Coded Aperture Light Field Reconstruction Using Deep Learning

A Project Report

submitted by

ATCHUT NAVEEN CH, EE14B018

in partial fulfilment of requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

MAY 2018

THESIS CERTIFICATE

This is to certify that the thesis titled **Coded Aperture Light field Reconstruction**, submitted by **Atchuth Naveen CH, EE14B018**, to the Indian Institute of Technology Madras, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Kaushik Mitra
Project Guide
Assistant Professor
Dept. of Electrical Engineering
IIT Madras, 600036

Place: Chennai

Date: 10 May 2018

ACKNOWLEDGEMENTS

This work would not have been possible without the guidance and the help of several people. I take this opportunity to extend my sincere gratitude to all those who made this thesis possible. First, I would like to thank all my teachers who bestowed me with good academic knowledge. I am indebted to my advisor Prof. Kaushik Mitra whose expertise, generous guidance and support made it possible for me to work on a topic that was of great interest to me. I would also like to thank my lab mate and dear friend Anil Kumar Vadathya for sharing his valuable ideas and helping me whenever I am stuck with some problem. I would like to thank my family for giving support and guidance all through my life. I would also like to thank all my friends and well-wishers for helping me in difficult times and being a good source of support and guidance.

ABSTRACT

KEYWORDS: Light Field, Coded Aperture, Disparity Map, Compressive sensing, Code Design, Reconstruction

Consumer cameras have long been trying to use light field and provide a rich imaging experience. Light fields provide a rich viewing experience with features like refocusig and changing the viewpoint. Recent improvements made the availability of light field in consumer based cameras but such cameras often pose a problem with resolution. With the usage of compressive light field imaging, studies have been made to generate a high resolution light fields but compressive sensing requires some heavy hardware modifications. To ease the process we suggest a deep learning model using coded aperture to generate light fields. We adapted the same network used in compressive light field sensing and tried to obtain an optimal code design.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Light fields provide a rich representation of real-world scenes, enabling exciting applications such as refocusing and viewpoint change.Generally, they are obtained by capturing a set of 2D images from different views or using a micro lens array. The early light field cameras required custom-made camera setups which were bulky and expensive, and thus, not available to the general public. Recently, there has been renewed interest in light field imaging with the introduction of commercial light field cameras such as Lytro. However, because of the limited resolution of the sensors, there is an inherent trade-off between angular and spatial resolution, which means the light field cameras sample sparsely in either the angular or spatial domain.

To reduce the curse of dimensionality when sampling light fields, we turn to compressive sensing (CS). CS states that it is possible to reconstruct a signal perfectly from small number of linear measurements, provided the number of measurements is sufficiently large, and the signal is sparse in a transform domain. Thus CS provides a principled way to reduce the amount of data that is sensed and transmitted through a communication channel. Moreover, the number of sensor elements also reduces significantly, paving a way for cheaper imaging.Marwah et al. (4) proposed a dictionary-based learning for local light field atoms (or patches) coupled with sparsity-constrained optimization. Gupta et al.(2017)(5) proposed a end to end light field reconstruction method using traditional auto encoders and 4D CNNs. Kalantari et al.(2016)(2) used learning based methods for view synthesis. Anil Kumar, Kaushik Mitra and others (1) proposed a learning based solution for full sensor resolution light field reconstruction from a single coded image. But, introducing a sensor between the sensor and lens involves some complex hardware changes. We want to go after a simpler method of using a coded aperture and propose a optimal design for code to obtain best results.

CHAPTER 2

Background

2.1 Light Field

Plenoptic function describes the possible radiant energy that can be perceived from the point of view of source. In case of a dynamic scene where we are also characterizing the wavelength it's a 7 dimensional function written as :

$$p = P(\theta, \phi, \lambda, x, y, z, t)$$

In case of static scene where we do not care about capturing wavelength specific information as well it becomes a 5d function as represented

$$p = P(\theta, \phi, x, y, z)$$

The radiance along the direction of ray remains constant for all practical purposed so we can discard the redundant information coming from one dimension which is z and this results into a 4d plenoptic function. This information in the 4D constitutes light field.

