Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs

Bo Li[1,2], Chunhua Shen[2,3], Yuchao Dai[4], Anton van den Hengel[2,3], Mingyi He[1]
[1] Northwestern Polytechnical University, China
[2] University of Adelaide, Australia; [3] Australian Centre for Robotic Vision
[4] Australian National University

Abstract

Predicting the depth (or surface normal) of a scene from single monocular color images is a challenging task. This paper tackles this challenging and essentially under-determined problem by regression on deep convolutional neural network (DCNN) features, combined with a post-processing refining step using conditional random fields (CRF). Our framework works at two levels, super-pixel level and pixel level. First, we design a DCNN model to learn the mapping from multi-scale image patches to depth or surface normal values at the super-pixel level. Second, the estimated super-pixel depth or surface normal is refined to the pixel level by exploiting various potentials on the depth or surface normal map, which includes a data term, a smoothness term among super-pixels and an auto-regression term characterizing the local structure of the estimation map. The inference problem can be efficiently solved because it admits a closed-form solution. Experiments on the Make3D and NYU Depth V2 datasets show competitive results compared with recent state-of-the-art methods.

1. Introduction

Both depth and surface normal estimation are common intermediate components in understanding 3D scene structure. Many approaches have been proposed to tackle these two problems. We propose a common deep learning framework for predicting both depth and surface normals in this work. Depth estimation is to predict pixel-wise depth for a single or multiple images. It was shown that depth information can benefit tasks such as recognition [1, 2], human computer interaction [3], and 3D model reconstruction [4]. Traditional techniques have predominantly worked with multiple images to make the problem of depth prediction well posed, which include N-view reconstruction, structure from motion (SfM) and simultaneous localization and mapping (SLAM). However depth estimation from a monocular, static viewpoint lags far behind its multi-view counterpart. This is mainly due to the fact that the problem is ill-posed and inherently ambiguous: a single image on its own does not provide any depth cue explicitly (*i.e.*, given a color image of a scene, there are infinite number of 3D scene structures explaining the 2D measurements exactly).

When specific scene dependant knowledge is available, depth estimation or 3D reconstruction from single images can be achieved by utilizing geometric assumptions such as the "Blocks World" model [5], the "Origami World" model [6], shape from shading [7] and repetition of structures [8]. These cues typically work for images with specific structures and may not be applied as a general framework.

Data-driven depth estimation methods, predicting scene geometry directly by learning from data, have gained popularity. Typically, such approaches recast the underlying depth estimation problem in a scene labeling pipeline by exploiting relationship between image features and depth [9, 10]. These method can be roughly categorized as parametric approaches and non-parametric approaches. Parametric approaches such as [9] and [11] make a planar model for each super-pixel, where the model parameters are inferred by exploiting different unary, pair-wise and high-order cues. These work generally uses hand crafted features [10, 12]. In contrast, non-parametric approaches such as [13, 11, 14] adopt a depth transfer framework, where the whole depth map is transferred from retrieved candidate depth maps. Usually, a final optimization is required to enforce constraints on the depth map. However, these methods generally search the training data set online, thus prohibiting their use in real world applications.

To tackle the above shortcomings in depth estimation from a single image, in this paper, we present a new framework consisting of depth regression via deep features and depth refining via hierarchical CRF. First, to exploit the inherent relation between a color image and its associated depth, we use a deep network and formulate the problem of depth estimation as a regression problem. Multi-scale deep

features are extracted by a deep CNN network, and a regressor is trained. To our knowledge, this may be the first work showing the pre-trained multi-scale deep features [15] can be effectively transferred to the depth estimation problem. Second, to refine the estimation of the regressor and achieve efficient estimation, we introduce a hierarchical continuous conditional random field (CRF) model to take various potentials into consideration,thus refining the depth (or surface normal) estimation from the super-pixel level to the pixel level. In contrast to existing work, our model does not need to encode any kind of geometric priors explicitly (all the geometric relations such as occlusion can be encoded implicitly by exploiting a large amount of training data), thus enabling its powerful generalization ability in real world applications.

