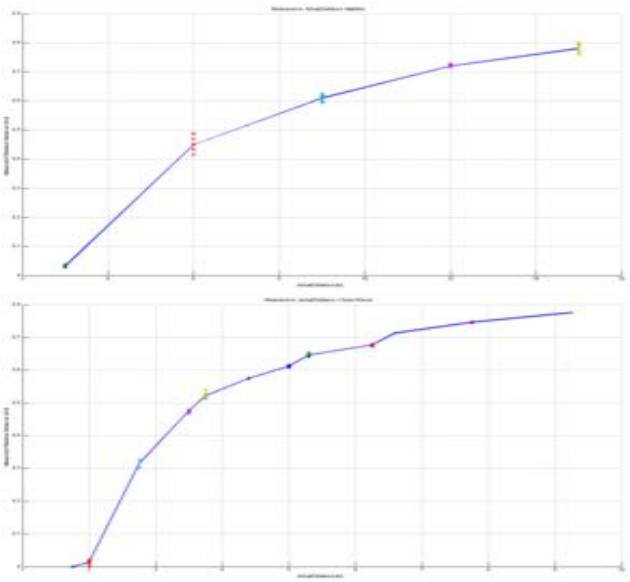

Figure 4. Average, relative depth map value vs. ground truth depth measurement for the marble dataset (top) and chess dataset (bottom). Relationship shows a log-shaped sensitivity.

