Panorama Based on Light Field Images

Kai Zhang
Department of Electrical Engineering
Stanford University
Stanford, USA
kzhang3@stanford.edu

Ziyi Yang
Department of Electrical Engineering
Stanford University
Stanford, USA
ziyi.yang@stanford.edu

***Abstract*—We have developed a method to calculate panorama with post-focusing ability from light field images. Firstly, light field information was extracted by Lytro desktop software. Then depth map was calculated from focal stack and perspective views via three different methods: locally standard deviation (STD), Laplacian filter, and gradient-based method. Secondly, using the information from depth map, we successfully stitched panorama at multiple focus. Finally, we produced anaglyphs from the left-most one and right-most one in the sub-image array and stitched anaglyphs into panorama.**

Keywords—Plenoptics, Light Field, Panorama, Anaglyph

I. Introduction

Recently, light field (plenoptics) imaging has been a very heated topic because of its post-focus ability and richer information (for example, depth information) than convention 2D picture. Inspired by this novel technology, we want to generate panorama from light field (plenoptics) images. For instance, panorama of a scene at a specific focus.

Figure 1 [3] is an illustration of lenslet-based plenoptic camera. Light coming from right is focused by the main lenslet (blue) onto the lenslet array and then split by sub-lenslets.

Lytro also offers a powerful tool to manipulate light field images: Lytro Desktop. By using Lytro desktop, we can export the depth map and focal stack (same image at different focus) from light field file.

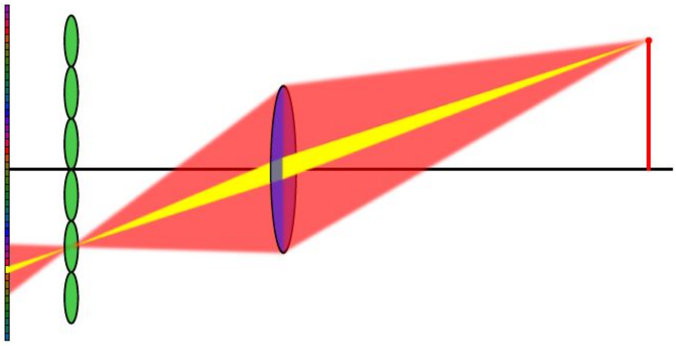

Fig. 1. Device Illustration and Diagram of Plenoptics Imaging

Light Field toolbox V0.4 developed by Dr. Dansereau is another strong tool to process light field images. One can do calibration, visualization and process of light field imaging system.

Anaglyphs are generated from a pair of images from different perspectives. Light field imaging has provided an intuitive way to produce the pair. From the sub-image array, pick the leftmost and rightmost image, and then they are two images of the same scene but from different perspectives.

II. related Work

A. *Light Field Imaging*

One of the very first papers on light field imaging is “light field rendering” [1] by Prof. Levoy and Prof. Hanrahan here in 1996. In this paper, 4D light field images is generated by an array of cameras. Each 2D images is interpreted as slices of 4D images – the light field.

In Lytro’s founder Dr. Ng’s doctoral dissertation [6] and his other publications [7], we can find those ideas leading to off-shelf plenoptics cameras, Lytro cameras that one wants to collect more information from light field imaging than conventional 2D imaging, while saving storage and keeping process algorithm efficient.

B. *Light Field Panorama*

There are only a few literatures ([2], [3]) on light field panorama, maybe due to lack of efficient methods to detect features. In [2] authors reproduce “light field panorama” from focal stack and failed to produce an 4D panorama. In [3], authors only conducted simulation and did not show the final result of panorama.

III. Rotation Axis & Camera Calibration

Before we shot the images and stitched the panorama, two things we needed to do were deciding the rotation axis and calibrating the camera. Rotation axis was always one key problem in panorama and it directly decided whether we can stitch images together. Camera calibration estimated the character of a lens and image sensors. And then these parameters were used to eliminate lens distortion.

A. *Decide Rotation Axis*

Figure 2 shows the first-generation Lytro camera atop the rotation stage. And in real experiment, the camera was tied to the rotation stage to eliminate shaking effect by tape. And this was not a highly-accurate rotation stage so that we could not precisely control the rotation angel. But it should not affect following progress.

