Deep Learning for Depth Learning
CS 229 Course Project, 2013 Fall

Enhao Gong, Electrical Engineering, Stanford ENHAOG@STANFORD.EDU
Hang Qu, Electrical Engineering, Stanford QUHANG@STANFORD.EDU
Song Han, Electrical Engineering, Stanford SONGHAN@STANFORD.EDU

Abstract

Extracting 3D depth information from images is a classic problem of computer vision. Traditionally image depth could be extracted by techniques such as stereo camera or images from multiple views. In this project, we are trying to recognize the depth information by using a single still image from single camera, which has great potential applications in vision and recognition. To learn the complex relationship between single RGB image with its depth information, we chose to use Deep learning algorithms, to learn the multiple level features and corresponding different levels of abstraction. The main goal of the project is to train a deep network that is able to extract local and non-local features and predict a depth-map given a still image. In addition, implementation of other learning algorithm (kernel based, dictionary learning) was also conducted for comparison.

1. Introduction

1.1. Depth information

Extracting depth information from 2D images is a basic topic in computer vision. Traditional methods can work on binocular vision given by stereopsis cameras (Scharstein & Szeliski, 2002), or uses multiple images taken for the similar scenario. Stereopsis camera can take two images for the same scene and the relative camera attitude of the two images is fixed and known. This further simplifies the algorithm because it is possible to match the feature points between the two images. Multiple images with closed relationship can also help with extracting depth information. Such images might come from motion (Ponce et al., 2011) or from defocusing (Das & Ahuja, 1995). The tightly coupled images offer rich information to reconstruct depth information. However estimating depth from a single monocular image is still difficult, because only limited information is contained in a single image.

1.2. Potential Applications

There are a lot applications possible based on this projects. Our main task is similar to (Saxena et al., 2005). In addition, the depth information can also be widely used for category and instance level recognition to get much better accuracy. Possible application can be done using depth information with deep network. Besides, the 3d information of the depth is also necessary for multi-view synthesizing.

To tackle depth estimation in a single monocular image, existing successful methods fall in two groups. First, by putting constrain on the reconstruction environment, an algorithm can get prior knowledge about the scene, and thus achieves reasonable performance. Such prior knowledge includes the distance of an outstanding object (Michels et al., 2005), the shape and texture of outstanding objects in the environment (Nagai et al., 2002), background color and texture, etc (Gini & Marchi, 2002). The algorithms are mostly designed based on the given constrain, and construct depth information based on it. With prior knowledge of outstanding objects, an algorithm can identify the objects and simplify the depth estimation. With background color and texture, an algorithm can easily distinguish between objects and background and thus make better estimation. Second, some algorithms make no assumption about the environment, and thus work on unconstrained environment. This is a more realistic assumption but introduces more challenges. For example, (Saxena et al., 2005) uses Markov Random Fields to capture the global structure of the image that depth information is continuous, and gets good results without prior knowledge of the image itself.

1.3. Brief reviews of deep learning and related learning algorithms

In the class, we learnt about basic neural network algorithm, but in deep learning, one would usually build neu-

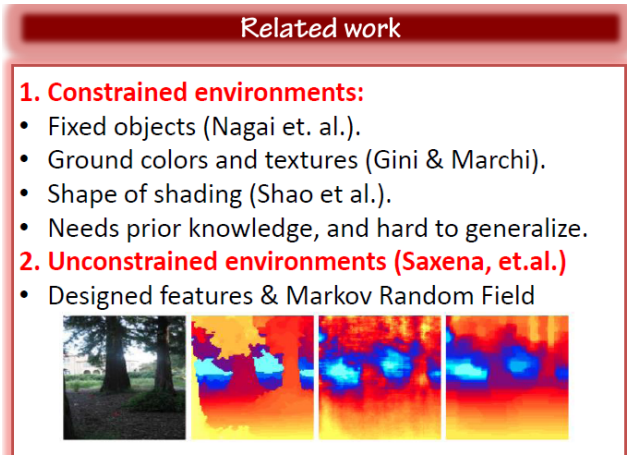

Figure 1. Review of previous algorithm.