It is worth noting that our framework is top-to-bottom in that it works from the super-pixel level to pixel level, while existing work such as [10, 11] adopts a bottom-to-top strategy. This brings the following benefits (a) It reduces the computation burden dramatically by extracting pre-trained CNN features at the super-pixel level only; (b) It avoids over-smoothing on the boundary and preserves small objects. Furthermore, the inference of our model has a closed-form solution thus the implementation of our framework is efficient. We show that using the same framework, we can estimate surface normals with minimal modification to the network parameters. This is not surprising because one can always calculate the surface normals given the depth information.

2. Related work

In this section, we briefly review recent advances in depth and surface normal estimation from a single image. Seminal work by Saxena *et al*. [9, 16] tackles the problem with a multi-scale Markov Random Field (MRF) model, with the parameters of the model learned through supervised learning. The work models the plane parameters as a linear function of the hand-crafted texture based, super-pixel shape and location based features. The model is only applicable to scenes where horizontal aligns with the ground plane. By contrast, our framework is much more general, which does not enforce strong assumptions about the scene layout.

Liu *et al*. [17] estimated the depth map from predicted semantic labels, simplifying the problem and achieved improved performance with a simpler MRF model. Recently Ladicky *et al*. [18] showed that perspective geometry can be used to improve results and demonstrated how scene labelling and depth estimation can benefit each other under a unified framework, where a pixel-wise classifier was proposed to jointly predict a semantic class and a depth label from a single image. Besides these parametric methods, recent work such as [13, 11, 14] tackle the depth estimation problem in a non-parametric way, where the whole depth map is inferred from candidate depth maps. However, these methods need to access a large color-depth data set to retrieve candidate depth maps at run time.

Most recently, Eigen *et al*. [19] presented a framework by training a large deep Convolution Neural Network (CNN) and regressing low-resolution depth maps directly from the raw color images. To train such a large network, an extremely large (*i.e*., hundreds of thousands of images) data set of labelled color-depth image pairs is required. By contrast, our work only needs hundreds of training images, which makes our method applicable in scenarios where only limited training samples are available. In addition, the regressed depth map by their work is blurred. On the contrary, we achieve rather realistic depth map with our effective CRF model.

To date, data-driven learning based normal estimation methods have not been well studied. It is believed that this may be due to the lack of available training data [12]. Ladicky *et al*. [12] presented a promising method to estimate surface normals from a single image using machine learning. The core idea is to discriminatively train a regressor using boosting techniques. Note that they rely on multiple hand-crafted features such as texton, SIFT, local quantized ternary patters. With CNNs, one can learn all the features from raw pixels.

Our work is also related to recent works on transfer learning and deep learning. In [15], Krizhevsky *et al*. trained a large deep CNN on the ImageNet data set and achieved a performance leap. Recently, more and more work shows that pre-trained CNN features can be transferred to new classification or recognition problems and boost remarkable performance [20, 21]. Our work here is the first one showing that *pre-trained deep CNN feature can be transferred to depth and surface normal estimation*. Since our common framework for depth and surface normal estimation is the same, in the sequel, we mainly focus on depth estimation.

3. Our approach

Our approach to pixel-level single image depth estimation consists of two stages: depth regression on super-pixel and depth refining from super-pixels to pixels. First, we formulate super-pixel level depth estimation as a regression problem. Given an image, we obtain super-pixels. For each super-pixel, we extract multi-scale image patches around the super-pixel center. A deep CNN is then learned to encode the relationship between input patches and the corresponding depth. Second, we refine the depth estimate from the super-pixel level to pixel level by inference on a hierarchical conditional random filed (CRF). Different potentials are taken into consideration as both super-pixel and pixel levels. Importantly, our MAP inference problem has a