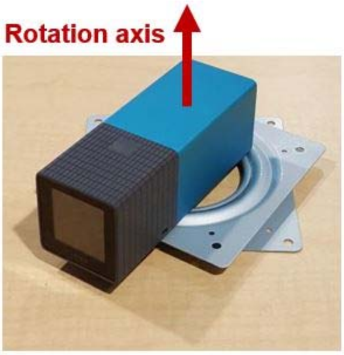

Fig. 2. Lytro camera amounted on rotation stage with rotation axis marked

Since Lytro did not give out the inner structure parameters of its first generation camera, one way to determine the rotation axis was via motion parallax.

One non-technical but vivid explanation of this method would be: think one of your eyes as camera. Reach out your hand and point up a finger and stare at it with only one eye open. Turn your head around your spine, and you can see that there is relative motion between your finger and a thing in the background, or your finger is move "faster" than things in the background and this is what we call motion parallax. Now, instead of turning your head around your spine, move your head around the center of the eyeball you are watching your finger with. (I know it can be not easy, but try it.) Now maybe you can see that there is no relative motion, or so-called no motion parallax. This is because now your eye is rotating around its projection, approximate the place of retina in eye.

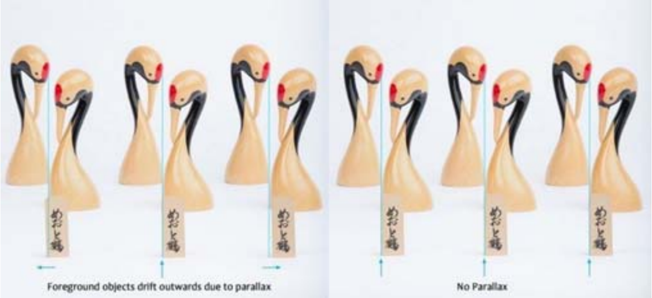

Fig. 3. An example of motion parallax [4]. We can tell whether if the rotation axis is the projection center by examining if the board in the foreground moves relative to the swans in the background.

Fig. 3 shows an example of motion parallax. In sum, when we rotated the camera, if there did not exist motion parallax, then this rotation center should be the projection we are looking for. We understand this was not one hundred percent accurate method since we judge the motion parallax by naked eyes, but from the outcome of panorama, this method should work very well.

B. *Camera Calibration*

We used the classic chess board to estimate intrinsic, extrinsic, and distortion coefficients. And the calibration process followed as described in the documentation file of light field toolbox V0.4. Fig.4 shows calibration result. The picture on the right is plenoptics image decoded without calibration. The right one is result decoded with calibration. The difference between them is not distinct and it means the camera we use is in pretty good condition.

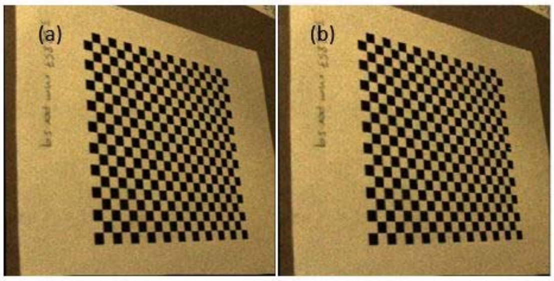

Fig. 4. Calibration result.

IV. Depth Map

From Lytro Desktop we extracted focal stack (images at different focus). And we proposed three methods to produce

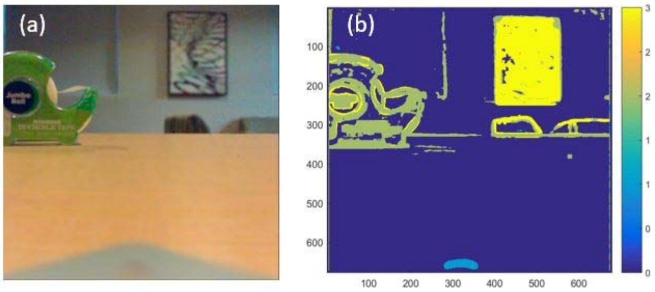

Fig. 5. Depth map obtained by local STD method. (a): 2D image of origin scene. (b): Depth map derived by local STD method