ral networks with other architectures, including neural network with multiple hidden layers. Given increasing computation power, training such a complicated model is no longer unfeasible. Introducing a multi-layer neural network can certainly increase the expressiveness of the model, but might increase KL dimension of the hypothesis set and more likely to cause overfitting. So sparse autoencoder (Lee et al., 2007) is introduced to solve the problem. An autoencoder neural network is an unsupervised learning algorithm that applies back propagation, setting the target values to be equal to the inputs. It is used for learning efficient feature coding, that is, to learn an effective feature for a set of data. It is quite similar to PCA for dimensionality reduction. And the autoencoder neural network has to satisfy the requirement that the number of hidden units should be small. But even when the number of hidden units is large, we can still discover interesting structure, by imposing sparsity constraints on the hidden units of the network.

1.4. Feature extraction reviews

To solve this problem, feature extraction using convolution is a good method of achieving this. Natural images have the property of being stationary, meaning that the statistics of one part of the image are the same as any other part. This suggests that the features that we learn at one part of the image can also be applied to other parts of the image, and we can use the same features at all locations. More precisely, having learned features over small (say 8x8) patches sampled randomly from the larger image, we can then apply this learned small feature detector anywhere in the image. Specifically, we can take the learned 8x8 features and convolve them with the larger image, thus obtaining a different feature activation value at each location in the image.

2. Methods

2.1. Data and Pre-processing

In the project, we were using the Washington RGB-D Object dataset and the NYU depth dataset (Silberman & Fergus, 2011). For Washington dataset, they are mainly intended to be used for object recognition. The RGB-D kernel descriptors and Hierarchical Matching Pursuit were proposed by the group to achieve state-of-art feature extraction for recognition, which can be used directly with linear SVM. The NYU dataset, are mainly for indoor segmentation. There are two set of data and toolbox is available for extracting raw data and visualization. For NYU dataset, (`http://cs.nyu.edu/ silberman/`) , we loaded the labeled dataset and undersampled the image frames to reduce the size of data used. The 1449 images and depths maps were resampled from 480x640 to 120x160 which does not terribly affect the quality of the image based on visualization.

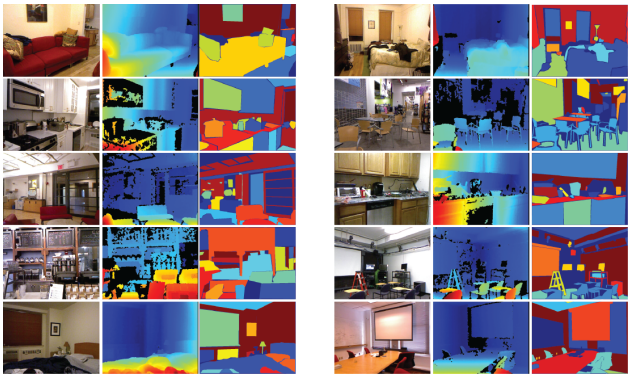

Figure 2. Samples of the RGB image, the raw depth image, and the class labels from the dataset.

2.2. Implementation of previous algorithm for comparison

Ashutosh et.al. (Saxena et al., 2005) used discriminatively-trained Markov Random Field (MRF) and designed feature convolutors that incorporates multi-scale local- and global-image features, and models both depths at individual points as well as the relation between depths at different points. We implemented similar idea for feature extraction. The features chosen to capture the local cues are texture variations, texture gradients, and haze. Images are also transformed to YCbCr color space to get a robust representation of the intensity which is more robust for texture feature extraction. The total features number is 646, including 17 convolutors, 3 resolution scales, 5 neighbour patches and 4 Vertical Patches. We used Support Vector based regression as a model to fit the depth information from the extracted

features. To train a model with Support Vector based regression, we randomly sampled the train image, processed the feature convolutor and constructed train data matrix. After training the model, it was applied on each pixel in the testing images. The depth map is computed and smoothed to get the results. Different from the referenced paper, we also implemented PCA to reduce the dimension of the features. The results shows that actually there are great redundancies in features and we reduced the dimension from 646 to 200 (covering 99 percent of energy).