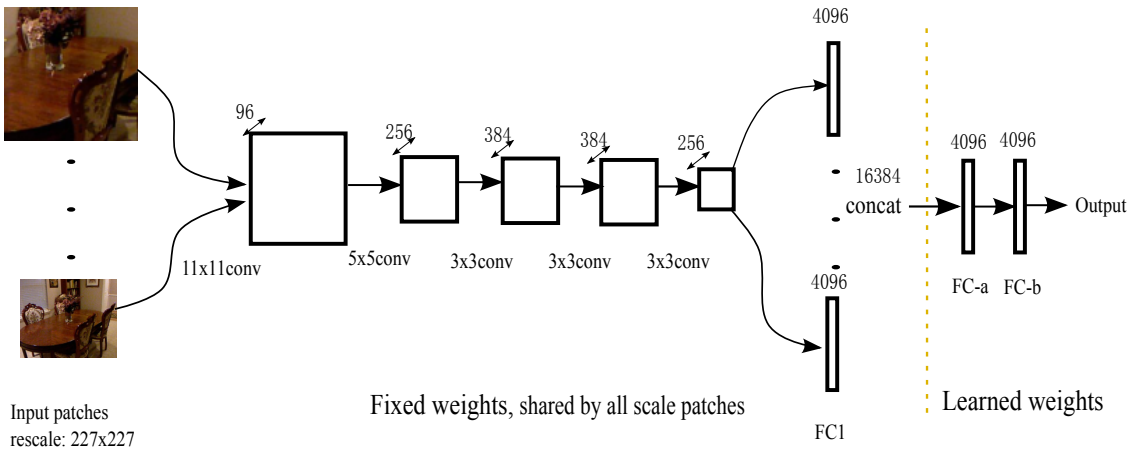

Figure 1: *Visualization of our multi-scale framework. Each patch goes through five convolutional layers and the first fully-connected layer (here transferred from AlexNet). The features are concatenated before they are fed to two additional fully-connected layers. We then refine the predictions from the CNN by inference of a hierarchical CRF (not shown here; see text for details).*

closed-form solution. An overall CNN architecture is presented in Fig. 1. Note that the fixed-weights part of the CNN can be transferred from pre-trained models such as Krizhevsky *et al.*'s AlexNet [15] or the deeper VGGNet [22].

3.1. Depth regression with CNNs

Most existing work predicts depth by regressing with hand-crafted features. We note that, for pixel-based approaches a local feature vector extracted from a local neighborhood area can be insufficient to predict the depth label. Thus, a certain form of context, combining information from neighboring pixels capturing a spatial configuration, has to be used. To encode the depth, we use a deep network and formulate depth estimation as a regression problem, as shown in Fig. 1. Here the first five convolutional layers and the first fully-connected layer (FC1) are transferred from the AlexNet, and the weights are fixed, shared by all input patches. The outputs of FC1 are then concatenated to feed into two additional fully-connected layers (FC-a and FC-b). The weights of FC-a and FC-b are learned using training data.

To predict the depth of a pixel/super-pixel, we first extract patches of different sizes around that point, then resize all the patches to 227×227 pixels to form the multi-scale inputs. Details of the network and training details are described in Section 3.3.

The multi-scale structure is inspired by the relationship between depth and scale. In addition, context information often includes rich cues as to the depth of a point. In our experiments, we provide extensive comparisons and analysis to demonstrate that a large-size patch with rich context information and multi-scale observations are critical to the task of depth regression.

Effect of multi-scale features and long-range context In our depth regression deep network we use multi-scale patches to extract depth cues. Since local features alone may not be sufficient to estimate depth at a point, we need to consider the global context of the image [10]. With increasing the patch size, more information is included and image context encodes depth more accurately. Therefore to regress depth or surface normals from image patches, it makes sense to use large patches to extract non-local information.

In real-world data sets, generally there is a scale change between scenes due to varying focal lengths. To deal with these scaling effects, one strategy is to extract the characteristic scale for each point according to scale space theory [23]. Here we exploit another strategy by applying a discrete multi-scale approach due to efficiency consideration. We extract patches of different sizes to capture the scaling effect across the data set.