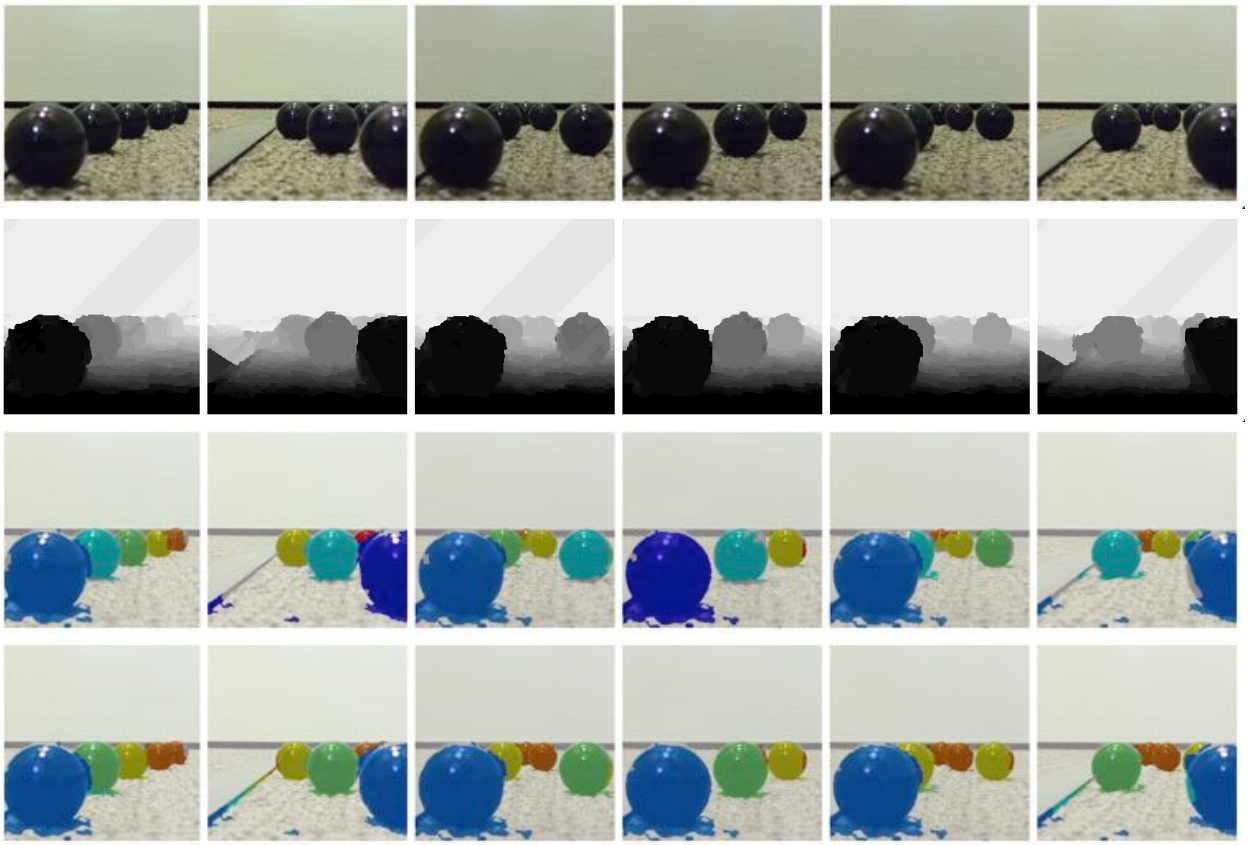

Figure 5. Random samples of depth-based image segmentation on marble dataset. Row 1: Refocused image data. Row 2: Depth maps. Row 3: Results using manually calibrated depth data. Row 4: Results using automatically segmented depth data.

written in Matlab to search through a folder, convert the depth map into a heatmap, and overlay it onto a grayscale representation of the original RGB data. It would be difficult to put a number to how accurate or "good" the images from a certain dataset are, but easy to qualitatively asses under this method. Usable data consists of smoothly transitioning values without the presence of sparse,

randomly positioned patches of disparate values. Unusable data was anything that couldn't show a smooth range of values. One example of each is shown to illustrate (**Fig. 2**).

3.3. Background Removal

I used three different methods for separating the objects in every image from background: Otsu's method, local adaptive binary thresholding, and k-means clustering, each chosen because of their relative simplicity, widespread use, relatively strong performance in the case of staged scenes with controlled colors and low noise.

Otsu's method is an automatic image clustering method for performing binary image thresholding. It works by assuming there are two classes of pixels present in the frame, each belonging to a histogram mode. It then chooses an optimum threshold value that maximizes inter-class variance while minimizing intra-class variance. Local adaptive binary thresholding uses a sliding window of variable, user-defined size that calculates local mean values, and thresholds the local window accordingly. K-means clustering aims to partition n observations into k clusters such that each observation (pixel) belongs to the cluster with the nearest mean.

Because of the careful choice of data type, including attention to colors, contrast, lighting, and reflections, all three methods performed remarkably similar to one another. In fact, the results were so close that I had trouble visually discerning differences in the final product for some images (**Fig. 3**).

3.4. Image Segmentation using Depth Maps

The removal of background signals for each image left us with a single blob of objects merged together and full of occlusions. Separating each object out was a matter of

