FAST DEPTH ESTIMATION FROM SINGLE IMAGE USING STRUCTURED FOREST

Shuai Fang[1], *Ren Jin*[1], *Yang Cao*[2]

1.School of Computer and Information, Hefei University of Technology, Hefei 230009, China
2.Department of Automation, University of Science and Technology of China, Hefei 230027, China

ABSTRACT

Depth estimation from single image is an important component of many vision systems, including robot navigation, motion capture and video surveillance. In this paper, we propose to apply a structure forest framework to infer depth information from single RGB image. The core idea of our approach is to exploit the structure properties exhibit in local patches of depth map to learn the depth level for each pixel. We formulate the problem of depth estimation in a structured learning framework based on random decision forests. Each trained forest infers a patch of structured labels that are accumulated across the image to obtain the final depth map. Moreover, we systematically investigate a variety of depth-relevant features and the regression forest framework automatically determines the best feature combination and uses as input of structure forest. Our approach achieves quasi real-time performance that is orders of magnitude faster than state-of-the-art approaches, while also achieving state-of-the-art depth estimation results on the Make3D dataset.

Index Terms— depth estimation, structured forests, monocular image

1. INTRODUCTION

Depth estimation from single monocular image is an important problem in image understanding. Recent years have witnessed the prosperity of incorporating depth information into image processing applications, such as robotics, pose estimations and surveillance. Although some low cost RGBD imaging devices such as Kinect are current, most of the data commonly afforded in image processing applications is still RGB version. Moreover, due to the impact of sunlight, the low cost RGBD imaging devices cannot be directly used in outdoor applications. Therefore, depth estimation from single monocular images has been a hot spot of research given its wider application range [1–12]. Unfortunately, it is a particularly challenging task, as one given image may correspond to abundant real world scenarios [3].

Since there is no reliable cues can be exploited, estimating depth from a single image is indeed an ill posed problem. Different priors have been explored in previous methods. For example, simple geometric assumptions (i.e., box models) have been used to infer the spatial layout of a room [4, 5]. Similarly, the Manhattan, or blocks world assumption have proven effective to estimate the structure of outdoor scenes [6]. These methods, however, are limited to represent only particular scene structures, and therefore are not suitable for general scene depth estimations. In contrast, several methods have been proposed by incorporating additional sources of information, e.g., similar images [7], user annotation [8], and semantic labelling [9]. However, these methods rely on the performance of additional information and tend to propagate errors through different stages. Very recently, several work applies convolutional neural network in stages for single-image depth map prediction and achieves the state of the art results [2, 10]. Nevertheless, all these methods have high computational cost and require particular computer installation.

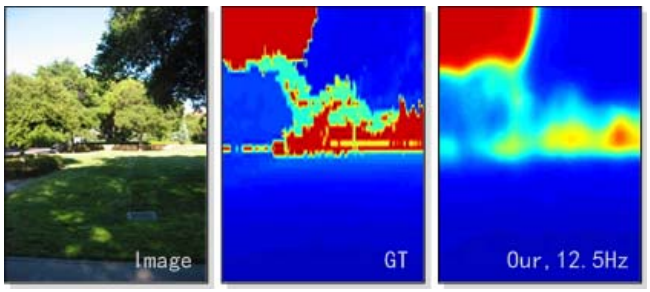


Fig. 1. Depth estimation result using our Structured Depth: Input image (Left), the ground-truth (Middle) and depth map estimated by our method (Right, and red is far, blue is close).

In this paper we propose to formulate depth estimation as a generalized structured learning problem, and apply a random forest framework to capture the inherent structured information in scene depth. Our approach does not rely on extra information, and is surprisingly computationally efficient. We can compute depth maps in quasi real-time without additional computing equipment, which is order of magnitude faster than state-of-the-art approaches. We formulate the problem of depth estimation as inferring a structured label for each local patch in the input image. Our approach firstly

Our source code is available on GitHub (https://github.com/king9014/rf-depth).

maps structured labels into a discrete space at each branch in the tree while training the decision trees, and then use these labels to determine the splitting function. Each forest infers a patch of structured labels that are accumulated across the image to obtain the final depth map, see Figure 1. We show state-of-the-art results on the Make3D [12] depth dataset. We demonstrate the potential of our approach by showing the performance on both estimation accuracy and computational efficiency.