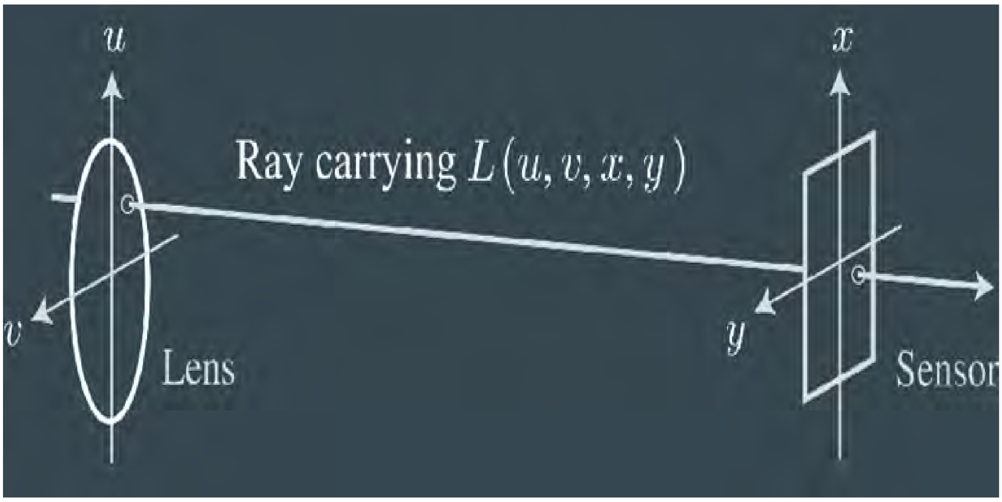

Figure 2.1: Light field

2.2 Coded Aperture

Coded Apertures or Coded-Aperture Masks are grids, gratings, or other patterns of materials opaque to various wavelengths of electromagnetic radiation.By blocking radiation in a known pattern, a coded "shadow" is cast upon a plane. The properties of the original radiation sources can then be mathematically reconstructed from this shadow. In a coded aperture more complicated than a pinhole camera, images from multiple apertures will overlap at the detector array. Later the original image is reconstructed using separate algorithms or through more modern procedures involving deep learning. A conventional camera captures blurred versions of scene information away from the plane of focus. Using a coded aperture is an example of computational photography where an optical element alters the incident light array so that the image captured by the sensor is not the final desired image but is coded to facilitate the extraction of information. Levin et al. (6) used coded apertures for extracting depth information from the coded image as well as reconstructing the original image.

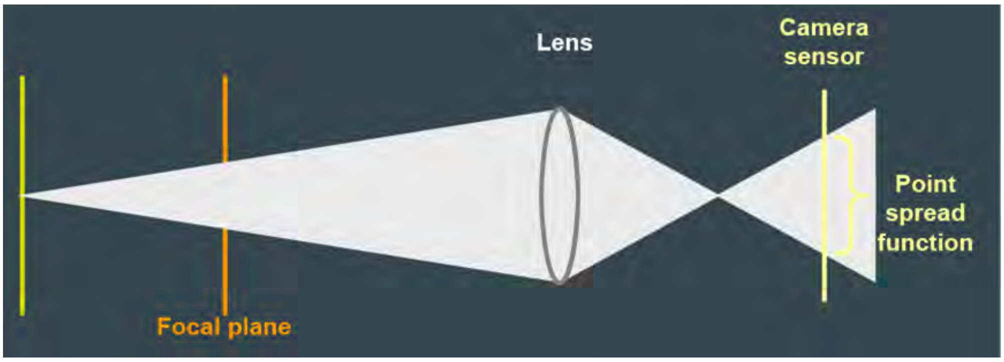

Figure 2.2: Normal Camera Functioning

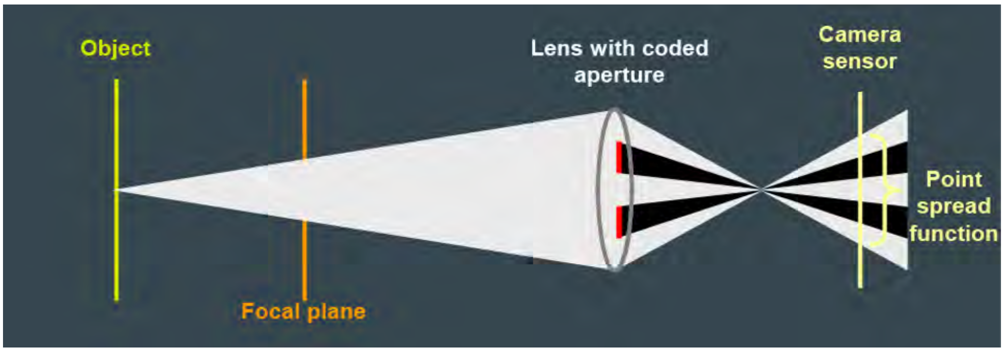

Figure 2.3: Camera with Coded Aperture Functioning

When an object is placed at the focus distance, all the rays from a point in the scene will converge to a single sensor point and the output image will appear sharp. Rays from an object away from the focus distance, land on multiple sensor points resulting in a blurred image. The pattern of this blur is given by the aperture cross section of the lens and is often called a circle of confusion. The amount of de-focus, characterized by the blur radius, depends on the distance of the object from the focus plane.For a simple planar object at a distance , the imaging process can be modeled as a convolution: $y = f_k * x(1)$ where y is the observed image, x is the true sharp image and the blur filter f_k is a scaled version of the aperture shape.