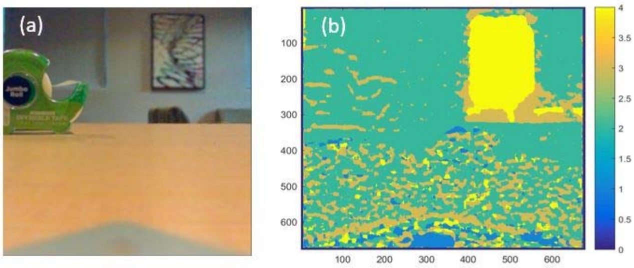

Fig. 6. Depth map obtained by Laplacian filter. (a): 2D image of origin scene. (b): Depth map derived by Laplacian Filter

depth map form focal stack: local STD, Laplacian filter and gradient method.

A. *Local STD*

In focal stack, if an area is not focused, it will look fussy. It's like the area is convolved by a Gaussian filter. In this method, we determined the depth of an area by calculating the local STD of an area around each point. Then we continued to calculate the STD of that point in all the images from the focal stack. The largest STD would correspond to the most focused image. We ranked the images from focal stack in a way from near-focused to far-focused so that once we knew the corresponding relation of an image with the area, we would know the relative depth of the area. The result is shown in fig. 5.

B. *Laplacian Filter*

Again, we used focal stack to calculate the depth. If we convolve an image with Laplacian filter, the sharp features will appear as white points in the result image. In this method, we summed up the values around one point to determine the contrast value of that specific point. We compared the contrast value of one specific point among all the images in the focal stack and determine the image that has the largest value. This image would be the one that this point being the sharpest. Again, by ranking the images from near-focused to far-focused, we could correspond one point to a relative depth. The result is shown in Fig. 6.

C. *Gradient Method*

We used perspective views to calculate the depth. From light field data, we could extract perspective views of a scene. If an object was away from us, it would move for a longer distance between the two perspective views while a near object will be the opposite. Thus, we could calculate the relative depth of every point by comparing how far it moves in the two perspective views. The result is shown in Fig. 7.

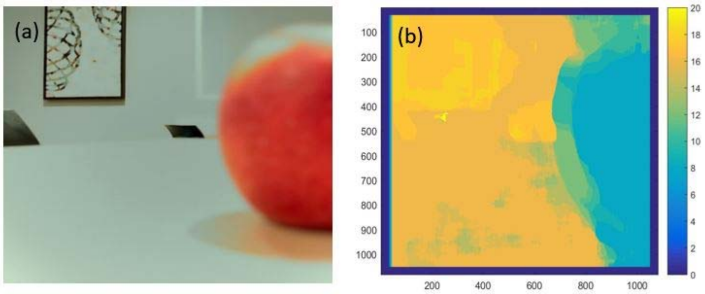

Fig. 7. Depth map obtained by Gradient Method. (a): 2D image of origin scene. (b): Depth map derived by gradient method

D. *Discussion*

As we compared the three methods of depth detection we found that smooth area like table/white wall will be non-trivial to evaluate. If we used focal stack, the white wall did not change a lot in images with different focus thus Laplacian method and local STD would lead to some wrong depth detection of these areas (i.e. some part of a white wall far away can be considered close to the camera). What we can do here is to distinguish the white wall from an object with more features by looking at how local STD in an area has changed in images within the focal stack. The local STD change of a flat area will be way lower than areas with more features.

Comparing the two methods using focal stack (Laplacian filter and local STD) to detect depth, we found local STD method is more robust than Laplacian filter. It would lead to less outliers in an object. Both methods work for images with apparent segmentation of the objects with one near, one middle and one far away. If the image had some continue structures like a crowd of people, the outliers will increase.

V. Panorama

In this section we will show how to stitch panorama using information obtained above.

