



IIT MADRAS

EE5176

COMPUTATIONAL PHOTOGRAPHY

Light Field Photography

Team:

Abhishek Naik
Shiva Krishna Reddy
YSSV Sasikiran

Roll No:

CS13B030
CS13B051
CS13B055

1 Abstract

This project involves exploring the concepts of light field photography and its wide range of applications. We simulate the capture of light field by synthetic image generation software, POV-Ray[9]. We then look at applications of Light Field Photography which include refocusing using Fourier Slice Theorem[4], synthetic aperture, looking behind an object, depth map estimation using focal stack. We finally develop an interactive GUI for visualizing the light field and its applications.

2 Introduction

Conventional cameras only capture images that contain only the information about intensity of light in the underlying scene. In other words they sample only the 2D spatial component of the plenoptic function[10]. Light field photography[4] captures the angular components of the image along with the spatial intensity. Many commercial implementations of the camera such as plenoptic camera[5], lytro camera[6] are available. However, we use images generated by synthetic means as the light field. We also use some of the light field datasets[8] provided by Stanford graphics team.

3 Tasks

3.1 Synthetic Image Generation

We use Persistence of Vision Ray Tracer ([POV-Ray](#)) for generating synthetic images of light fields. POV-Ray is a ray tracing program which generates images from text-based description of scenes. We simulate light field capture by 'keeping' cameras at different positions in the code.

We first shift the original scene camera along the X-axes by a predefined distance of separation to generate a camera array on X-axes. The look-at position of each of these camera is similarly shifted by the same amount. We then find the direction perpendicular to the look-at direction of the cameras and shift the camera in that direction by the same amount to obtain an array of cameras.

The rendering time depended on the complexity of the scene. For example, the glassware in Figure 1(d)[11] took over 20 hours to render, whereas the one with the die (Figure 1(c)) took just 10 seconds per image.



(a) Crate of oranges



(b) Waterfall scene



(c) Some die



(d) Glassware

Figure 1: Some POVRAY results

Calibration

The calibration of synthetically generated light field is done as suggested in [1]. We first place a synthetic square in the scene at the depth which we want to be in focus and calculate the displacement of the square in each of the light field images. We then shift each of the light field images by the same amount as the displacement. This is done to ensure all the objects at the required depth come into focus on just averaging the light field, resulting in the conventional photograph.

3.2 Refocusing

The calibrated images obtained from the above are just shifted versions of the original image but with one small important difference. One particular part P of the scene is at the same coordinate position in all the images. Hence by averaging all



Figure 2: The black square in (a) and the white square in (b) were used for calibration of the light fields.

the angular images we get a refocused image, focused at P . Similarly by shifting the image at (i, j) with $(k.i, k.j)$ pixels in X and Y direction respectively we get a refocused image which is focused at some other point P' . To account for magnification, we use the equation proposed in [4]. The shifting equation is then given by:

$$P_\alpha[L_F](x, y) = \frac{1}{\alpha^2 F^2} \int_u \int_v L_F \left(u(1 - \frac{1}{\alpha}) + \frac{x}{\alpha}, v(1 - \frac{1}{\alpha}) + \frac{y}{\alpha}, u, v \right)$$

However this integration is of the order of $O(n^4)$ for each refocusing. Integrations in the spatial domain can be efficiently obtained by taking the slice in the fourier domain as proposed in [4]. Hence we take the 4D Fourier transform of the light field and take then perform a change of basis to achieve the refocussed light field. The refocused image can be obtained by taking the inverse fourier transform of the slice of the 4D fourier domain.

Using the notation in [4] directly we get:

$$P_\alpha = \mathcal{F}^{-2} \circ \mathcal{B} \circ \mathcal{F}^4$$

$$\mathcal{B}_\alpha[G](k_x, k_y) = \frac{1}{F^2} G(\alpha.k_x, \alpha.k_y, (1 - \alpha)k_x, (1 - \alpha)k_y)$$

The advantage of this method is that once the 4D fourier transform has been computed which is $O(n^4 \log n)$, the slices and their inverse fourier transform can be computed in $O(n^2 \log n)$. This results in significant savings in time.

3.3 Synthetic Aperture

By using only a sub-grid of the images in the light field, it is possible to obtain an image that simulates that obtained by a conventional camera with the corresponding aperture. Refocusing can then be performed on them.