We have tried to use coded aperture for reconstruction of light field. This is different from the other methods as this method is hardware friendly as this requires minimum hardware changes when compared to the methods using compressive light field sensing. We have done several experiments using deep learning so as to propose the best optimal design for the code to be used for creating the coded aperture.

CHAPTER 3

Prior Work

In compressive sensing, a coded image is generated by placing a code nearer to the sensor. This coded image was later used to generate light field. Marwah et al. (4) has done this with the help of using dictionary based learning. Anilkumar et al.(1) has designed a deep learning based model and successfully generated good results of light field. We aim to generate an optimal design for code by supplementing our code design network with their disparity network. The network was taken from the work of Anil Kumar(1) on compressive light field imaging. We were As shown in the architecture, the network takes in a coded image which then proceeds to reconstruct the original view of the coded image. It uses deep neural networks for the sake of center view reconstruction. The generated view along with the coded image is then sent into a depth network which then gives out the disparity map of the image. The disparity map along with the generated view is then used to generate the required light field.

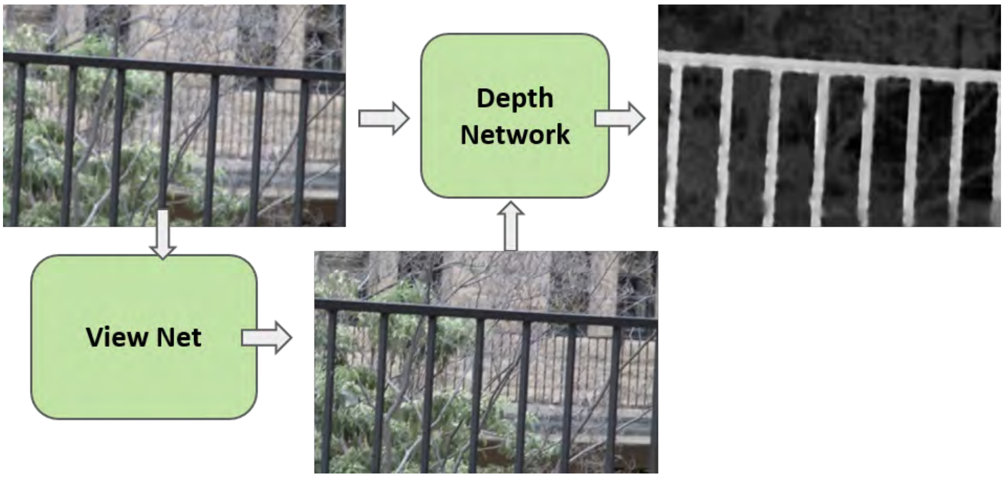

Figure 3.1: Network Architecture

3.1 Coded Image

The coded image is generally generated by placing a code at the aperture lens of the camera. The light rays entering the camera are then masked by the code placed at the aperture which results in the generation of the coded images. For the sake of our experiments, we took light field data set acquired from lytro illum cameras and then over leaved the images with the code present with us. We have done several experiments as to find the desired code.

3.2 Center View Generation

Image restoration from noised images have been a major interest of research for a long time. With the advent of deep neural networks, this area of interest has seen some drastic improvements. But as the extent of depth in a neural networks increased, it is observed that there is a decrease in performance after crossing a certain depth. This is because the gradient becomes too small after a certain depth and it is almost ignored when it reaches the top layers. To solve this problem the concept of residual networks is introduced. Taking the residual networks a bit ahead, with the introduction of skip connections Mao et al.(2017)(3) proposed a deep neural network model called Residual encoder decoder network called REDnet architecture.

The framework is fully convolutional and deconvolutional. Rectification layers are added after each convolution and deconvolution. The convolutional layers act as feature extractor, which preserve the primary components of objects in the image and meanwhile eliminating the corruptions. The deconvolutional layers are then combined to recover the details of image contents. The output of the deconvolution layers is the clean image that is required.Skip connections are also added from a convolutional layer to its corresponding mirrored deconvolutional layer. The passed convolutional feature maps are summed to the deconvolutional feature maps element-wise,and passed to the next layer after rectification.

