DeepStereo
Dense Depth Estimation from Stereo Image Pairs using Convolutional Neural Networks

Matt Vitelli
Department of Computer Science
Stanford University
mvitelli@stanford.edu

Saumitro Dasgupta
Department of Computer Science
Stanford University
saumitro@stanford.edu

March 16, 2015

Abstract

We present a framework that generates a dense depth map given two rectified RGB images from a calibrated stereo rig. Unlike traditional methods for stereo reconstruction, our method is based on convolutional neural networks and does not directly incorporate any knowledge of multiple view geometry.

1 Introduction

Stereo reconstruction methods have been extensively explored since the early days of computer vision. The main challenge such methods seek to overcome is known as the correspondence problem: given two images of the same scene, match the pixels in the first image to the pixels of the second one. Usually these reconstruction methods make use of rectified images in order to reduce the dimensionality of this correspondence search from a two-dimensional search to a one-dimensional search along the scanlines of the images [1, 2].

While many algorithms have been created to solve the correspondence problem, they generally fall into two broad categories: local methods and global methods. Local methods tend to be used in real-time applications where computational efficiency is valued over accuracy. These methods often involve minimizing a photometric error term that is computed from a local neighborhood of pixels in each image. Many different scoring functions can be used including SAD scores, cross-correlation, L2-norm, etc. Such methods typically fail in scenes with little texture or complex lighting conditions and often result in noisy depth estimates. Global methods are used when depth estimation accuracy is valued more than computational efficiency and as such are usually limited to offline applications. These algorithms typically impose some sort of spatial smoothness constraint in addition to minimizing a photometric error term[3]. However, even these global optimization methods tend to produce sub-optimal results with texture-less regions, specular highlights, or complex occlusion boundaries.

The goal of this project is to utilize convolutional neural networks as a means of computing dense depth map estimates from a stereo camera system. We believe this leads to a nice compromise between the accuracy of the reconstructions and the computational efficiency of producing such depth estimates. Additionally, we believe that deep learning methods may help in providing robustness to challenging cases such as specular highlights, texture-less regions, and partial occlusions.

2 Related Work

Convolutional networks have recently been applied to the task of depth estimation. Eigen et al.[4] use a multi-scale convolutional network to perform monocular depth estimation, where the low-resolution scales are used to perform coarse depth estimates and high-resolution scales are trained to perform local depth refinements. Zbontar et al. [5] utilize convolutional networks as a means of determining corresponding patches in left and right stereo images, but rely on traditional stereo reconstruction methods such as semi-global block matching to estimate the depth at each pixel. Liu et al. [6] make use of convolutional networks to learn the binary and unary potentials for their conditional random field (CRF) framework for the task of single image depth estimation. Sun et al [7] extract sparse feature vectors from monocular infrared images and train a fully-connected neural network with a single hidden layer to produce dense depth estimates.

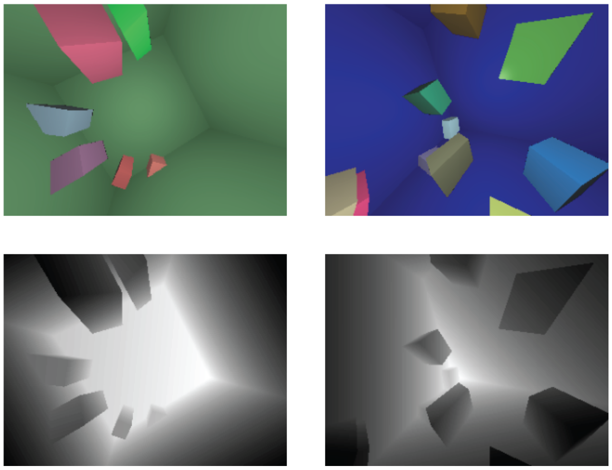

Figure 1: Sample scenes from our procedurally generated dataset.

