Scene Representations from Focal Stack for Depth Estimation

Logan Bruns
Stanford University
lbruns@stanford.edu

Abstract

This paper explores the use of focal stacks as deep learning scene representation embeddings to improve depth estimation with or without the focal stack at inference time. The general approach taken was to train a deep learning model to first create a scene embedding for each image and its focal length in the focal stack. These scene embeddings are then used to create a single combined scene embedding which is then sent through a decoder network to predict the depth map. The intuition is that this allows the network to learn the depth cues from varying blur in each image along with its focal length. More training would be desirable as the test performance is still improving but a few promising conclusions seem evident. The deep learning model is able to learn from the blur depth cues in the focal stack. Beyond learning how to render the depth map from the focal stack it also appears to have learned a prior on the common features of the depth maps such as the walls. This prior can improve on the sensory gathered depth maps much as a human's grounding would be able to guess what would be in shadow based on what can be seen.

1. Introduction

Humans are fairly good at estimating depth and relative depth of a scene using a number of different cues as well as prior knowledge about the physical world.

Understanding the depth of a visual scene has a variety of useful applications including augmented reality, visual editing, data capture, and robotics. With the ubiquity of mobile devices which are capable of creating focal stacks a system to create 3d depth estimations using mobile phone cameras would allow for all kinds of applications with existing hardware.

Focal stacks already provide some depth information by way of knowledge of the camera optics and examining which part of the scene is sharpest at which focal setting. However, this often requires a deep stack and significant computation time to create a high quality depth map. These approaches often also have problems with surfaces that do not have very high contrast detail.

The notion here is to see if it is possible to use deep learning to introduce some prior knowledge to the depth estimation process. Much as human recognizes objects like boxes or spheres and understands the relationships of their surfaces it may be possible to train a model to do the same and there by improve depth estimation. Improve it either in terms of quality or runtime for common cases.

2. Related Work

Suwajanakorn1 et al [6] did some impressive work in this area in their paper "Depth from Focus with Your Mobile Phone". Their work which did not use deep learning focused on using many images and optimization to push the limits of the optical information encoded in the focal stack. It looks like the code is available. Depending on level of bit rot and complexity to set up it may make for a good baseline.

Eslami et al [3] explored using neural scene representations and deep learning to learn a representation of the scene that encoded some prior knowledge of the spatial relationships. Their approach of training the model to optimize a mental rotation task to force the model to learn to encode spatial relationships then using these representations for other supervised tasks depending on spatial relations is partial inspiration for this project.

Srinivasan et al [5] explored using deep learning to predict a depth map using aperature as the supervision. Namely given an all in focus image they trained models to predict both the shallow depth of field image and the depth map. The final loss being on the quality of the predicted shallow depth of field image. They used two main approaches to train it. One using a simulated lightfield and another using multiple blur kernels. Their focus was predicting the depth map based on a single all in focus image using aperature a supervision during training.

3. Method and Approach

The goal was to train a model to predict depth map from a focal stack. The approach uses some concepts that were

developed in Srinivasan et al [5] although not in quite the same fashion. A similar deep learning pipeline to their lightfield pipeline was constructed.

The input to the model is a variable length focal stack of images with corresponding focal lengths. The model is trained to produce a depth map from the focal stack. The bottleneck of the network is treated as a scene embedding. The intuition being that each of these focal image embeddings are adding to a better scene representation that is used to create the depth map. This in some sense is inspired by Eslami et al [3]'s work. Although the objective is different and the scene representation only speaks to depth.

Intuitively there are at least two ways for the model to learn from a focal stack. One way is learn from the depth cues from varying blur in each image paired with its corresponding focal length. Even without deep learning this can be done to some extent by detecting areas of highest contrast. Suwajanakorn1 et al [6] work in this area for example. The other is learn from the structure of objects in the image where it learns to recognize objects and use them in the depth map construction. For example, as seen in some pictures recognizing a chair creating it in the depth map sometimes with more of the chair shown in the depth map than actually shown in the photo.