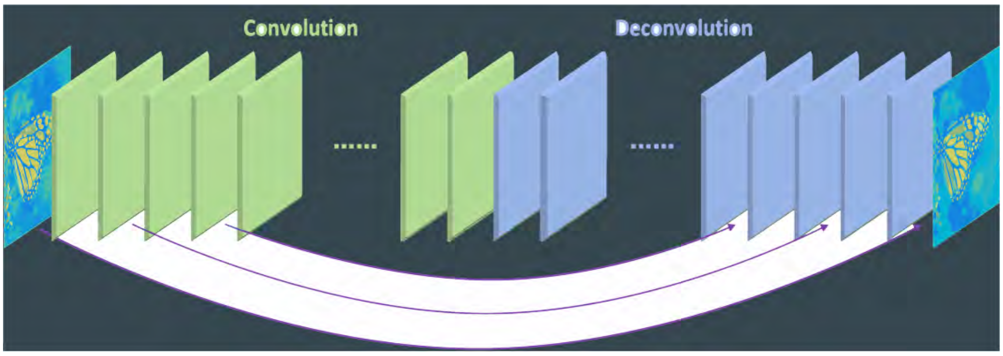

Figure 3.2: RedNet Architecture

We have trained this network with sample coded images to check on the reconstruction quality of the image and the results are pretty good. As we go further and learn the desired code, these reconstruction results are further improved.

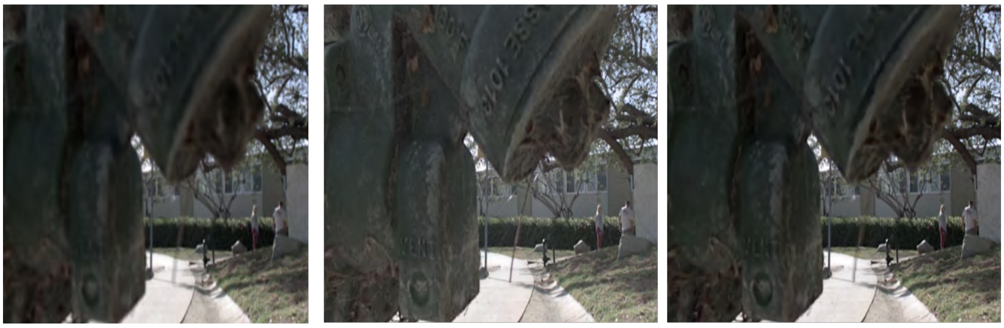

Figure 3.3: RedNet Results [left] Coded Image [Center] Ground truth image[Right] Reconstructed image

3.3 Depth Network

The reconstructed network is then passed into a depth network for the estimation of disparity maps. This network takes in the coded image and reconstructed view as input and generates the disparity map. The network is taken and improvised from the work on Focus-Defocus light field reconstruction done by Anil Kumar V.(1) It uses 2D convolution networks of series of encoder and decoder networks. The network is as shown below. Encoder performs strided(> 1) convolutions encoding the disparity information. Also, this provides a way for increasing the receptive fields. Decoder performs strided deconvolutions bringing back the feature map to the input resolution. The feature maps from decoder and skip connection are concatenated, followed by four convolutional layers to output final disparity map which is used to generate the required light field.

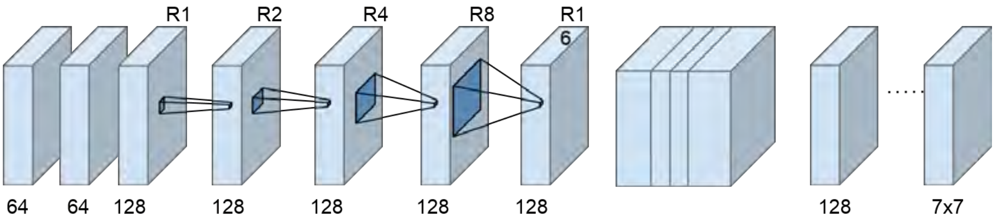

Figure 3.4: Disparity Network

CHAPTER 4

Code Design

My major contribution is to design the optimal code to be used at the aperture for efficient reconstruction of light fields. We have done a series of experiments, first by using a random code. And later we used a deep learning based method to train the code.