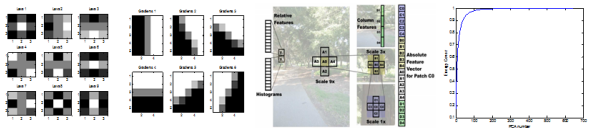

Figure 3. Method related figures from Saxena NIPS 2005, feature convolutor, multi-scale feature setting, PCA results (from left to right).

2.3. Building Deep Network

2.3.1. Re-generate larger dataset

The NYU labeled dataset we were using contains 1449 labeled RGB-D images. However, the samples are far from enough to train a complex deep network mapping 120x160 image to its corresponding depths map. To solve this problem, we proposed several possibilities.

First, the solution we will show in this manuscript is to modified the learning target and to use patch based resampling to create larger dataset. In order to learn a global high-resolution depth map, we re-designed the learning target to 1) to learn a low-resolution global averaged depth map; 2) to learn a high-resolution local relative (mean extraction) depth map. Both of these two problem will require smaller input and output data size. In addition, for the modified learning target, we can use sampling local and global patches (1000 patches per image) and adding noises to significantly improve the available labeled data.

Second, as inspired by Wei Song who was also working on similar topic, we can map the output depth-map to its sparse representation using Dictionary Learning (with kSVD algorithm etc.) which can also greatly decrease the output data size and the difficulty of the learning problem.

In this manuscript, we chose the first approach, and achieved 1000 times more samples. The 24x24 local patches were sampled from 120x160 images, while 96x96 global patches were sampled from 120x160 images and then down-sampled to 24x24 by taking the average. Then we used PCA/ZCA whitening to remove the redundancy.

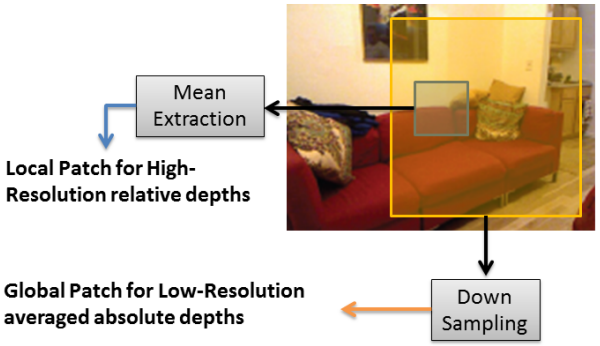

Figure 4. Illustration of two patch sampled in images to generate larger dataset and change the problem to 2 simpler learning problems.

Algorithm 1 Train deep network with patch based resampling and SAE

Input: image data I, size $m \times n$, depth groundtruth D, size $m \times n$
Re-sample: patch image $\hat{I}$, size $m_2 \times n_2$,
patched depth $\hat{D}$, size $m_2 \times n_2$, $m_2 < m, n_2 > n$
Whitening using PCA/ZCA
Train:
Un-supervised train W_{SAE}.
Supervised train W_{ouput}.
Initialize $W_{NN} = W_{SAE}$.
Supervised train $W_{entire} = [W_{NN}, W_{output}]$.
Output: deep network weights W_{entire}

2.3.2. Create sparse coding deep network

We have tried, Stacked Auto-Encoder(SAE) (Lee et al., 2007) and Convolutional Neural Network (CNN) (Lee et al., 2009) and Deep Brief Network initialized multilayer neural network. Taking SAE initialization as an exampled, we trained the SAE (3 layer with 200 node) to learn relative and averaged depth information from local and global patches respectively. The output layer used logistic regression and the groundtruth was re-mapped to [0,1] before training. The Stacked auto-encoders were trained in succession with self-taught pre-learning on each layers and supervised training was for the regression output layer. Then the Stacked Auto-Encoder was used to initialize the Deep Network and we than use supervised fine-tune on the entire network.