3 Problem Statement

Formally, our problem is defined as follows: given two RGB images, I_L and I_R, taken from a a rectified stereo camera rig with known intrinsic and extrinsic camera parameters, we would like to estimate a new monochrome image $\hat{D}$ where each pixel value is the depth estimate for the corresponding pixel in I_L. The subscripts L and R represent the left and right images respectively.

4 Dataset

Conventional stereo reconstruction datasets (for instance, the Middlebury Stereo Vision dataset) usually contain very few training examples with ground truth depth data. However, convolutional neural networks require significantly larger amounts of training data, making most stereo datasets impractical for training. As a result, we opted to generate a synthetic stereo dataset for training our networks. Figure 1 shows a couple of sample scenes along with the corresponding depth maps for our procedurally generated dataset.

Our synthetic dataset consists of 60,000 images. This is split into 50,000 for training and 10,000 for validation. The left and right images, I_L and I_R, are rendered as 8-bit 320 × 240 RGB PNG images, while the depth map, D, is rendered as a single channel 8-bit PNG image. We decided to restrict ourselves to the Manhattan world assumption[8]. The scenes consist of a cubcic room with plane aligned cubic geometry. The shape and color of the room as well as its contents are randomized. The position of the objects within the room is randomized as well. Finally, the viewpoint is randomized as well, subject to the rules described below.

The following rules are used for randomization:

- The value/brightness component of the randomized colors are clamped, so as to prevent near-black colors. No restrictions are imposed on the hue component.
- The object sizes are clamped to prevent extremely skewed aspect ratios.
- The objects are plane-aligned (floor, walls, ceiling).
- The objects are allowed to have arbitrary in-plane rotations.

5 Network Architecture Experiments

We implemented and analyzed two convolutional neural network architectures for depth estimation. In this section, we describe and analyze both of these in detail.

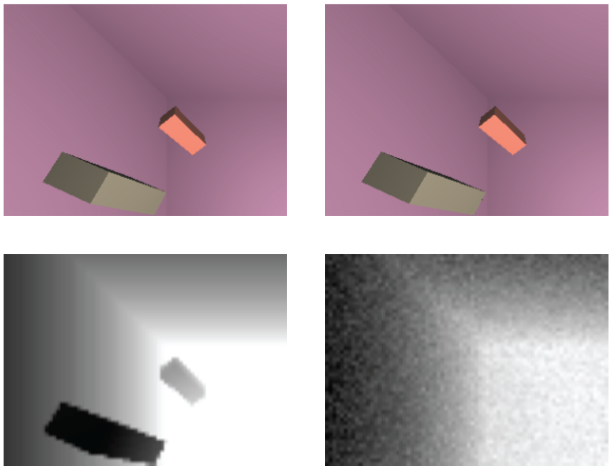

Figure 3: The depth-map estimate generated by the first architecture. On the left is the ground truth, while on the right is our network's estimate.

5.1 Architecture I

The first architecture (shown in 2) is largely based on the original AlexNet implementation [9]. The left and right 3 channel 304 × 228 RGB images (produced by taking the center crop of the original 320 × 240 images) are concatenated along the channel dimension to produce the 6 × 304 × 228 input. For regression, we use an Euclidean loss layer (the L2 norm of the difference between the estimated and the ground truth depth map):

$$\text{Loss} = \frac{1}{2N} \sum_{i=1}^{N} || \hat{D}^{(i)} - D^{(i)} ||_2^2$$

where N is the batch size. Conceptually, this architecture is similar to the first scale level of [4], only with a L2 loss function rather than a logarithmic loss function.

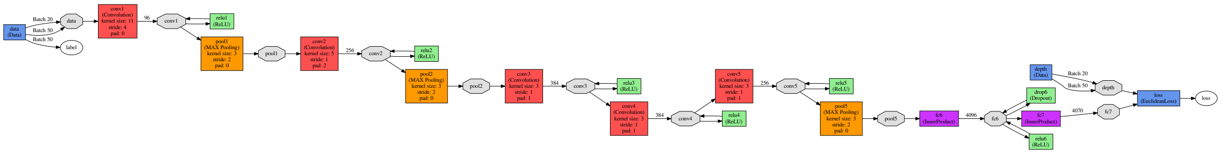

Figure 2: Network diagram for the first architecture.