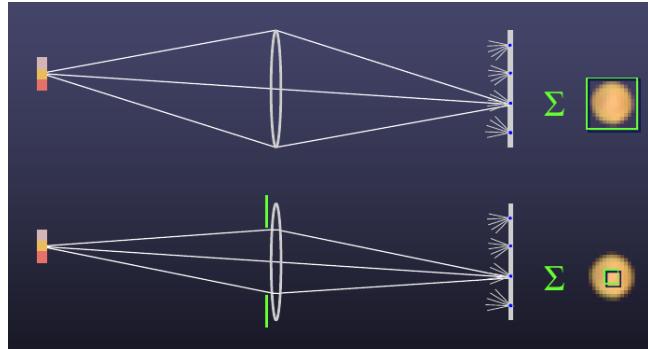


Figure 3: Stopping down by summing only a sub-grid of the light field views

3.4 Depth Estimation

Usual stereo vision uses only two views of the scene. Since we have a full light field image, we can do much better. Firstly the refocused images at various depths are obtained. Then for every 15×15 patch, we find the focus measure of each refocused image. The level with highest measure is chosen. Here we use energy of the gradients method proposed in [2] and implemented by [3]. The depth estimated is smoothed using Markov random fields[7].

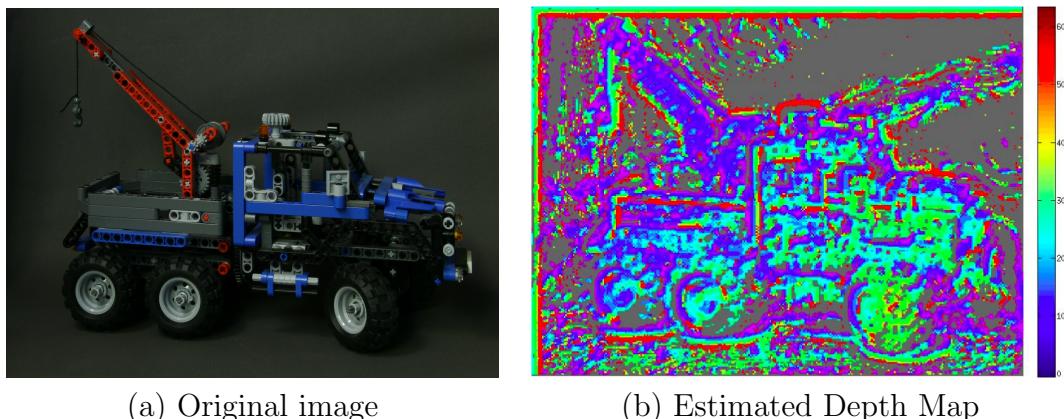


Figure 4: The depth map calculated for the Stanford Truck Data.

3.5 Looking behind an object

For images where an object in the front blocks the object which is of interest, some part of the object of interest is clear in atleast one image. Thus by using the information from all the images, one can sufficiently reconstruct the image where the object of interest is perfectly in focus. For scenes with sufficiently good depth information, by refocusing at a particular depth, the other parts of the image can be blurred out. This is demonstrated.



(a) Original image



(b) Looking ‘behind’

Figure 5: The scene to demonstrate ‘looking behind objects’

3.6 Light Field Interactive GUI

An interactive GUI was developed to explore the light field and all its implemented applications. Some of the screenshots of the GUI for Stanford truck are shown in Figure 6, with a different point-of-view selector, digital refocusing slider, and a synthetic aperture.

4 Challenges Faced

4.1 Fourier Slice Theorem and its implementation

While refocusing using Fourier slice photography, few artifacts arise in the refocused image such as roll-off and aliasing artifacts. The artifacts arise due to approximating the band limit filter such as *sinc* with a finite extent filter. Roll-off effects arise