The most suitable datasets for training and evaluation were depth datasets that have high quality images with corresponding depth maps. The datasets do not contain focal stacks so these had to be generated via an optical transform as described in a subsequent section.

Modeling and data generation was mostly done in Python with Tensorflow [1]. One small step was done in Matlab. The source code is available via github as listed in the supplemental section.

3.1. Datasets

To produce the focal stacks two datasets were considered. Both datasets have raw images as well as high quality depth maps. The idea is to use an optical transform to create a synthetic focal stack from each of these images to use for training, validation, and evaluation. The two datasets that were considered were NYU's "Indoor Segmentation and Support Inference from RGBD Images" [4] dataset and Matterport's "Matterport3D: Learning from RGB-D Data in Indoor Environments" [2]. NYU's dataset is available for immediate download. Matterport's dataset requires asking permission. It was decided that NYU's dataset was sufficient for this effort.

The process, described below, to generate the focal stacks proved to be computational expensive when done with enough layers to look visually correct. For this reason it was precomputed into tfrecords and more of the data was pulled into the training process as it became available. Because of this only a little under 200 scenes from the labeled 1449 dataset were used during training as of this time. Since the labels are not required it could potentially be possible to use not only the full labeled dataset but also the large unlabeled 407,024 image dataset.

3.2. Synthetic Focal Stacks

Since the dataset does not already contain a focal stack the approach was to use an optical transform to create a synthetic focal stack from each of these images to use for training, validation, and evaluation.

The general approach is similar to how it is done in Srinivasan et al [5] and is shown in Figure 1. Although to improve the quality of the refocused images 32 times (for a total of 256) more blur layers were used. As shown in the figure for each randomly chosen simulated focal length a blur matrix is created using a gaussian kernel size proportional to the distance from the focal plane. In addition a mask matrix is constructed where each mask plane in the matrix has 1 for pixels at that distance and zero otherwise. Elementwise multiplying these two matrices and reducing to the sum along the distance axis gives the resulting image focused at the chosen focal length. Note that originally this was implemented using Tensorflow's fft2d and ifft2d operators using a point spread function but there were some artifacts in the resulting blur matrix and this approach was shelved to focus on other aspects of the project.

An example of a synthetic focal stack is shown in Figure 2. For each original image and depth map a focal stack with five images corresponding to five randomly chosen focal lengths is constructed. With the original image this makes a focal stack containing six images.

3.3. Data Augmentation

In order to improve model training effectiveness and workaround the currently small dataset sizes data augmentation was used as well. A dataset transform was added to the dataset loading pipeline to randomly crop out a different part of each focal stack after every shuffle (epoch). The original image size was 640x480 and the random crop was down to 320x240 so each focal stack is turned into 76,800 variations.

3.4. Model Architecture

Figure 3 shows the model architecture. The overall goals of this design are:

1. Support varying length of focal stacks
2. Learn a compact embedding as scene representation
3. Stability during training