To analyze the effects of both multi-scale and context, we conducted experiments to evaluate performance of depth estimation on NYU V2 data set with increasing size of single patch. Experimental results are reported in Table 1. Performance of depth estimation gradually improves with the increasing patch size. With increased number of scales, performance of depth estimation improves with respect to the increase in the number of image patches. Both experiments demonstrate that multi-scale image patches and large image context are of critical importance in achieving good performance in depth estimation.

3.2. Refining the results via hierarchical CRF

So far we have shown how depth may be predicted for super-pixels using regression. Now our goal is to refine the predicted depth or surface normals from the super-pixel

input patch size	$\delta < 1.25$ $\delta < 1.25^2$ $\delta < 1.25^3$	rel	$\log_{10}$	rms
55×55 pixels	47.00% 76.85% 91.19%	0.328	0.129	1.052
121×121 pixels	53.48% 82.89% 94.41%	0.280	0.112	0.972
271×271 pixels	57.68% 86.27% 95.84%	0.254	0.103	0.889
407×407 pixels	59.15% 86.71% 96.13%	0.247	0.099	0.8717
Final result (4 scales of patches)	**62.07%** **88.61%** **96.78%**	**0.232**	**0.094**	**0.821**

Table 1: *Depth estimation results on the NYU V2 data set under different sizes of single-scale image patch setting and multi-scale setting. The error metrics definition could be found in Section 4*

level to pixel level. To address this problem, we formulate a hierarchical CRF built upon both the super-pixel and pixel levels. The structure of our hierarchical CRF is illustrated in Fig. 2.

More specially, let $\mathcal{D} = \{d_1, ..., d_n\}$ be the set of depth for each pixel, $\mathcal{S} = \{s_1, ..., s_m\}$ be the set of super-pixels. n is the total number of pixels in one image, and m is the number of super-pixels. In our model, we assume the depth value of the super-pixel to be the same as its centroid pixel. Thus, we remove the super-pixels variable explicitly in our formulation.

Here our energy function is:

$$\mathbf{E}(\mathbf{d}) = \sum_{i \in \mathcal{S}} \phi_i(d_i) + \sum_{(i,j) \in \mathcal{E}_{\mathcal{S}}} \phi_{ij}(d_i, d_j) + \sum_{\mathcal{C} \in \mathcal{P}} \phi_{\mathcal{C}}(\mathbf{d}_{\mathcal{C}}), \tag{1}$$

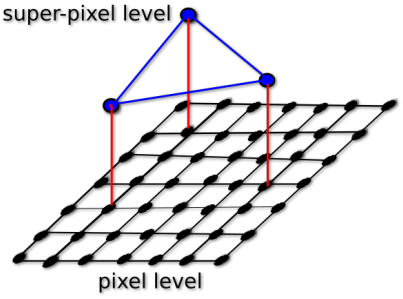

Figure 2: *Illustration of our hierarchical CRF. Two layers are connected via region hierarchy. The blue nodes represent the super-pixels, where the depth is regressed by the proposed CNN. The blue edges between the nodes represent the neighborhoods at the super-pixel level; and the black edges represent the relation at the pixel level and the red edges represent the relation between these two levels which is forced to be equal.*

where $\mathcal{E}_{\mathcal{S}}$ denotes the set of pairs of super-pixels that share a common boundary and $\mathcal{P}$ is the set of patches designed on the pixel level, aiming at capturing the local relationships in depth map.

Generally speaking, this is similar to a high-order CRF defined on both super-pixel and pixel levels. Now, we explain the potentials used in the energy function Eq. (1), where the first two potentials are defined on the super-pixel level, while the third one is defined on the pixel level.

Potential 1: Data term

$$\phi_i(d_i) = \left(d_i - \overline{d_i}\right)^2, \tag{2}$$

where $\overline{d_i}$ denotes the depth regression result from our multi-scale deep network, This term is defined at the super-pixel level, measuring the quadratic distance between the estimated depth d and regressed depth $\overline{d}$.

Potential 2: Smoothness at the super-pixel level

$$\phi_{ij}(d_i, d_j) = w_1 \left(\frac{d_i - d_j}{\lambda_{ij}}\right)^2, \tag{3}$$

this pairwise term enforces coherence between neighbouring super-pixels. Here we define the smoothness at super-pixel level. The quadratic distance is weighted by λ_{ij}, *i.e.* the color difference between connected super-pixels in the CIELUV color space [24].