2. ALGORITHM

In this paper we proposed to apply a random forest framework to infer depth information from single RGB image. We firstly formulated depth estimation as a structured learning problem based on random forest, and then systematically investigated a variety of depth-relevant features in the regression forest framework.

2.1. Structured random forest for depth estimation

Structured random forest is an extension of random forest with structured outputs. A decision tree $f_t(x)$ classifies a data sample $x \in X$ by recursively splitting left or right down the tree till a leaf node is reached. Specifically, each node j in the tree is associated with a binary split function:

$$h(x, \theta_j) \in \{0, 1\} \tag{1}$$

where θ_j is the parameter. If $h(x, \theta_j) = 0$, node j direct x left, otherwise right. The splitting process terminates until reaches at a leaf node. The output of decision tree on x is the inference stored at the reached leaf, which may be a label $y \in Y$ or a distribution over the labels Y.

A common choice of the split function $h(x, \theta_j)$ is to compare a single feature dimension of x to a threshold. Since an individual decision tree is prone to over fit, an ensemble of multiple independent decision trees, i.e. decision forest, is proposed by Criminisi et al [13]. Given a data sample x, the inference $f_t(x)$ from multiple trees are combined using an ensemble model into a single output. Although more sophisticated ensemble models may be employed, a simple model as presented in [14] is also effective in practice. To achieve a necessary diversity of trees, introducing randomness to nodes has proven to be applicable. Specifically, randomly subsampling the data can be used to train each tree, and randomly subsampling the features and splits can be used to train each leaf node.

Note that the leaf node reached by the tree relies only on the input x, while any kind of outputs y can be stored at each leaf node. This allows the extension of random forest with structured outputs. Kontschieder et al. [15] firstly propose to learn random forests for structured labels where the output labels represent a semantic labeling for an image patch. Dollar et al. [14] apply structured forest framework to learn an edge detector. In this paper, we extend the structured output to a more complex space so that the learning framework can be used to estimate depth from single image.

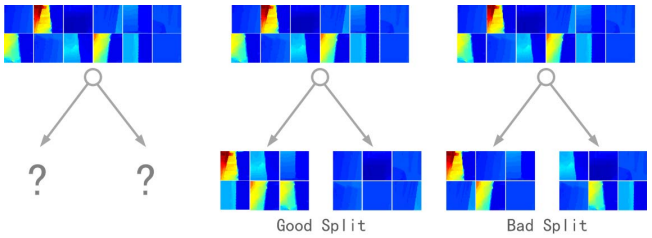


Fig. 2. Illustration of the decision tree node splits: (a) Given a set of structured labels as depth labels, a splitting function must be determined. Intuitively a good split (b) groups similar depth labels, whereas a bad split (c) does not.

Given an image patch, our approach takes the depth relevant features as the input $x \in X$, and the corresponding labels indicating the depth level as the output $y \in Y$. Since the labels within a local patch are highly correlative, the output spaces Y are high dimensional and complex, which presents a challenge computationally. For example, for a 16×16 local patch, there are N^{256} (N means the number of depth levels) unique depth labels. Accordingly, we need to use an approximate splitting method to reduce the dimensionality of Y. Specifically, we map the output labels $y \in Y$ into a discrete set of labels $c \in C$, where $C = \{1, ..., k\}$. The goal is clustering similar lables into same discrete label, see Figure 2. To better preserve the structure property of the output space, we apply Fuzzy c-means (FCM) method to discretize the output space to labels C. Given the discrete labels C, we can directly use existing random forest procedures to calculate information gain over C, and so as to learn structured random forests effectively.

2.2. Input depth relevant feature

Now we introduce how we extract the input features x. Different image features that are related to the properties of scene structure have been presented in previous work, such as texture variations, texture gradients, occlusion, haze and defocus. Although these features may only work well in particular scenarios, they provide a base for our approach. Our core idea is to utilize the structured forest framework to automatically determine the best feature combination for depth estimation. This offers our approach the flexibility to learn adaptive decision forests for specific situations.

