Learning Depth in Light Field Images

Douglas V. Johnston

1 Introduction

Photographic images reduce the three dimensional world they are capturing into a 2D plane. There are a variety of applications in which it would be useful to determine the original distance to objects in the scene from the focal plane of the image. Such applications include robotic vision systems, 3D scene reconstruction, and surveying. Recent efforts to provide automatic analysis of scene depth using a single image have proved quite successful[4], however, ambiguities still arise in complex scenes, and non-general assumptions of the environment are made in ensure the best response of the algorithms. In contrast to traditional cameras, recent work has made use of a light field camera, which captures the 4D light field of the scene. The details of how this is done are described below. By using the extra information in the light field, we hope to provide a more general implementation of a depth map learning algorithm, which requires fewer parameters to train, and which will work in a variety of environments.

2 Light Field Acquisition

A hand-held, plenoptic camera[3] is capable of capturing information such that the raypath of light hitting a pixel can be determined. In practical terms, this has the advantage, among others, of enabling a single photographic image to be refocused at varying focal planes across the scene. The ability to refocus the elements of the scene comes at the price of reducing the resolution of the 2D image. For every additional separate focal plane captured, the resolution of the image is halved. The details of the implementation of a plenoptic camera can be found in [3].

This paper will structure the data is a slightly different manner than that of a plenoptic camera, but one that can be mathematically manipulated in a similar fashion. Instead of one high resolution camera, with a sensor of m by n pixels, we have a number of low resolution pinhole cameras, arranged in a plannar grid. The dimensions of this grid are labeled as u and v. Each camera has a resolution of s by t pixels. See figure 1. It is easy to see that the overall number of sensor elements is $u \times v \times s \times t$, which we will set equal to the $m \times n$ resolution of the single high resolution camera. Our implementation assumes an original resolution of 16 megapixels, or 4096×4096, with u and v both equal to 16, giving an array of 256 cameras. Dividing the original resolution by 256, we end up with individual camera resolutions of 256×256 pixels, which is the resolution used in this project.

In order to make use of the data, the images are aligned and placed into a focal stack. The focal stack is created by giving different offsets to each image in the $u \times v$ plane, and hence creating focus at different depths using pictures taken at the same time. The rest of this paper will use the data from a camera array as opposed to a microlens camera, due to the abundance of camera array training data. However, the theory presented here can easily be transformed to operate on a microlens camera.

2.1 Image processing

The synthetic aperture data is processed in Matlab to give independent focal plane data, by recombining different offset pixels from each image.

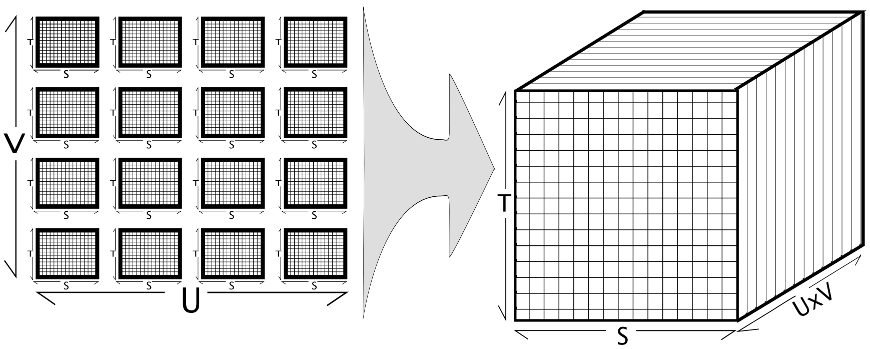

Figure 1: To simulate a plenoptic camera, an array of camera is used to capture many low resolution images of the same scene simultaneously. The images are aligned and, using a synthetic aperture, a focal stack is created, with each slice having sharp focus at a different depth in the scene.

Several results are shown in the figure 2.

3 Determining Depth

To determine a depth map for the entire image, we break the image down into component subimages, and assign a depth value to each subimage. Because of the powerful depth estimation available to us by using light field images, we place a high importance on the information we can extract from the focal stack. The reasons for doing so are twofold. First, to improve run-time performance, we attempt to limit the number of features used to be as small as possible[2]. Secondly, the ability to use the focal stack is what sets this technique apart from others, so we attempt to use the data contained within to the greatest extent. The larger synthetic aperture used, the greater the circle of confusion will be, and hence the more blurred objects will appear if they are not at the focal plane of the individual focal stack image being evaluated. To account for image registration error, we apply a two pixel wide Gaussian blur function to the resulting edge image.

Features must be identified which can give information on the depth of a subsection of the image. We use the notion of both relative and absolute depth to help build our features. For absolute depth, we look for texture using a set of convolution filters. We independently calculate the response of each of 3, 3×3 spot detectors. For relative depth, we compare the gradient for each image patch and its surrounding 6 neighbors (left, right, up, down, front, and behind). We define our depth measurement potential as

$$\Psi = \sum_{i=1}^{M} \sum_{j=1}^{F} \frac{|d_{ij} - x_{ij}^T \theta|}{\lambda_1}$$

where i is the set of image patches, j is the set of focal stack images, d is the depth, x is the absolute feature vector, and θ and λ are parameters of the model. Secondly, we create a depth smoothness prior[1] using our described method for calculating relative depth.

$$\Phi = \sum_{i=1}^{M} \sum_{j=1}^{F} \sum_{k \in N(i)} \frac{(x_{ij} - x_{kj})^2}{\lambda_2}$$

where the variables are the same as above, with the additon of $N(i)$ which is the set of six neighbors. Final, we use the general Markov Random Field equation

$$P(X = x) = \frac{1}{Z} \exp(-\frac{1}{T} U(x))$$

using the combination of Ψ and Φ as our energy function, and Z, our normalization constant. Putting the three together we have