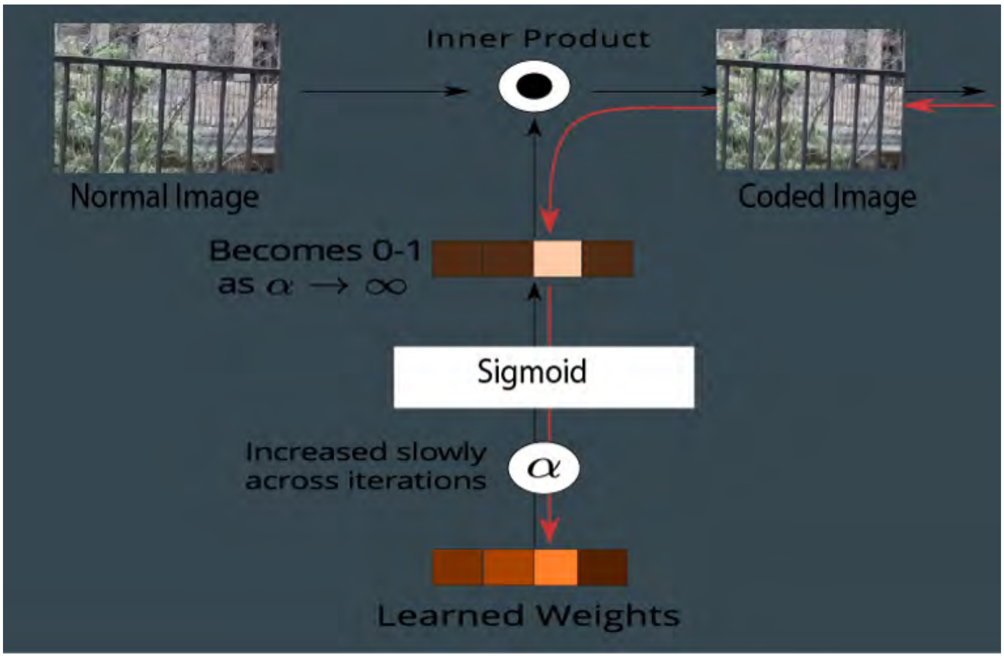
Inner Product
Normal Image
Coded Image
Becomes 0-1 as $\alpha \rightarrow \infty$
Sigmoid
Increased slowly across iterations
α
Learned Weights

Figure 4.1: Code Design

We initialized the code to be a tensor variable and let it trained. But under such conditions, the training was too slow and no significant changes were seen in the the code. So, we have added a temperature parameter α which is multiplied with the code and sigmoid is applied so as to constrain the values between 0 and 1. The value of α is increased along the iterations. So, as the iterations are increased , the code is more likely to attain a 0-1 pattern.

We have two losses , the loss arising from the reconstruction of center view and the loss arising from depth network. This loss gradient, as can be seen from the figure is propagated back to the code training weights which induces the code to change accordingly over the training period. We used Adam Optimizer as the gradient descent optimizer and a training rate of 0.00001. L1 loss is used for the calculation of loss in both reconstruction loss and depth loss. Here we present the experiments and the subsequent observations as we finally obtain the optimal code.

4.1 With Ground Truth

The first set of experiments involved the usage of ground truth image for Disparity network training. Ground truth images are used in place of center view so as to help in the identification of the rate at which α is to be increased. Also, this gives us in understanding the training pattern during the code design. For all the experiments, results were shown with the help of these two cases. The PSNR reported is an average value of 24 test cases obtained from test data set of Kalantari et al.(2)

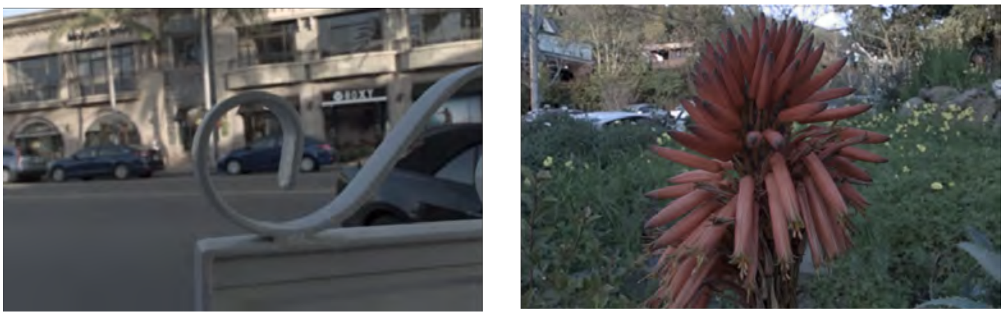

Figure 4.2: Ground truth images

4.1.1 Random

A random code is generated and is used to interleave the light field and generate a coded image like that in the case of an original camera when a code is placed at the aperture. We observed very good results in the estimation of disparity maps.