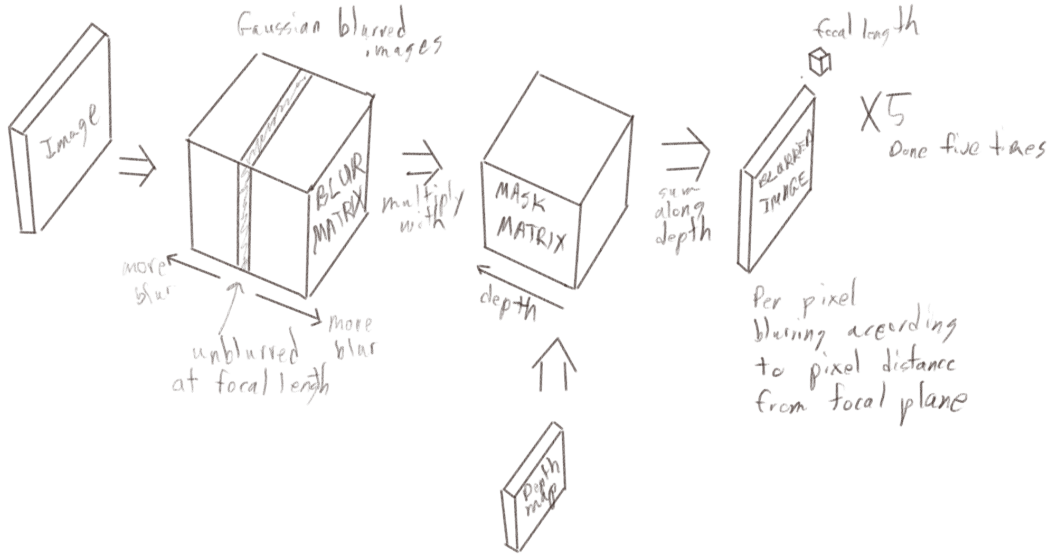

Figure 1. Creation of Synthetic Focal Stacks

Figure 2. Example of a Synthetic Focal Stack

As shown in the figure the network has three main components: encoder, combiner, and decoder. The choice of this structuring is to support goals 1 and 2.

The encoder network first stacks the focal length onto the input image as another channel then uses a series of convolution layers to reduce the representation of the combined image and its focal length. The convolution strides (listed in the figure) were chosen such that it is reversible as deconvolution in the decoder. At the end is layer normalization to support goal 3 and prevent drift.

The combiner network is an LSTM which takes a sequence of the scene representation embeddings and outputs a combined scene representation embedding. Other simpler mechanisms like max pooling were considered but since the LSTM was able to learn fairly rapidly the thinking was the additional parameters and capability would help with further learning about relations between information in different scene representation embeddings.

The decoder network is largely a reverse of the convolution layers in the encoder network. This is followed by a layer normalization and final convolution to output the depth map.

In general, all the activations are ReLU except in the LSTM where it is tanh to take advantage of CuDNN optimizations and final output is linear.

Most of the architecture exploration was in the CNN structure and combiner network choice. Done by observing shorter runs and the loss curve. If more time had been available it would also have been interesting to have tried using SSIM or PSNR in the model loss rather than just MSE.

4. Analysis and Results

Figures 4 and 5 show examples from the test and train datasets respectively. Note that each set of figures starts with the focal stack and then has the model predicted (left) and ground truth (right) depth maps in the last row.