Potential 3: Auto-regression model Here we use the auto-regression model to characterize the local correlation structure in the depth map, which has been used in image colorization [25], depth in-painting [4], and depth image super resolution [26, 27]. Depth maps for generic 3D scenes contain mainly smooth regions separated by curves. The auto-regression model can well characterize such local structure. The key insight of the auto-regression model is that a depth map can be represented by the depth map itself locally. Denote by d_u the depth value at location u. The predicted depth map by the model could be expressed as:

$$d_u = \sum_{r \in \mathcal{C}/u} \alpha_{ur} d_r, \tag{4}$$

where $\mathcal{C}/u$ is the neighbourhood of pixel u and α_{ur} denotes the model auto-regression coefficient for pixel r in the set of $\mathcal{C}/u$. The discrepancy between the model and the depth map (*i.e.*, the auto-regression potential) can be expressed as:

$$\phi_{\mathcal{C}}(\mathbf{d}_{\mathcal{C}}) = w_2 \left(d_u - \sum_{r \in \mathcal{C}/u} \alpha_{ur} d_r\right)^2. \tag{5}$$

We need to design a locally auto-regression predictor α with the available color image. Here we set $\alpha_{ur} \propto \exp(-(g_u - g_r)^2/2\sigma_u^2)$, and $\sum \alpha_{ur} = 1$, where g represents the intensities value of corresponding pixels, and σ_u is the variance of the intensities in the local patch around u.

Theoretically, the parameters w_1, w_2 could be learned by maximizing our conditional log-likelihood. In our formulation, we estimate w_1, w_2 by cross validation on the training data.

A closed-form solution Once the parameters in our hierarchical CRF are determined, the MAP solution can be obtained in closed form, due to the least-squares loss (Gaussian CRF). For convenience of expression, we express the energy function Eq. (1) in a matrix form:

$$\mathbf{E}(\mathbf{d}) = \|\mathbf{H}\mathbf{d} - \overline{\mathbf{d}}\|_2^2 + w_1\|\mathbf{Q}\mathbf{H}\mathbf{d}\|_2^2 + w_2\|\mathbf{A}\mathbf{d}\|_2^2, \quad (6)$$

where $\overline{\mathbf{d}_s}$ is the output of the regression network, $\mathbf{H}$ is the indication matrix that selects corresponding super-pixels from the entire set, $\mathbf{Q}$ expresses the neighbouring relationship in the super-pixel level while $\mathbf{A}$ is a neighbouring matrix corresponding to the auto-regression model in local patch.

As the energy function is quadratic with respect to $\mathbf{d}$, a closed-form solution can be derived algebraically:

$$\mathbf{d}_{\mathrm{MAP}} = (\mathbf{H}^\top\mathbf{H} + w_1\mathbf{H}^\top\mathbf{Q}^\top\mathbf{Q}\mathbf{H} + w_2\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{H}^\top\overline{\mathbf{d}}. \quad (7)$$

3.3. Implementation details

Before proceeding to the experimental results, we give implementation details for our method.

In both data sets, we utilize SLIC [28] to obtain the super-pixels. For depth regression, we fix the multi-scale patch sizes at 55×55, 121×121, 271×271, 407×407 pixels. These patches are extracted from the original image and resized to 227×227 pixels, which is the input size of our depth regression network. For the NYU V2 dataset, the number of training samples is 800,000 while the number is 400,000 for the Make3D data set, *i.e.*, around 1000 points are sampled from each image in both data sets. During training and testing, we transfer the ground truth depth value into log space. The trade-off parameters in the depth refining are set as: $w_1 = 1$, $w_2 = 0.01$ for the Make3D data set, while $w_1 = 10$, $w_2 = 0.01$ for the NYU V2 data set.