A. *Stitching by Focal Stack*

Say we had a target image and a focal stack of image to stitch. In this case, the focus of focal stack was at near, middle and far. (Fig. 9) We developed an algorithm to automatically select the image with closest focus with target image from focal stack. The algorithm is following:

Calculated difference between STD in region around one SIFT feature of target and image to stitch, and then sum these differences up to obtain Δ . I.e. $\Delta = \sum_{feature} (std_1 - std_2)/std_1$. We chose the picture with smallest Δ to stitch. (In this example is the near-focused one). Since the STD of one region reflects the level of fussy, this algorithm actually picks out the image whose features have the

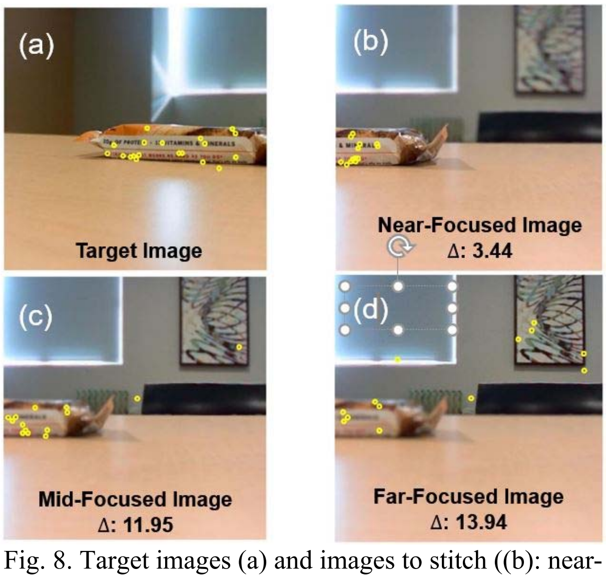

Fig. 8. Target images (a) and images to stitch ((b): near-focused, (c): mid- focused and (d): far-focused with Δ marked at the bottom of image). Yellow circles are features by SIFT, and we set the number of features to be 20.

closest level of fussy with the target image.

B. Panorama

1) Match Two Images Together

We had successfully produced the panorama (Fig. 9) of near-focused and far focused. After we found a focused-match image from the focal stack with the target image. We used the following algorithm to stitch the two images together. First, we swept the second image two dimensionally around the target image. We subtracted the overlapped area of the two images and calculated the L2 difference in the overlapped area as a measurement of whether the second image's current position really matched the target image.

To accelerate the process, we used a two level sweeping method. We carried out a rough scan on the image which is followed by a fine scan. Suppose the image had 900 pixels. We divided it by 21 which means we firstly scanned 43 points and each of them will be separated by 21 pixels. After we found a minimum L2 difference position of these 43 points, we would carry out a second level local sweep of the 21*2 points around the selected point. This method will help us find the most matched position of target image and second image with a total of (43+2*21)=85 scans instead of 900. Since we needed to scan both horizontally and vertically, this method will be 100 times faster than scan the 900 pixels one by one. As a result, it only took us a couple of seconds to find the most matched position of the two images.

2) Images Fusing

Since the brightness of the two images coule be different, we needed to change the brightness in the overlapping area in order to remove any apparent boundaries between the two stitched images. First, we found the overlap area between the two images. For the left image, we linear reduced the brightness in the overlapping area from 1 to 0 while for the right image we linearly increased the brightness in the overlapping area from 0 to 1. The above process was achieved by applying a mask to the original image. The brightness of the left mask and right mask was gradually changed as the following Fig. 10(a) and 10(b) show. The results after applying of the masks are also shown in Fig. 10(c) and Fig. 10(d).

3) Discussion

Using the method described above, we had successfully produced the panorama (Fig. 10) of near-focused and far focused. The panoramas look good, which means the rotation axis is in correctly chosen, and the method in section A works.

VI. Anaglyphs

The intrinsic property of light field imaging can naturally produce anaglyphs. Since its 2D slices are from different perspectives. After we decoded the light field image by light field toolbox, we obtained a 5D file LF. (5D instead of 4D because RGB channel.) Apply MATLAB function *squeeze* to this 5D file, we produced 3D (considering RGB channel) slices from different i, j, k, l position. Picked the leftmost and right most image and then use them to produce anaglyphs. The result is shown in Fig. 11.

And then stitched anaglyphs and into panorama (Fig. 12).