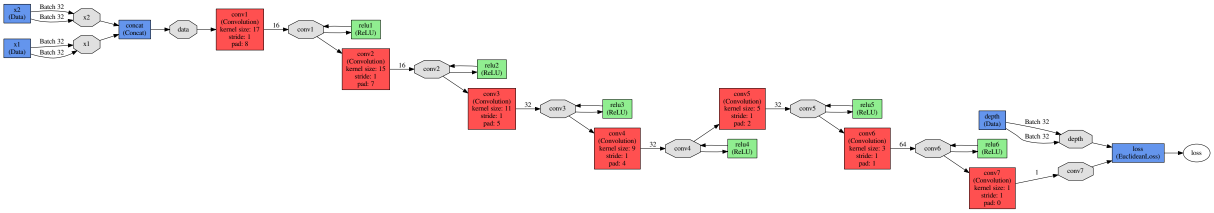

Figure 4: Network diagram for the second architecture.

5.2 Architecture II

Our second architecture is fully convolutional, as shown in figure 4. We discard the max-pooling and fully connected layers for a series of convolutional layers with decreasing kernel sizes interleaved with rectified linear units. A unit stride along with an appropriate amount of padding is used to ensure that the spatial dimensions remain the same at the output of each layer. Due to memory constraints, we are conservative with the number of outputs for each convolutional layer (as shown in figure 4).

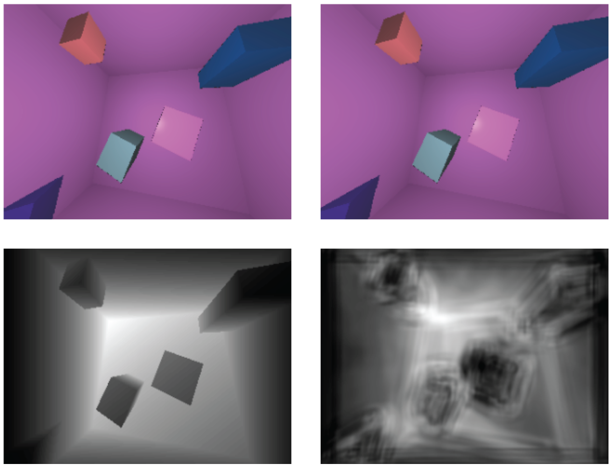

Figure 5: The depth-map estimate generated by the second (fully convolutional) architecture. On the left is the ground truth, while on the right is our network's estimate.

6 Implementation

6.1 Procedurally generated dataset

Our prodecural stereo scene generator was implemented using C++ and OpenGL. The depth maps were generated using a custom GLSL shader. The rendered frames were stored as

Architecture	RMSE
Modified AlexNet	0.098
Fully Convolutional	0.033

Figure 6: Root-mean-square error between the estimated and the ground truth depth-maps for the two architectures.

LMDB databases. Besides mean subtraction, the rendered images were not preprocessed in any other way.

6.2 Network

We implemented and trained our networks using Caffe [10], an open-source framework for convolutional neural networks developed by the Berkeley Vision and Learning Center. Each network was trained for two days on an Nvidia K40 GPU.

7 Results and Analysis

7.1 Overview

We found in practice that the fully convolutional architecture consistently outperformed the original AlexNet-inspired architecture described above. Table 7 compares the RMSE (root-mean-square error) for each network. We are able to achieve lower error rates with our fully convolutional network than our initial network architecture. In addition, we are also able to produce smooth depth estimates in textureless regions where traditional stereo reconstruction methods typically fail. Empirically, the depth maps produced by our improved network are reasonably close to the ground truth, but contain ringing-like artifacts around the silhouettes of small objects and black regions near the borders of our images. We believe the black borders are caused by the use of zero-padding along the input edges and are amplified by the large filter sizes in the initial convolution layers.