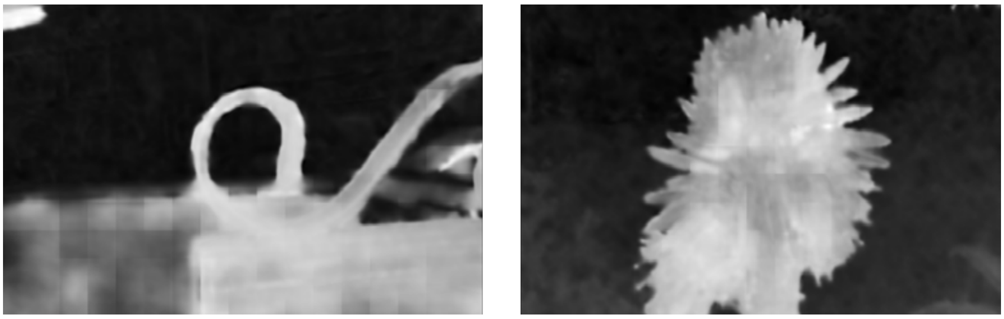

Figure 4.3: Disparity Map, Random Code, Groundtruth

4.1.2 Scheduling of α

Different schedules were followed to increase the value of α. Firstly α was increased according to a quadratic equation, $\alpha_t = 1 + (\gamma t)^2$ where t is the iteration count and the value of gamma is $2.5x10^-5$ This approach is inspired from the works of Ayan Chakrabarti et al.(2017) .This approach hasn't given desired performance, although the code is trained, the disparity estimation SNR value and the training loss both were pretty bad. So, we improved the training by resorting to a linear schedule for increasing the value of α for which we have achieved considerable good results. Following this schedule the value of α is increased by a factor of 0.2 for every 1000 iterations. The resultants codes that were trained are as shown.

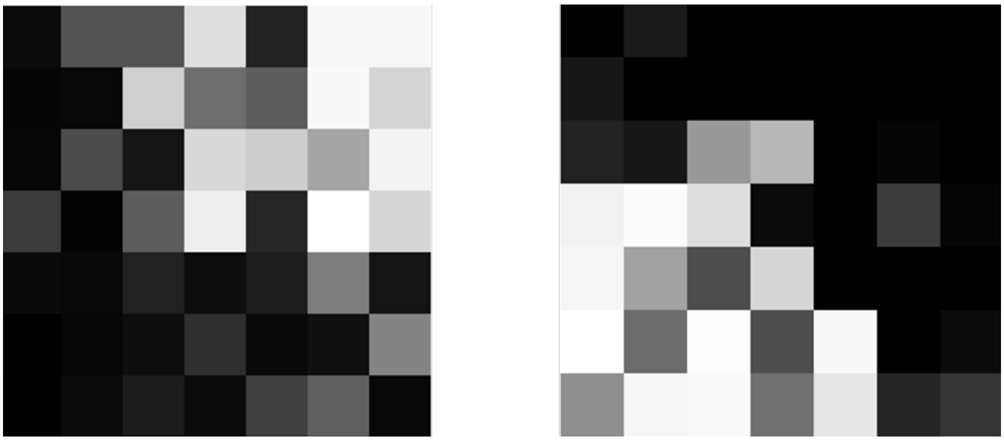

Figure 4.4: Learned codes [left] Quadratic scheduling [right] Linear scheduling

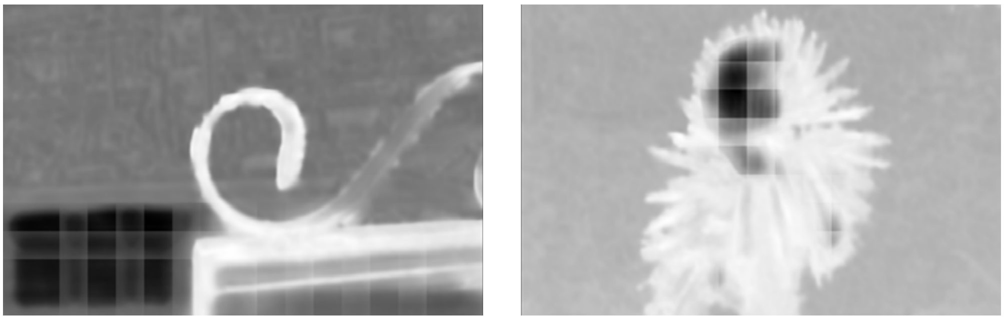

Figure 4.5: Disparity Map, Learned Code, Groundtruth

4.1.3 Results from Learned code, Linear Scheduling

4.1.4 Final observations

we can observe from the disparity maps of learned code, better texture recognition is being done by the leaned code, which even is resulting for some artifacts. We have seen that random codes model performed better than the learned one in some cases too. Also from both the codes obtained from quadratic scheduling and linear scheduling it is observed that the codes tend to form a gradient from white to black across the diagonal. This pattern is important as it helps the depth network for better depth estimation. Let's see how this changes when reconstructed images are used instead of ground truth images to learn the code.