VII. Conclusion

In conclusion, we found the projection center of the Lytro camera and carried out camera calibration. From focal stack and perspective views, we have produced depth map by three methods: locally STD, Laplacian filter, and gradient-based method. From focal stack, we developed an algorithm to pick up the most focus-matched image with the target image. And then we produce panorama at different focus by the accelerated stitching algorithm. Finally, anaglyphs are generated from the leftmost image and rightmost image from sub image array and stitched into panorama.

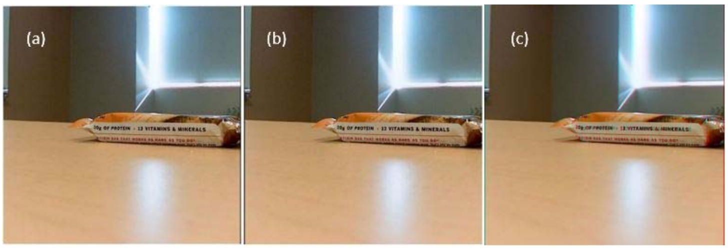

Fig. 11. Anaglyphs. (a): leftmost image. (b): rightmost image. (c): Anaglyphs generating from (a) and (b).

VIII. Future Work

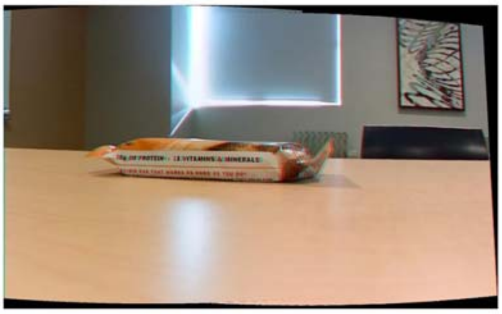

Fig. 12. Panorama of anaglyphs.

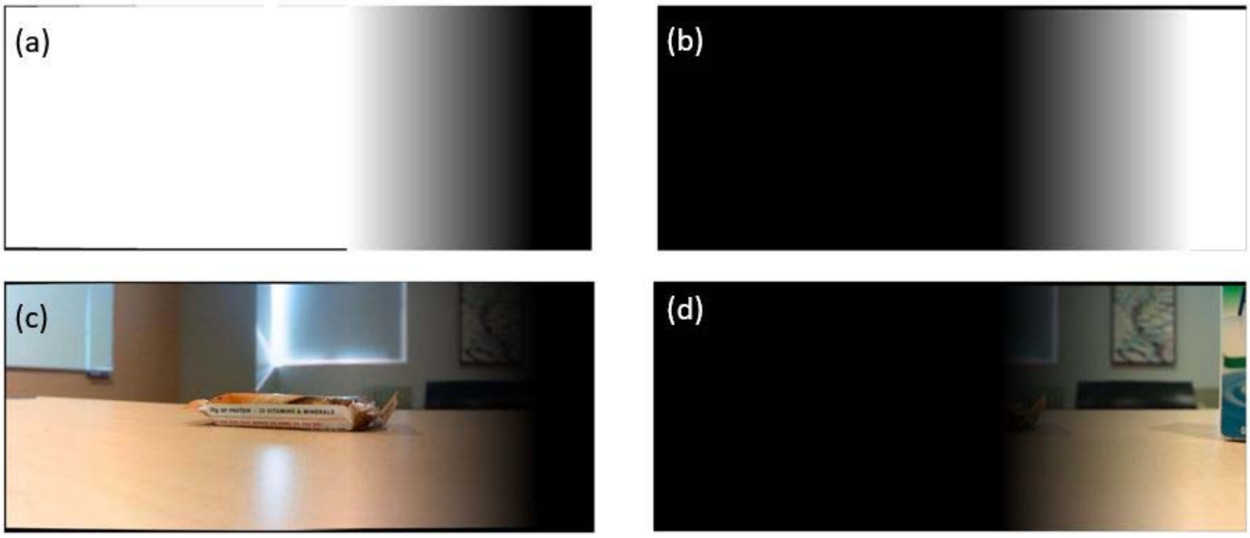

Fig. 9. Linear Blending. (a): Mask of target image with linear blending. (b) Mask of image to stitch with linear blending. (c) Target image with linear blending. (d) Image to stitch with linear blending