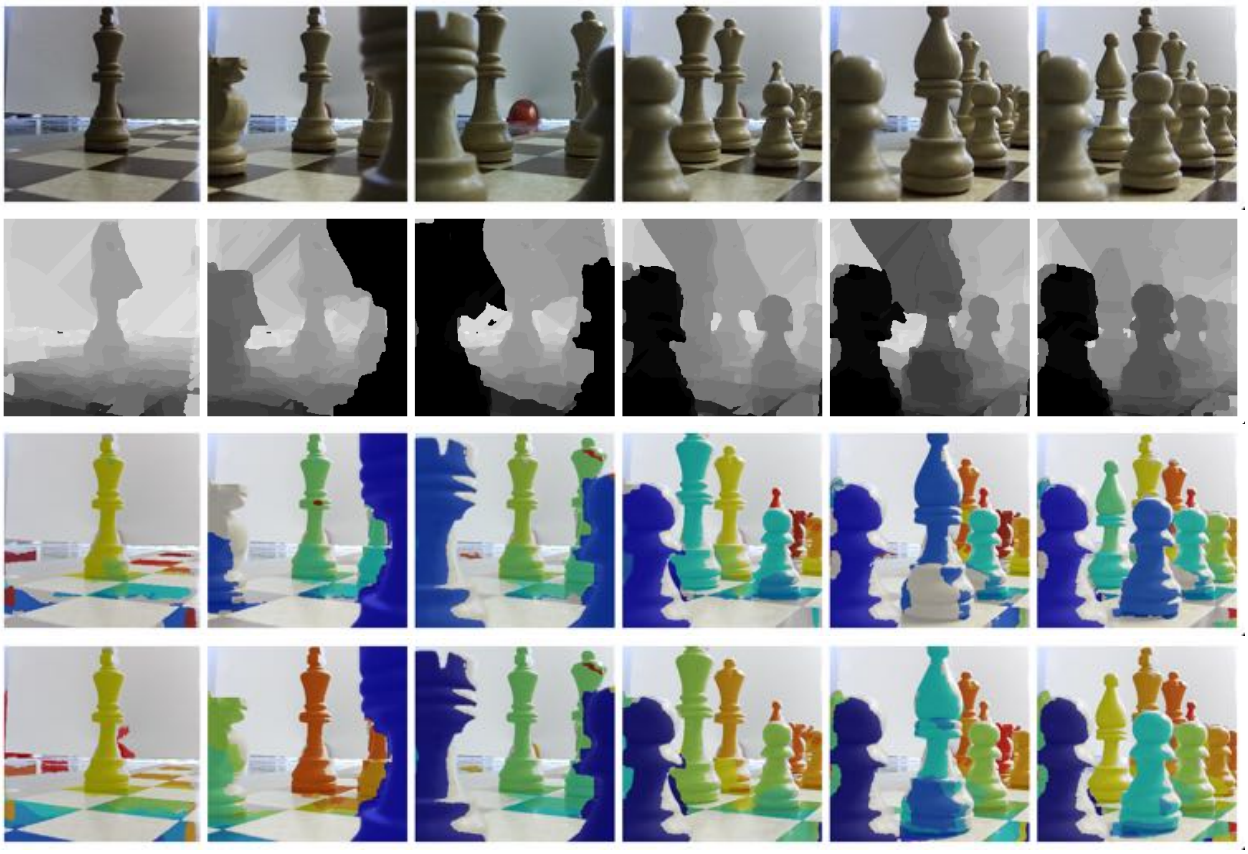

Figure 6. Random samples of depth-based image segmentation on chess dataset. Row 1: Refocused image data. Row 2: Depth maps. Row 3: Results using manually calibrated depth data. Row 4: Results using automatically segmented depth data.

identifying at what depth each piece was placed. From previous experimentation, it was observed that the sensitivity of the depth maps, even under ideal test cases was somewhat noisy. My solution to this was bin the depth data into enough partitions so as to identify all pieces present, but spaced far enough apart so as to avoid noisy, incomplete separation. Two methods were used: (1) calibrated depth measurements and (2) automatic depth clustering via k-means.

Each marble in the first dataset was placed in three-inch increments from the camera, and each chess piece in the second dataset was placed centrally in its corresponding square pad. This system made it possible to move pieces during the photo shoot while keeping the camera still and know each piece's ground truth depth without measuring each individual piece for each image. When going through the calculated depth maps, mean grayscale values of individual pieces were measured manually using ImageJ [23], tabulated, and inputted into Matlab.

The plots of depth-map based relative distance vs. ground truth distance from camera (**Fig. 4**) show two things: (1) the Lytro's sensitivity to distance has a logarithmic scale, and (2) the spread of measurements is never so great as to spill into the next bin. Even the chess piece dataset, which spans a greater distance than the marbles and includes more than twice as many pieces is cleanly separated.

The second method for binning depth data was again using k-means. In this case, a constant value of $k=n_{pieces}+1$ was chosen. I initially had thought that I would need to vary my k-number for the number of pieces that were visible in the scene. However, I later determined this was unnecessary and that both approaches offered similar results.

Each method's depth-map clustering resulted in image masks with clustered values corresponding to bin number, with 0 for background and $1\text{-}n_{pieces}$ for the individual pieces. This mask was then multiplied by its corresponding binary thresholded image. Finally, individual object labels were given a color label according to a jet colormap, and then overlaid onto the grayscale representation of the original RGB data for the purposes of visualization.

4. Evaluation

Figs. 5 and 6 show the final results for both datasets. Each figure shows the refocused (all-in-focus) RGB data, the depth data, the results from using calibrated depth data to separate the depth labels (what we'll call method 1), and results from using k-means to automatically separate the depth labels (method 2).

In the marble dataset, overall accuracy was extremely high regardless of which method was used. Performance tended to degrade slightly toward some of the edges of the marbles, where reflections from my lamp made the marbles appear lighter than they are. The same issue arose around some of the shadows of the marbles, making portions of my carpet labeled as part of the marbles. Additionally, both methods had a harder time toward the back of the frame, where linear changes in distance represented the smallest change in terms of depth map sensitivity.