4.2 With Reconstructed view

We used the same random code, but this time the ground truth image sent to the depth network is changed to the image that we generated from the Rednet architecture The obtained results are as follows. A significant deterioration in the quality of disparity maps is seen. The deterioration is probably due to the poor quality in the regenerated image when compared to the ground truth due to which information about the edges is lost.

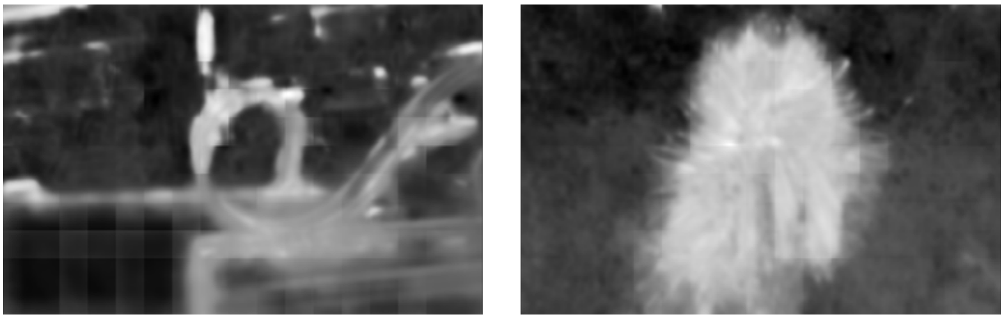

Figure 4.6: Disparity Map, Random Code, Generated view

4.3 Code Learning Using Deep Learning

We tried to learn the code using deep learning from jointly training with both the RED-net architecture and Depth network architecture. In this way both the network architectures gets trained to provide the best results possible. Initially the training produced results with training loss almost similiar to the loss as in case of Random code. But under careful observation it is discovered that the quality of images that were regenerated from Rednet architecture in case of the latter case were far better to those that were generated using the model from the Random code Rednet model. This is because of the change in code. The code generated is shown below(The one on the right).

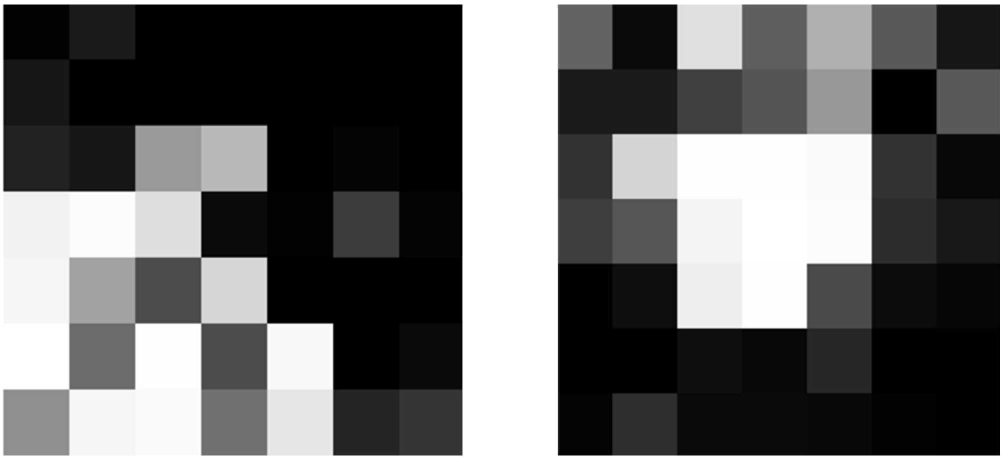

Figure 4.7: Learned codes [left] With ground truth [right] With rednet generated view

As can be seen the code is more white in the model when compared to at the corners. This particular type of code is highly preferred in case of image regeneration. So, regenerated images are of high quality. Although the regenerated images are of high quality, the depth network doesn't prefer this type of code. It prefers the diagonal gradient pattern. It even tried to bring in the diagonal gradient in the latter code too, but the white patch in the center is against its pattern. As a result, the results obtained were no better than the results from random code. So, to improve the model accuracy, we stopped the code and REDnet architecture from training and trained just the depth network with the obtained code from last experiment. This showed an improvement of 1db SNR value when compared to the random codes value. The results are as shown.