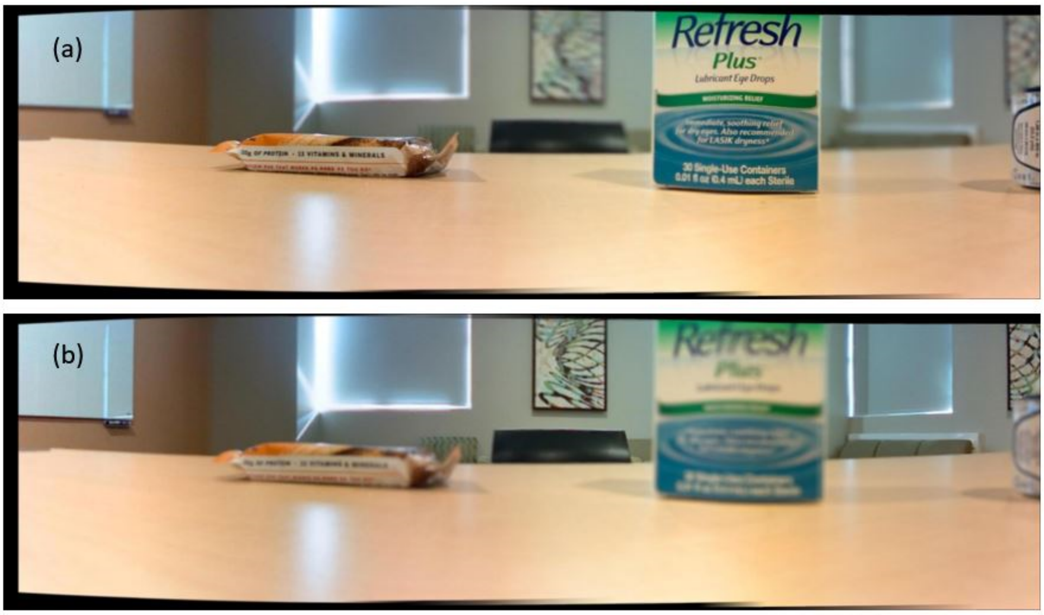

Fig. 10. Panorama. (a): Panorama of near-focused image. (b) Panorama of far-focused image.

Although we have successfully stitched panorama of scenes at different focus, 4D panorama is still out of scope. In order to achieve that, firstly, we need to develop an algorithm that detect feature of 4D images, just like the SIFT for 2D image. Note that this should be an efficient algorithm, considering existent method's large running time. Secondly, find the homograph matrix between images. In 2D images, homograph matrix is 3×3, and in light field domain, homograph matrix should be 5×5.

Acknowledgment

We'd like to thank Dr. Donald Dansereau for mentoring this project. He has been very helpful and given very insightful advice. And we also want to thank Prof. Gordon Wetzstein and TAs for their efforts through this quarter.

References

[1] Levoy, Marc, and Pat Hanrahan. "Light field rendering." Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. ACM, 1996.

[2] Birklbauer, Clemens, and Oliver Bimber. "Panorama light‐field imaging." Computer Graphics Forum. Vol. 33. No. 2. 2014.

[3] Xue, Zhou, et al. "Light field panorama by a plenoptic camera." IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics, 2014.

[4] M. Bailey, "Shooting for stitched panoramas & avoiding Parallax shift (Podcast 452)," in Martin Bailey Photography, 2014. [Online].

[5] Yang, Jason C., et al. "A Real-Time Distributed Light Field Camera."Rendering Techniques 2002 (2002): 77-86.

[6] Ng, Ren. Digital light field photography. Diss. stanford university, 2006.

[7] Ng, Ren, et al. "Light field photography with a hand-held plenoptic camera."Computer Science Technical Report CSTR 2.11 (2005): 1-11.

[8] D. G. Dansereau, "Plenoptic signal processing for robust vision in field robotics," Ph.D. dissertation, Australian Centre for Field Robotics, School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, Jan. 2014.

[9] Kim, Changil, et al. "Scene reconstruction from high spatio-angular resolution light fields." ACM Trans. Graph. 32.4 (2013): 73-1

Appendix

Kai did the camera rotation axis, depth map and stitch panorama from focal stack, part of report. Ziyi did rotation axis, camera calibration, anaglyphs panorama, writing of poster and report.