Inspired by Saxena et al.s work [12], we extract two types of features given an image patch: absolute depth features and relative depth features, which are used to estimate the absolute and relative depth within the image patch, respectively. Moreover, considering objects in the outdoor scenes tend to be vertically connected to themselves (objects cannot float in

the air), we also extract the features of the column where the corresponding patch lies in.

The details of our input depth features are presented as follows.

a. SCN Features

In [12], Saxena et al. propose to use SCN features for absolute depth estimation. Similar to SCN, we use the output of 17 filters (9 Laws masks, 2 color channels in YCbCr space and 6 oriented edges) for each image patch. These filters capture the texture of a 3×3 patch and the edges at various orientations, which have high correlation to scene depth.

b. Dark Channel Features

Dark channel prior [16] is based on the observation that most local patches in outdoor images contain some pixels with very low intensities in at least one color channel. Therefore, distant objects tend to reflect more atmospheric light. From this perspective, dark channel is indeed a depth cue. Dark Channel of an image J is defined as:

$$J^{dark} = \min_{c\in\{r,g,b\}} (\min_{y\in\Omega(x)} (J^c(y))) \quad (2)$$

where J^c is a color channel of J and $\Omega(x)$ is a local patch centered at position x.

c. Color Features

Image color is also an important cue for inferring scene depth. For example, the sky region tends to be grey or blue, and the ground region tends to be green or dark. Moreover, since dark pixels may not exist in areas with low saturation, the color saturation of image can be used to compensate the regions where the dark channel prior may fail. In our approach, we compute the three channels in HSV, RGB and YUV space as color features.

d. Relative Features

We use a different feature vector to learn the dependencies between two neighboring patches. Different from SCN, we directly compute the pairwise differences between two neighboring pixels within the patch and use as relative depth features.

3. EXPERIMENT

We evaluated our method on a publicly available datasets Make3D [12]. we compared our results with Make3D [12], Semantic Labels [9], Depth Transfer [7] and Discrete-Continuous Depth [11], which represents the state-of-the-art methods for depth estimation from a single outdoor image. In addition, we also investigated the effectiveness of our features and discretization method.

For our quantitative evaluation, we used three following commonly-used metrics:

average relative error (rel): $\frac{1}{T}\sum_p \frac{|d_p^{gt}-d_p|}{d_p^{gt}}$

average log10 error (log10): $\frac{1}{T}\sum_p |log_{10}d_p^{gt} - log_{10}d_p|$

root mean squared error (rms): $\sqrt{\frac{1}{T}\sum_p (d_p^{gt}-d_p)^2}$

where d_p^{gt} and d_p were the ground-truth and predicted depths at pixel indexed by p, and T was the total number of pixels in all the evaluated images.

3.1. Implementation

We implemented the structured forest training based on the Dollar's Matlab toolbox with our own modification. Training was done on a standard desktop with Intel core i7 4770 CPU and 32GB memory. Detailed parameters were presented as follow:

Input features: Our approach predicted a structured 16×16 depth label from a larger 32×32 image patch. In order to reduce the dimension of column feature, we resized the input image to 256×336. Color features were extracted from the original scale, then the color features had 9 channels. We computed SCN and dark channel features at 2 scales (original and half resolution), a total of 36 channels. We also used row script as 1 channel of features. So the features were total of 46 channels. For each image patch, we computed the absolute and relative depth features, and the corresponding column features. For absolute features, we downsampled by a factor of 2, to yield totally $16 \times 16 \times 46 = 11776$ dimensions. For relative features, we downsampled patch to a resolution of 5×5 and computed all candidate pairs, to yield $C_{5\times5}^2 \times 46 = 13800$ dimensions. For column absolute features, we downsampled to a total of $32 \times 4 \times 46 = 5888$ dimensions. For column relative features, we downsampled to $C_{16\times2}^2 \times 46 = 22816$ dimensions. So the total dimensions of features were 54280 .

Training parameters: At each node splitting, we discreted the depth space to 4 labels c using FCM algorithm.

Ensemble model: We trained 8 trees for the structured forest. In the ensemble model, we used a checkerboard pattern to improve results, where 4 trees and a stride of 2 pixels were used. For a 16×16 output patch, each pixel received $16^2T/4 \approx 64T$ votes. In practice, we used $T = 4$ and thus the output of each pixels in the final depth map was averaged over 256 votes.