For training, we experimented with both Stochastic Gradient Descent with momentum, as well as AdaGrad. We found SGD with momentum outperformed AdaGrad, which tended to plateau early.

Below, we analyze each of the architectures in detail.

7.2 Architecture I

We found that this network tended to produce depth estimates that were accurate at a global level, but was unable to learn complex geometry. In addition to this, fine details tended to be washed out and small 3D shapes often would be missing entirely from the resulting depth maps. We believe the primary cause of this was due to the aggressive nature of the max-pooling layers.

7.3 Architecture II

The second architecture produced significantly improved depth estimates. We attribute this improvement to the following:

1. Stereo has a fixed range of accuracy governed by the maximum expected disparity.

   While mathematically the relationship between disparity and depth holds regardless of where each 3D point falls on each image plane, there are practical limitations that effectively bound the range of depth values that can be observed. For instance, very small disparity values will be limited by the resolution of each camera sensor and obtaining accurate estimates for points very far from the camera rig will not be possible.

Similarly, very large disparities may not be directly observable in both sensors, thus reducing the effective range of the stereo system. However, for most indoor scenes, neither of these extreme cases hold true and generally the distribution of disparities lies within a narrow window of values.

2. Max-pooling tends to wash out fine-grained details.

   The main advantage of pooling is to reduce the size of the input volume and thus has the added benefit of reducing the number of parameters for fully-connected layers in the network. For our particular application, reducing the size of the input was not desirable and empirically we observed that max-pooling tended to discard useful high-frequency depth information. Because of this, we believed it would be advantageous to remove pooling entirely and preserve the original input size of the data.

   The two observations listed above motivated us to create a new convolution network that relied solely on convolution layers. A diagram of our network is shown above. Essentially, our network makes use of large filters for the initial layers and slowly reduces the filter sizes, while increasing the depth of each successive layer. The motivation for creating this architecture was that we believe that larger filter sizes allow the network to search for large disparities, while smaller filter sizes allow the network to estimate small disparities. By slowly reducing the filter sizes at each successive layer, we are in a sense able to incorporate global depth information into our model without incurring a large increase in the number of parameters introduced by fully-connected layers.

Future Work

This project has just scratched the surface of what is possible. We believe that there are many other exciting directions to take our research in applying convolutional networks to traditional 3D computer vision problems. One possible extension could be to apply our method to stereo videos and incorporate temporal constraints into the depth estimation process. Another application could include regressing on other features of the scene, such as the scene's reflectance field and making use of stereo depth estimates as a prior to guide the reflectance field decomposition. Additionally, we could make use of our dense depth estimation as an initial estimate of the scene geometry for use in structure-from-motion algorithms.

It would also be interesting to analyze to what degree the network is learning "stereo features". Repeating our experiments with just the left channel should yield informative results.

Conclusion

We have demonstrated that convolutional networks can be used to map RGB pixels from stereo images directly to depth values. In contrast to traditional stereo-based reconstruction techniques, our method is capable of reconstructing texture-less regions in the scene and does not rely on explicitly searching for pixel correspondences in each image.

Acknowledgment

The authors would like to thank Peet's coffee.

References

[1] Andrea Fusiello, Emanuele Trucco, and Alessandro Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000.

[2] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[3] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[4] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, pages 2366–2374, 2014.

[5] Jure Žbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. *arXiv preprint arXiv:1409.4326*, 2014.

[6] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. *arXiv preprint arXiv:1411.6387*, 2014.

[7] Lin Xi, Shao-yuan Sun, Lin-na Li, and Fang-yu Zou. Depth estimation from monocular infrared images based on svm model [j]. *Laser & Infrared*, 11:025, 2012.

[8] James M Coughlan and Alan L Yuille. The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. In *NIPS*, pages 845–851, 2000.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.