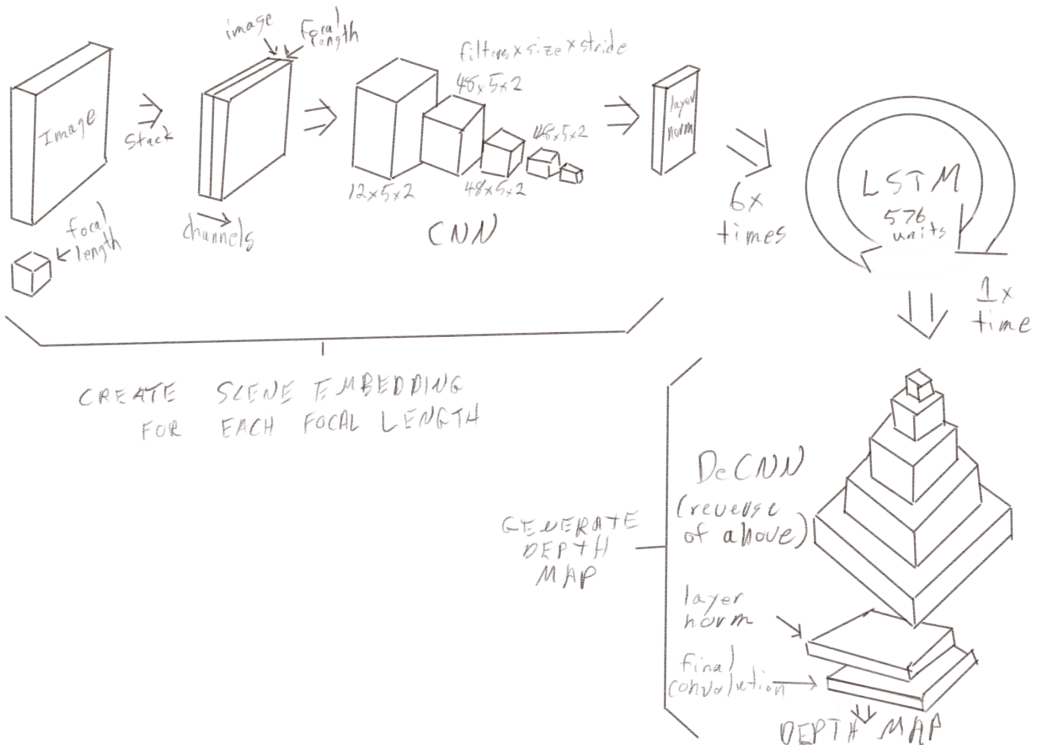

Figure 3. Model Architecture

4.1. Ability of model to learn without blur depth cues

As a baseline the single original image was used with the same model. Without random cropping for data augmentation it showed some minimal progress in memorizing the depth map. With random cropping data augmentation it appeared unable to learn anything all.

4.2. Ability of model to learn from blur depth cues

Adding the focal stack allowed the model to train and decrease loss. On the train dataset the loss goes down to around .04 MSE and fluctuates a little around there. You can see from Figure 5 that it able to overfit the training dataset.

The test loss continues to very slowly fall though. It remains relatively high but even so outperforms the quality of the "ground truth" depth map for certain cases. Both of these aspects can be seen in Figure 4. Notice that in the focal stack and model predicted depth map the shape of the room and especially wall edges on the left are captured correctly while in the sensor collected depth map this part is not captured correctly.

The NYU dataset consists on interior room pictures. It would make sense that the told more start to generalize best for features common to virtually all examples such as walls, corners, and such. Comparably chairs seem to be captured well in depth maps. The hope would be that as more data is added and with more training time that the detail and overall quality of test predictions would improve.

4.3. Ability of model to infer without blur depth cues

Interestingly in the case where the model is trained with the focal stack and then run without the focal stack it is still able to predict some aspects of the depth map. For example, examine Figure 6 where inference is performed on only the original image from the test dataset. The model prediction in the middle shows that it is still able to infer the basic depth geometry of room. Not very well but to some extent and this is without any depth cues from focal stack.

4.4. Quantitative Results

Mean Squared Error (MSE), Peak Signal to Noise (PSNR), and Structual Similarity (SSIM) for test and train datasets are shown in Table 1. Test performance, although not quantitatively very good, continues to improve so this table just a snapshot into in progress training not the final result.