2.3.3. Reconstruct depth-map from learned patches

From learned patches, the full-scaled global depth-map can be reconstructed by fusing the results of local relative

Table 1. Parameter of different deep network architectures used.

ARCHITECTURE	HIDDEN LAYERS	DETAIL PARAMETERS
SAE+NN	2	100×100
CNN	2	$10c - 2p - 20c - 2p$
DBN+NN	2	100×100

Algorithm 2 Dictionary Learning based performance enhancement

Input: RGB image sample S_{RGB},
corresponding depth samples S_{depth}
Learn Dictionary: Joint Dictionary D_{joint}
$D_{joint} = [D_{RGB}\, D_{depth}] = argmin\|S_{RGB} - D_{RGB}x\|_2^2 + \alpha\|S_{depth} - D_{depth}x\|_2^2 + \lambda\|x\|_1$
Estimate Sparse Representation Coefficient from estimation:
$x = argmin\|I_{RGB} - D_{RGB}x\|_2^2 + \alpha\|\hat{I}_{depth} - D_{depth}x\|_2^2 + \lambda\|x\|_1$
Output: Enhanced estimation of depthmap with weighting
$\hat{I}_{depth} := (1-\beta)\hat{I}_{depth} + \beta D_{depth}x$

depth-map and global averaged depth-map. For the testing, Patches are sampled with overlap and the final depth-map was generated by taking the mean of the summation of averaged depth and relative depths.

2.3.4. PERFORMANCE IMPROVEMENT USING DICTIONARY LEARNING

Dictionary learning was proposed and applied widely in computer vision problems such as super-resolution. Here in this project, dictionary learning can be used to reduce the output dimension by train the network to learn the sparse representation coefficients of depth-map dictionary instead of the absolute depths map. In addition, we were also trying to use (pair-based) dictionary learning to improve the sparse encoding dictionary (Coates & Ng, 2011) to improve the reconstructed depth-map.

2.3.5. PLATFORM

Our current platforms are mainly Matlab and Python, while we are also start using GPU to accelerate the training larger networks with parallel computing.

3. Results

3.1. Result Comparison

Figure 5 provide an example of the learning algorithm output for the kernel based algorithm and the deep network based algorithm. The final depth-map for the deep network based algorithm was synthesized from two part of the learning result for local relative depth-map and global averaged depth-map shown in figure 5.

Table 2. Depth-map recovery accuracy using conventional (kernel based) algorithm and deep network based algorithm.

ALGORITHM	RMSE TRAIN	RMSE TEST
KERNEL BASED	5.1%	17.0%
CNN	13.3%	23.9%
SAE+NN	8.3%	14.9%
DBN+NN	8.8%	16.7%
DBN+NN+DL	7.0%	13.4%

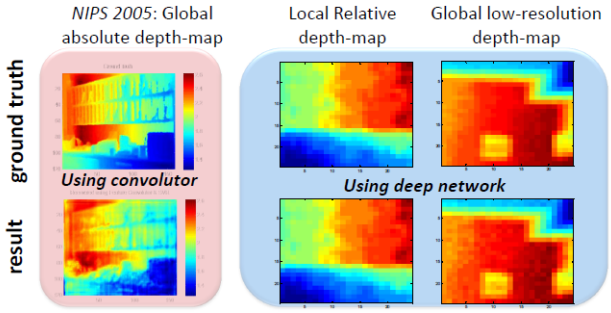

Figure 5. Result Comparison

3.2. Performance Analysis

Here we compared the performance of algorithms using different algorithms, including using conventional kernel based algorithm, deep network (CNN, SAE initialized NN, etc.). The criteria of Root-Mean-Square-Error was used and shown in 2 The RMSE using deep network based algorithm has advantages compared with traditional kernel based algorithm.