The biggest advantage here is that occlusion were well handled, regardless of how much one marble is overlapping with another. This can be owed to keeping track of depth values regardless of what's in view, and attributing labels accordingly. Segmentation performance remained high even in cases where entire marbles are completely blocked from view.

For this simple dataset, method 1 slightly outperforms method 2. Discrepencies can most easily be seen when trying to differentiate marbles 4 and 5, toward the back. For example, method 1 may label portions of marble 5 as marble 4, but it never completely misses the target. Method 2, however, completely mislabels marble 5 every time.

In the chess dataset, which was designed to be more difficult, the tables were turned. Accuracy for this dataset as a whole was lower than in the marble dataset, but not discouragingly so. In fact, most of the errors came from labeling chess pieces as combinations of multiple labels, and labeling the floor which did not get filtered out during the simple thresholding stage.

Unlike in the first dataset, method 1 underperformed here. We often see pieces that carry multiple labels. This can be attributed to pieces here having large changes in shape and diameter, which the calibrated data was sensitive to.

Method 2 was the surprise here, out performing method 1 and not displaying as much of the long-range underperformance we saw in the marble dataset. Instead, the k-means depth clustering worked to our advantage and clustered pieces with relatively large size variability together, while still managing to separate out other pieces fairly well.

5. Discussion

It is perhaps not surprising to understand why the data acquisition portion of this project took so much time when we look at Fig. 4. With a log-scaled sensitivity to depth, many of my long-scale shots had no chance of coming out right. The log sensitivity curve of the camera makes sense in hindsight when we consider how depth might be found. When we take a snapshot, we're able to record accumulated light levels and angles of incidence, but not origin (not directly at least). Instead, depth must be indirectly calculated, likely by refocusing the image, computing image gradients, and determining local patches of in-focus data, similar to a standard autofocus algorithm. The only difference here is that our focal stack comes from just one image acquisition.

With that in mind, I was forced to orient my experiments such that I wasn't just relying on object height (much like I would under a standard microscope setup), but instead on physical displacement from the camera's objective, which I had not forseen. If I were to try to use a light field camera for microscopy work, I would either focus on fluorescence imaging, like others have wisely done, or orient the microscope similarly. Either way, a large increase in spatial resolution would be necessary.

The results from calibrated and automatic depth binning are encouraging for the purposes of automatic cell tracking. Having both methods work fairly consistently means this might be an appropriate method for the future. In addition, it's nice that occlusion recognition performance was so high, considering the likely need to orient the microscope's objective similarly to how it was done here.

One particular detail that I had read about the Lytro was that the physical sensor is a standard CCD one might find in any other consumer-grade camera, but cropped so that it's a square 1,080 x 1,080 pixels. This keeps the parallax equal in both vertical and horizontal directions, but likely introduces a slight variance in resolution depending on camera orientation, as the pixels are likely rectangular. I was unable to verify any change in resolution or depth sensitivity depending on camera orientation, which is curious, if what I read is true.

6. Future Work

This project likely pushed the sensitivity of the Gen1 Lytro to its limits, and I thoroughly enjoyed doing so. In the future, I'd like to try a higher resolution camera such as the Illum. But in particular, I'd like to try a camera that I can control to a much greater extent, e.g., to take pictures at programmable temporal resolution with constant gain and adjustable depth range. This would enable a proper time-lapse dataset acquisition, as I originally intended, and allow me to try motion based segmentation algorithms. I'd also love to try to adapt this or another microscope to perform microscopy, and see if I could use the diffraction

signal patterns in the light field to identify and segment individual cells in culture, similar to [18].

References

[1] "Lytro Gen1 Technical Specifications, Lytro, Inc." Lytro. N.p., n.d. Web. 19 Mar. 2015. <https://www.lytro.com/camera/specs/gen1/>.

[2] M. Levoy and P. Hanrahan. Light Field Rendering. In Proc. Siggraph, pages 31–42, 1996.

[3] T. Okoshi. Three-Dimensional Imaging Techniques. Academic Press, 1976.

[4] Ng, Ren. Digital Light Field Photography. Thesis. Stanford University, 2006. N.p.: n.p., n.d

[5] Ruigang Yang; Xinyu Huang; Sifang Li; Jaynes, C., "Toward the Light Field Display: Autostereoscopic Rendering via a Cluster of Projectors," Visualization and Computer Graphics, IEEE Transactions on , vol.14, no.1, pp.84,96, Jan.-Feb. 2008.