3.2. Results

The Make3D dataset contained 534 images with corresponding depth maps. We used 400 images for training and 134 images for testing as offical. Due to the limited range and resolution of the sensor used to collect the ground-truth, far away pixels were arbitrarily set to depth 80 in the original dataset. Accordingly, we reported errors based on two different criteria: (C_1) Errors were computed in the regions with ground-truth depth less than 70; (C_2) Errors were computed in the entire image.

In Table $1-3$, we compared the results of our approach with popular methods of single image depth estimation. Note that, using criteria C_1, we outperformed than other algorithm with rel, log10 and rms. Due to the influence of the sensor's precision, in the criteria C_2, our rms was slightly less than

[11] while rel and log10 were still ahead of other methods and except [2]. In addition, in terms of computational efficiency, our algorithm was far ahead of other methods. In the 4 trees prediction, it could be used in quasi real-time applications without the use of GPU (computing acceleration card) at 12.5 frames per second. In the case of a loss of accuracy (1-tree), it can run at 14 frames per second.

Method	rel(C_1)	log10(C_1)	rms(C_1)
Depth Transfer[7]	0.355	0.127	9.20
DC Depth[11]	0.335	0.137	9.49
DCNF-FCSP[2]	**0.331**	0.119	7.77
Our Method(1-Tree)	0.353	0.123	8.17
Our Method(4-Tree)	0.334	**0.117**	**7.39**

Table 1. Compare with other method of C_1, lower is better, C_1 and C_2 are two criteria, see text for details.

Method	rel(C_2)	log10(C_2)	rms(C_2)
Make3D[12]	0.370	0.187	NR
Semantic Labels[9]	0.379	0.148	NR
Depth Transfer[7]	0.361	0.148	15.10
DC Depth[11]	0.338	0.134	12.60
DCNF-FCSP[2]	0.330	0.133	14.46
Our Method(1-Tree)	0.351	0.140	15.44
Our Method(4-Tree)	0.334	0.134	14.98

Table 2. Compare with other method of C_2, lower is better.

Method	Time-consuming
Make3D[12]	5s
Semantic Labels[9]	30s
Depth Transfer[7]	60s
DCNF-FCSP[2]	10s
Our Method(1-Tree)	**0.07s**
Our Method(4-Tree)	0.08s

Table 3. Compare with other method of time-consuming.

Table 4 illustrated the effect of various label discretization methods on the accuracy of depth estimation. We included 4 methods, PCA, GMM, K-means and FCM in our experiment. As can be seen, FCM achieved the best performance for the discretization of the depth labels. However, since PCA was the fastest method, it will be a good choice when accelerating is required.

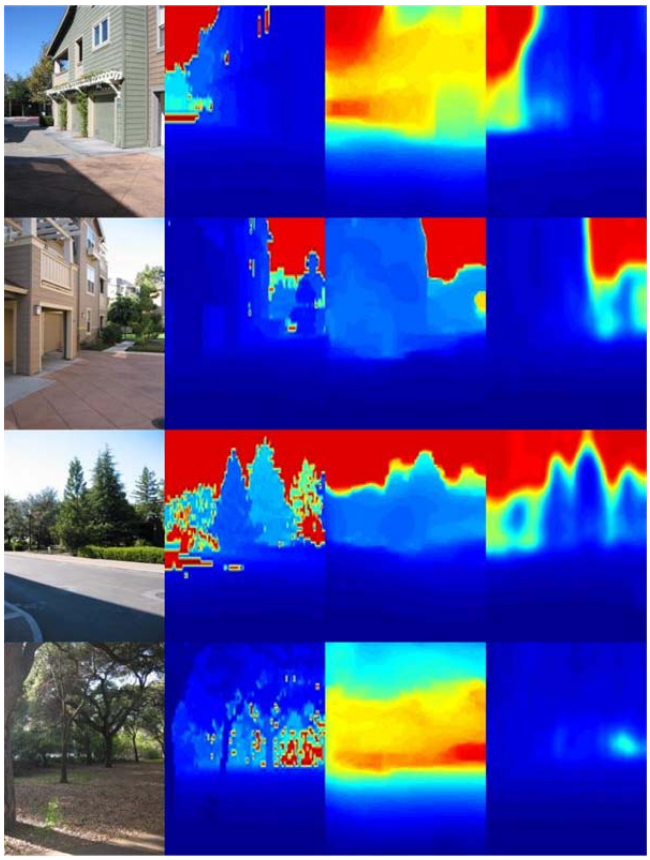
Figure 3 presents the qualitative comparison of the estimated depths on the Make3D dataset. As can be seen, our approach yields better visualizations aligning to local details.