4. Discussion

We realized traditional and deep network based algorithms to decode 3D information from still 2D images.
In order to overcome the problem of data limitation, we redeisgned the learning target into two separate problems: to learn global averaged absolute depth and to learn local relative depth. Given these learning targets, we designed patch sampling strategy to greatly improve the labeled data size and improve the performance. The performance of recovering high resolution depth-map can be further improved by using Convolution, Pooling and Dictionary Learning.
Also we took a look at the weight learned and from the

results shown in figure 6, we thought the learned features contained both local and non-local information that contributes to the reconstruction of depth-map.

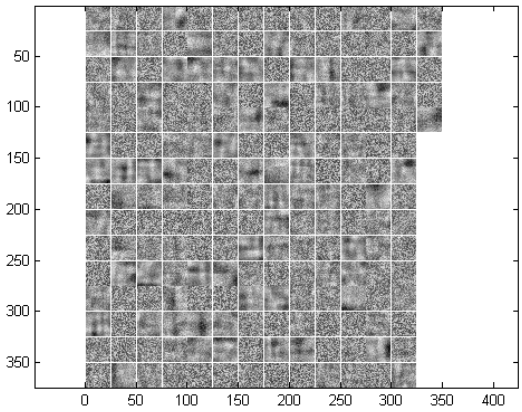

Figure 6. Visualize the weight in network

In addition, we are starting but not finished the work using GPU to train larger network. High performance and parallel computation is necessary for higher dimensional input and larger dataset. Current result on small-scale network has proved the concept, moreover, it has advantages in training efficiency (minutes V.S. days), more suitable for tuning the algorithm parameter and architectures, and is able to do on-line training.

5. Conclusion

We realized traditional and deep network based algorithms to decode 3D information from still 2D images. Results show the advantages of deep network based algorithm. Undergoing work using high performance computation and larger dataset potentially will result in better performance. This project has great This project has potential application in 3D vision, tracking and recognition, which we would like to keep exploring.

Acknowledgments

Thanks a lot for the instruction from Professor Andrew Ng, the ideas and guidance from Brody Huval and other TAs, as well as the discussion with Wei Song who was working on the similar project topic.

References

Coates, Adam and Ng, Andrew. The importance of encoding versus training with sparse coding and vector quantization. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 921–928, 2011.

Das, Subhodev and Ahuja, Narendra. Performance analysis of stereo, vergence, and focus as depth cues for active vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(12):1213–1219, 1995.

Gini, Giuseppina C and Marchi, Alberto. Indoor robot navigation with single camera vision. In *PRIS*, pp. 67–76, 2002.

Lee, Honglak, Ekanadham, Chaitanya, and Ng, Andrew. Sparse deep belief net model for visual area v2. In *Advances in neural information processing systems*, pp. 873–880, 2007.

Lee, Honglak, Grosse, Roger, Ranganath, Rajesh, and Ng, Andrew Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 609–616. ACM, 2009.

Michels, Jeff, Saxena, Ashutosh, and Ng, Andrew Y. High speed obstacle avoidance using monocular vision and reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning*, pp. 593–600. ACM, 2005.

Nagai, Takaaki, Naruse, Takumi, Ikehara, Masaaki, and Kurematsu, Akira. Hmm-based surface reconstruction from single images. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 2, pp. II–561. IEEE, 2002.

Ponce, Jean, Forsyth, David, Willow, Equipe-projet, Antipolis-Méditerranée, Sophia, d'activité RAweb, Rapports, Inria, Logo, and Alumni, Inria. Computer vision: a modern approach. *Computer*, 16:11, 2011.

Saxena, Ashutosh, Chung, Sung H, and Ng, Andrew. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems*, pp. 1161–1168, 2005.

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

Silberman, N. and Fergus, R. Indoor scene segmentation using a structured light sensor. In *Proceedings of the International Conference on Computer Vision - Workshop on 3D Representation and Recognition*, 2011.