[6] Wetzstein, G.; Lanman, D.; Hirsch, M.; Heidrich, W.; Raskar, R., "Compressive Light Field Displays," Computer Graphics and Applications, IEEE , vol.32, no.5, pp.6,11, Sept.-Oct. 2012.

[7] Xin Tong; Gray, R.M., "Interactive rendering from compressed light fields," Circuits and Systems for Video Technology, IEEE Transactions on , vol.13, no.11, pp.1080,1091, Nov. 2003.

[8] Chuo-Ling Chang; Xiaoqing Zhu; Prashant Ramanathan; Girod, B., "Light field compression using disparity-compensated lifting and shape adaptation," Image Processing, IEEE Transactions on , vol.15, no.4, pp.793,806, April 2006.

[9] Kitahara, M.; Kimata, H.; Shimizu, S.; Kamikura, K.; Yashima, Y., "Progressive Coding of Surface Light Fields for Efficient Image Based Rendering," Circuits and Systems for Video Technology, IEEE Transactions on , vol.17, no.11, pp.1549,1557, Nov. 2007.

[10] Magnor, M.; Girod, B., "Data compression for light-field rendering," Circuits and Systems for Video Technology, IEEE Transactions on , vol.10, no.3, pp.338,343, Apr 2000.

[11] Lifeng Wang; Lin, S.; Seungyong Lee; Baining Guo; Heung-Yeung Shum, "Light field morphing using 2D features," Visualization and Computer Graphics, IEEE Transactions on , vol.11, no.1, pp.25,34, Jan.-Feb. 2005.

[12] Kubota, A.; Aizawa, K.; Tsuhan Chen, "Reconstructing Dense Light Field From Array of Multifocus Images for Novel View Synthesis," *Image Processing, IEEE Transactions on* , vol.16, no.1, pp.269,279, Jan. 2007.

[13] Chia-Kai Liang; Yi-Chang Shih; Chen, H.H., "Light Field Analysis for Modeling Image Formation," *Image Processing, IEEE Transactions on* , vol.20, no.2, pp.446,460, Feb. 2011.

[14] Raghavendra, R.; Raja, K.B.; Busch, C., "Presentation Attack Detection for Face Recognition Using Light Field Camera," *Image Processing, IEEE Transactions on* , vol.24, no.3, pp.1060,1075, March 2015.

[15] Dansereau, D.; Bruton, L.T., "A 4-D Dual-Fan Filter Bank for Depth Filtering in Light Fields," *Signal Processing, IEEE Transactions on* , vol.55, no.2, pp.542,549, Feb. 2007

[16] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz, "Light Field Microscopy," ACM Trans. Graph. 25(3), 924–934 (2006).

[17] Prevedel, R., et al., "Simultaneous whole-animal 3D imaging of neuronal activity using light-field microscopy," Nature Methods, July 2014.

[18] Wetzstein, G.; Roodnick, D.; Heidrich, W.; Raskar, R., "Refractive shape from light field distortion,"*Computer Vision (ICCV), 2011 IEEE International Conference on* , vol., no., pp.1180,1186, 6-13 Nov. 2011.

[19] Han, B.; Paulson, C.; Wu, D., "Depth-based image registration via three-dimensional geometric segmentation," *Computer Vision, IET* , vol.6, no.5, pp.397,406, Sept. 2012.

[20] Joshi, G.; Sivaswamy, J.; Krishnadas, S.R., "Depth Discontinuity-Based Cup Segmentation From Multiview Color Retinal Images," *Biomedical Engineering, IEEE Transactions on* , vol.59, no.6, pp.1523,1531, June 2012.

[21] Ji-Eun Lee; Rae-Hong Park, "Segmentation with saliency map using colour and depth images," *Image Processing, IET* , vol.9, no.1, pp.62,70, 1 2015.

[22] Sekkati, H.; Mitiche, A., "Concurrent 3-D motion segmentation and 3-D interpretation of temporal sequences of monocular images," *Image Processing, IEEE Transactions on* , vol.15, no.3, pp.641,653, March 2006.

[23] Rasband, W.S., ImageJ, U. S. National Institutes of Health, Bethesda, Maryland, USA, http://imagej.nih.gov/ij/, 1997-2014.