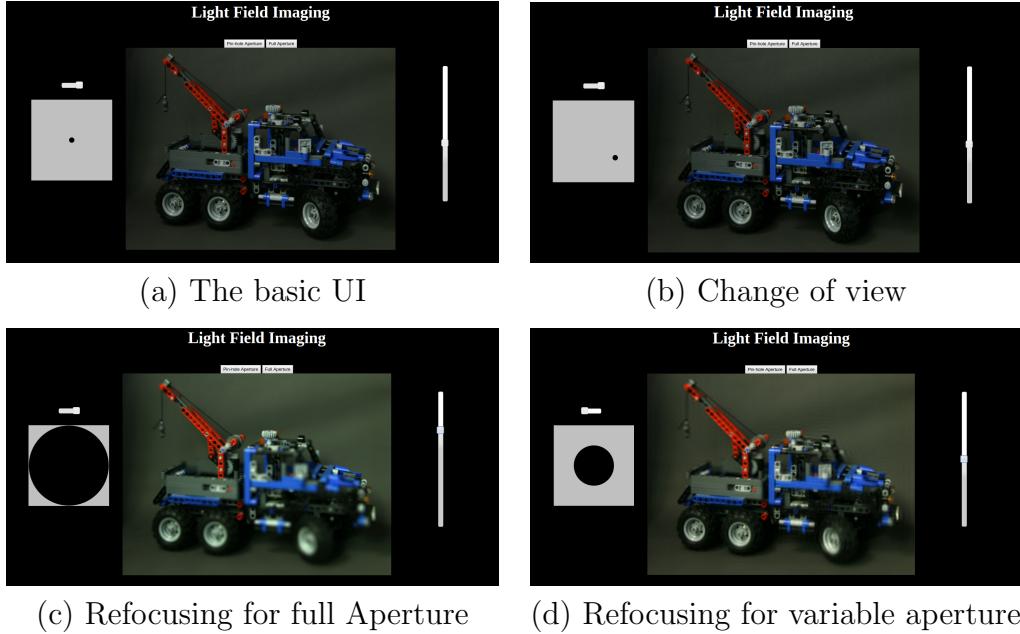


Figure 6: Few screen shots of the interactive GUI for Stanford Truck data.

because the filter is not unit value in the band limit. Due to this the borders of the refocused image get darkened. Aliasing artifacts occur due to presence of the filter beyond the band limit. This results in periodic replicas in the refocused image.

Roll-off was eliminated by pre-multiplying the light field signal with the reciprocal of the inverse Fourier transform of the filter. Aliasing artifacts were handled using oversampling, superior filtering and appending zeros to the light field. Both these methods were suggested in [4]. In superior filtering, we used a simple Gaussian filter rather than Kaiser-Bessel as suggested in [4]

Figure 7 and Figure 8 show the effects of artifacts and the resulting image obtained after removing the artifact.

4.2 Computational power

None of the team members had a CUDA-enabled NVIDIA GPU in their laptops. This lack of computation power resulted in long delays in obtaining and validating the results using Matlab.



(a) Before Oversampling

(b) After Oversampling

Figure 7: Artifact removal via Over Sampling.

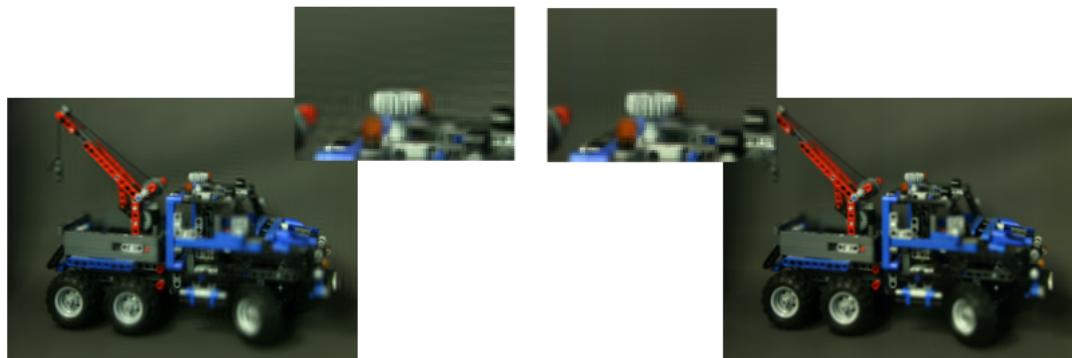


Figure 8: Artifact removal via Gaussian Interpolation.

4.3 Extensive parameter tuning

Each different light field image set required tuning of parameters - the refocusing parameter α , the radius of the gaussian interpolation of Fourier slice, and finally, the extent to which the image was to be downsampled, for obtaining results in a reasonable amount of time.

4.4 Memory constraints

The team only had 8GB of RAM in their laptops, which made it impossible to some of the datasets with a oversample the 2D Fourier slice with a factor of 2.

References

- [1] Calibration of dense camera arrays. http://graphics.stanford.edu/papers/plane+parallax_calib/paper.pdf.
- [2] Depth from defocus: A spatial domain approach. <http://link.springer.com/article/10.1007/BF02028349>.
- [3] Focus measure matlab implementation. <https://in.mathworks.com/matlabcentral/fileexchange/27314-focus-measure>.
- [4] Fourier slice photography. <https://graphics.stanford.edu/papers/fourierphoto/fourierphoto-600dpi.pdf>.
- [5] Light field photography with a hand-held plenoptic camera. <http://graphics.stanford.edu/papers/lfcamera/>.
- [6] Lytro light field cameras. <https://www.lytro.com/>.
- [7] Markov random field optimisation code-peter orchard. http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV0809/ORCHARD/.
- [8] The (new) stanford light field archive. <http://lightfield.stanford.edu/>.
- [9] Persistence of vision pty. ltd. (2004). <http://www.povray.org/>.
- [10] The plenoptic function and the elements of early vision. http://persci.mit.edu/pub_pdfs/elements91.pdf.
- [11] Povray hall of fame. <http://hof.povray.org/>.