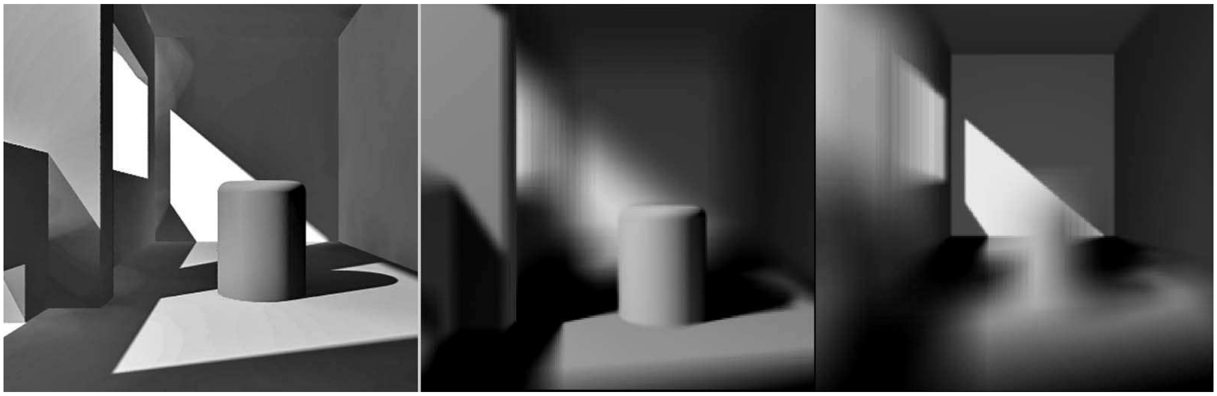

Figure 2: Single camera image and two focal stack images. The focal stack contains a discrete set of focal depths throughout the scene. Sixteen images comprise the focal stack during this experiment. Shown here are two such images. Left: image from one camera. Center: only the foreground is in focus. Right: only the background in focus.

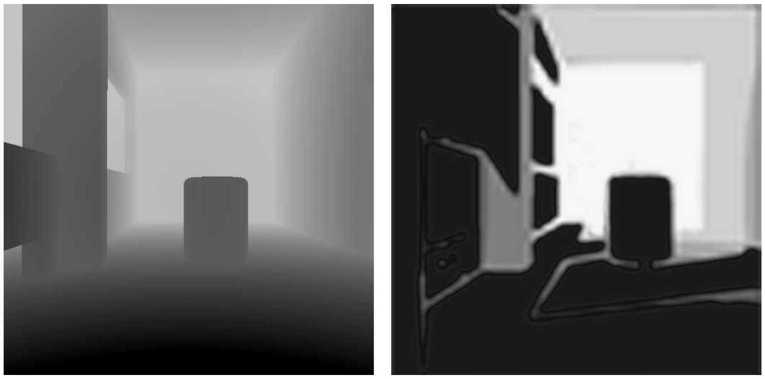

Figure 3: Left: Actual depthmap. Right: Computed depthmap. Areas without texture are not computed well.

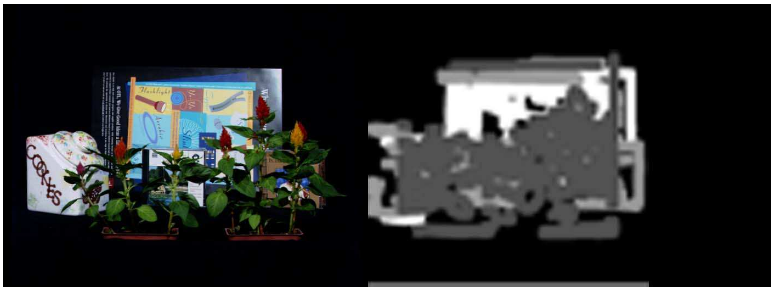

Figure 4: Depthmap created from real data using Stanford Camera Array. Left: original image from one camera. Right: computed depthmap. Significant improvement over images with no texture is made in images such as this with lots of texture variation.

$$P(d|X;\theta,\lambda) = \frac{1}{Z}\exp(-\frac{1}{2}(\Psi + \Phi))$$

which we use to learn the depthmap.

4 Results

The model was trained on both real and synthetic data, from a variety of scene locations. Synthetic data was modeled in POV-Ray, while real data was acquired from the Stanford Camera Array. In all, over 500 images were used to train. Because one of the goals of the project is to make as generic a learning model as possible, the test data varied significantly from the training data. The model responded well to the new images, as the learning mainly relies on edges. This makes depth recognition in previously untrained locations quite possible.

5 Limitations of the Model

There are some trouble areas, and room for improvement in the model. Because of the reduced spatial resolution of the images, fine textures are not able to be resolved. Many of the man-made objects present in the scenes, such as wood tables, white walls, and granite floors exhibit texture only at small spatial scales, which is lost in our images. In image segments without significant features, there is little difference in the segment in different focal stack images. Therefore, our heavy reliance on edge detection in the segment breaks down, and learning in the segment is poor.

6 Acknowledgments

We thank Vaibhav Vaish and Mark Levoy for the 2D camera array image data used for training.

References

[1] James Diebel and Sebastian Thrun. An application of markov random fields to range sensing. *NIPS 18*, 2005.

[2] Jeff Michels, Ashutosh Saxena, and Andrew Y. Ng. High speed obstacle avoidance using monocular vision and reinforcement learning. *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML 2005, 2005.

[3] Ren Ng. Light field photography with a hand-held plenoptic camera. *Stanford Tech Report*, CTSR 2005-02, 2005.

[4] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. Learning depth from single monocular images. *NIPS 18*, 2005.