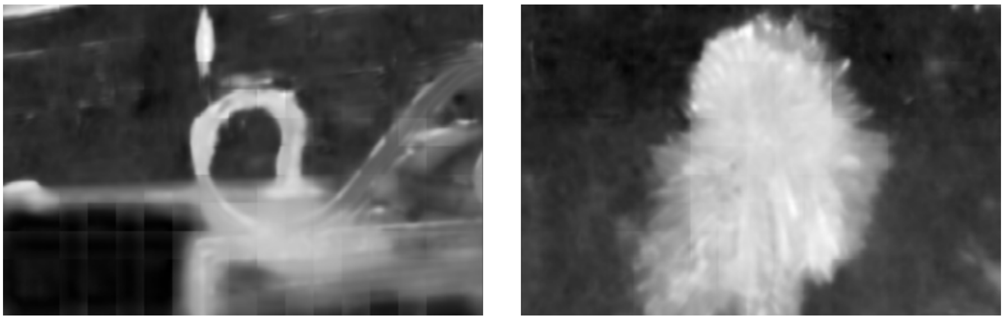

Figure 4.8: Disparity Map, Learned Code, Generated view

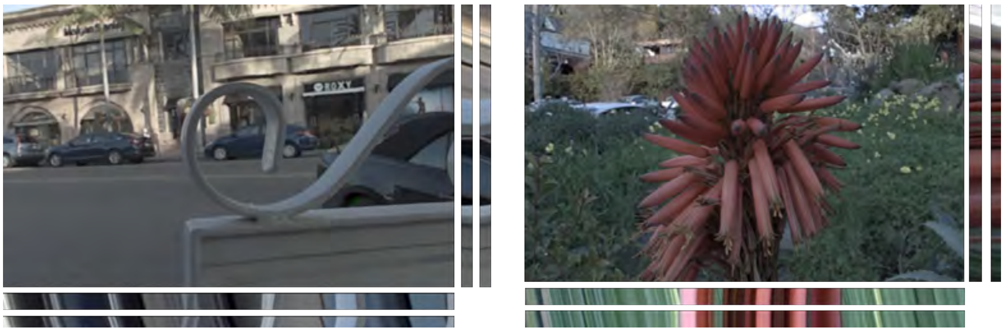

Figure 4.9: eps of final Learned light field

4.3.1 Final observations

Experiment	Average PSNR
Random code, Ground truth	34.959
Learned code, Ground truth	35.463
Random code, Generated view	31.31
Learned code, Generated view	32.281

Table 4.1: Results from various experiments

Although our best result of 32.281 is less when compared to the 35.707 psnr of heterodyne model, we have an advantage of easy hardware modifications. Also, we believe that with a suitablee tweaking in the architecture more efficient code which is not dominated by one of the networks can be achieved.

CHAPTER 5

Conclusion

An optimal design has been proposed to attain better performance when compared to using a random code. Although the method is not accurate as compressive light field sensing, ours is easier to implement and our reconstruction quality of image is better because of our images coed structure. It is observed that during training,the training is dominated by the enhancement of REDnet. With further improvements to the depth network , we believe our model can perform better. Also, we have been using the disparity network taken from the model of focus de focus pair. A more customized disparity network will lead to further improvements.

REFERENCES

[1] Anil Kumar Vadathya, Saikiran Cholleti, Kaushik Mitra *Learning Light Field Reconstruction from a Single Coded Image* ACPR 2017

[2] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi. *Learning-based view synthesis for light field cameras*. ACM Transactions on Graphics (TOG), 35(6):193, 2016.

[3] X.-J. Mao, C. Shen, and Y.-B. Yang. *Image denoising using very deep fully convolutional encoder-decoder networks with symmetric skip connections.* arXiv preprint, 2016.

[4] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar. *Compressive light field photography using overcomplete dictionaries and optimized projections.* ACM Transactions on Graphics (TOG), 32(4):46, 2013.

[5] ,Arjun Jauhari, Kuldeep Kulkarni Mayank Gupta,Arjun Jauhari, Kuldeep Kulkarni *Compressive Light Field Reconstructions using Deep Learning*

[6] Anat Levin ,Rob Fergus ,FrÂt'edo Durand ,William T. Freeman *Image and Depth from a Conventional Camera with a Coded Aperture*