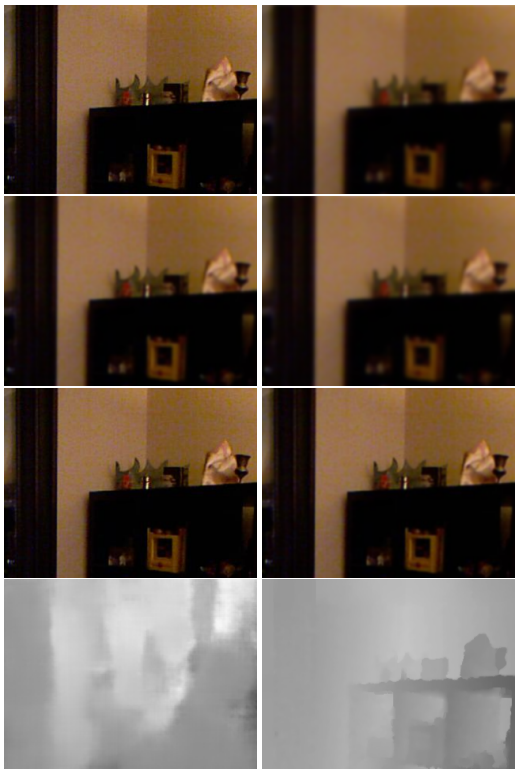

Figure 4. Test Example

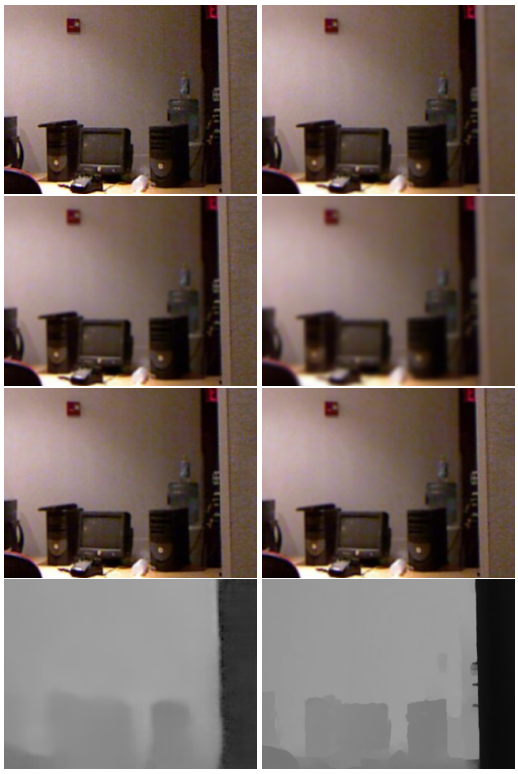

Figure 5. Train Example

Table 1. Quantitative Results

Dataset	MSE	SSIM	PSNR
Train	0.0434	0.9724	37.7189
Test	5.5372	0.8318	19.6759

4.5. Conclusions

In summary, the deep learning model is able to learn from the blur depth cues in the focal stack. Beyond learning how to render the depth map from the focal stack it also appears to have learned a prior on the common features of the depth maps such as the walls. This prior can improve on the sensory gathered depth maps much as a human's grounding would be able to guess what would be in shadow based on what can be seen. Even with random cropping data augmentation creating significantly different images the training examples fit very well. Test examples are improving and are able to predict gross shapes in many cases. The hope is that as more training examples are added it will continue to improve the generalization and test performance.

Figure 6. No Focal Stack Test Inference Example

4.6. Future Work

Since test loss continues to drop almost purely monotonically with continued training and accelerate some with

additional data injection one obvious follow up would be to train longer with more data. Data generation takes a long time and is still running so the amount of available data has barely been tapped so far. It would also be interesting to test it with a real focal stack taken by a camera and compare the results.

Additionally, one of the large benefits of forcing a network to learn a compact embedding or scene representation in this case is the opportunity for transfer learning. Both the NYU [4] and Matterport [2] datasets also contain pixel level instance and object labels. It would be interesting to see if the scene representations trained for depth estimation would benefit instance and object classification. At least at an intuitive level this would seem possible.

4.7. References

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[3] S. M. A. Eslami, D. J. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, D. P. Reichert, L. Buesing, T. Weber, O. Vinyals, D. Rosenbaum, N. Rabinowitz, H. King, C. Hillier, M. Botvinick, D. Wierstra, K. Kavukcuoglu, and D. Hassabis. Neural scene representation and rendering, 2018. DOI 10.1126/science.aar6170.

[4] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[5] P. P. Srinivasan, R. Garg, N. Wadhwa, R. Ng, and J. T. Barron. Aperture supervision for monocular depth estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[6] S. Suwajanakorn, C. Hernandez, and S. M. Seitz. Depth from focus with your mobile phone. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

5. Supplemental Material

`https://github.com/loganbruns/deepdepth`