4. CONCLUSION

In this paper, we presented an approach to estimate depth from single image, which was capable of quasi real-time frame rates while achieving state-of-the-art accuracy. We proposed to apply the structure learning framework to depth estimation, in which the output space was higher dimension and much complex. We also demonstrated that the proposed framework could determine the best combination of depth relevant features. The experimental results on Make3D dataset showed the potential of our approach on both estimation accuracy and computational efficiency. The limitation of our work was that we only investigated the features used in outdoor scene. This will be our future work to extend our approach to indoor case.

Discretization Method	rel(C_2)	log10(C_2)	rms(C_2)
PCA	0.339	0.139	14.85
K-means	0.341	0.140	15.02
GMM	0.342	0.140	14.97
FCM	**0.336**	**0.138**	**14.84**

Table 4. Compare the effects of discretization method on the accuracy.



(a) Image (b) Ground-truth (c) Depth Transfer[7] (d) Our method

Fig. 3. Qualitative comparison of the depths estimated on the Make3D dataset. Color indicates depth (red is far, blue is close).

References

[1] Changchang Wu, J. Frahm, and M. Pollefeys, "Repetition-based dense single-view reconstruction.," *Rodng*, vol. 42, no. 7, pp. 3113 – 3120, 2011.

[2] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, pp. 1–1, 2015.

[3] David Eigen, Christian Puhrsch, and Rob Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Eprint Arxiv*, pp. 2366–2374, 2014.

[4] V HedauD HoiemD Forsyth, "Thinking inside the box: Using appearance models and context based on room geometry," *Springer Berlin Heidelberg*, pp. 6316:224–237, 2010.

[5] David C. Lee, Abhinav Gupta, Martial Hebert, and Takeo Kanade, "Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces," in *In NIPS*, 2010, pp. 1288–1296.

[6] Abhinav Gupta, Alexei A. Efros, and Martial Hebert, *Blocks World Revisited: Image Understanding Using Qualitative Geometry and Mechanics*, Springer Berlin Heidelberg, 2010.

[7] Kevin Karsch, Ce Liu, and Sing Bing Kang, "Depth extraction from video using non-parametric sampling," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 36, no. 11, pp. 775–788, 2012.

[8] B. C. Russell and A. Torralba, "Building a database of 3d scenes from user annotations," in *IEEE Conference on Computer Vision & Pattern Recognition*, 2010, pp. 2711–2718.

[9] Beyang Liu, S. Gould, and D. Koller, "Single image depth estimation from predicted semantic labels," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 1253 – 1260.

[10] David Eigen and Rob Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," *International Conference on Computer Vision*, 2015.

[11] Miaomiao Liu, Mathieu Salzmann, and Xuming He, "Discrete-continuous depth estimation from a single image," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014, pp. 716–723.

[12] Saxena Ashutosh, Sun Min, and Andrew Y Ng, "Make3d: learning 3d scene structure from a single still image.," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 31, no. 5, pp. 824–840, 2009.

[13] J. Shotton A. Criminisi and E. Konukoglu, *Decision forests: A unifiedframework for classification, regression, density estimation, manifoldlearning and semi-supervised learning.*, 2012.

[14] Piotr Dollar and C. Lawrence Zitnick, "Fast edge detection using structured forests," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 8, pp. 1558–1570, 2014.

[15] Peter Kontschieder, Samuel Rota Bulo, Horst Bischof, and Marcello Pelillo, "Structured class-labels in random forests for semantic image labelling.," in *IEEE International Conference on Computer Vision*, 2011, pp. 2190–2197.

[16] Kaiming He, Jian Sun, and Xiaoou Tang, "Single image haze removal using dark channel prior," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2341–2353.