The proposed depth regression network is trained using stochastic gradient decent with a batch size of 100 samples, momentum of 0.9, and weight decay of 0.0004. Weights for the convolution layers C1, C2, ..., FC1 are initialized by the pre-trained AlexNet model [15]. The weights of FC-a, FC-b are randomly initialized with standard deviation 0.01. Besides, we add the ReLU layer and dropout layer after these two layers. The size of layer F-cat is 16384. The size of both FC-a and FC-b layers are 4096. For more detail about the "shared weights", please refer to [29]. The learning rate is initialized as 0.01, and divided by 10 after 5 cycles through the training set. In our experiment, we trained the network for roughly 20 to 30 epochs on both data sets.

As for the surface estimation, we have utilized almost the same setting with minimal modification. Here we used the VGGNet (VGG16) model [22] to transfer the first a few convolutional layers and the first FC layer. All the other parameters (learning rate, weight decay, etc.) are the same as the case of AlexNet. We have used three-scale patch sizes of 100×100, 224×224, 400×400 pixels. The size of layer F-cat is 12288. The FC-a and FC-b layers have 1024 and 512 neurons respectively. In order to refine the predicted surface normals map, we transfer the surface normal vectors into the spherical coordinate, *i.e.*, $(x, y, z) \rightarrow (\theta, \phi)$. This transformation avoids the normal constraint. In addition, we refine the θ map and ϕ map respectively, with $w_1 = 0.1$, $w_2 = 0.01$ for both θ and ϕ.

The Euclidean loss function is used,

$$\mathbf{E} = \frac{1}{2N}\sum_{i=1}^{N}\|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2^2, \quad (8)$$

where $\mathbf{x}_i$ could be a ground-truth depth or surface normal vector. $\hat{\mathbf{x}}_i$ is the correspondent regression value.

4. Experimental results

In this section, we report our experimental results on single image depth estimation for both outdoor and indoor scenes. We use the Make3D range image data set and the NYU V2 Kinect data set, as they are the largest open data set we can access at present. We compare our method with all the state-of-the-art methods published recently.

In addition, we present an analysis of the underlying problem and our method. Specifically, we first give a baseline implementation with depth regression only; *i.e.*, without depth refining, thus explaining the roles of both components in achieving final depth map. Secondly, we analyze the influence of the size of super-pixel in depth estimation.

Error metrics For quantitative evaluation, we report errors obtained with the following error metrics, which have been extensively used [9, 17, 19, 18, 11].

- Threshold: % of d_i s.t. $\max\left(\frac{\hat{d}_i}{d_i}, \frac{d_i}{\hat{d}_i}\right) = \delta < thr$;
- Mean relative error (rel): $\frac{1}{|T|}\sum_{d\in T}|\hat{d} - d|/d$;
- Mean $\log_{10}$ error ($\log_{10}$): $\frac{1}{|T|}\sum_{d\in T}|\log_{10}\hat{d} - \log_{10} d|$;
- Root mean squared error (rms): $\sqrt{\frac{1}{|T|}\sum_{d\in T}\|\hat{d} - d\|^2}$.

Here d is the ground truth depth, $\hat{d}$ is the estimated depth, and T denotes the set of all points in the images.

4.1. NYU2 data set

The NYU V2 data set contains 1449 images, of which 795 images are used as a training set and 654 images are used as a testing set. All images were resized to 427×561

pixels in order to preserve the aspect ratio of the original images. In Table 2, we compare our method with state-of-the-art methods: depth transfer [13], discrete-continuous depth estimation [11], pulling things out of perspective [18]. Our method outperforms these methods by a large margin under most of the error metrics. We achieve comparable if not better performance compared with the most recent multi-scale deep network method [19], which used hundreds of thousands of labelled images to train the network.

In Fig. 3, we provide a qualitative comparison of our method with the work in [13], [11], [18], and [19]. From the figure, we observe that usually our method preserves the structure of the scene better than counterpart methods, which is much desired in many applications such as 3D modelling.

To analyse the contribution of each component in our method (depth regression and depth refining), we provide experimental results for depth regression only as a baseline, where pixels in each super-pixel are assigned identical depth from our depth regression network. By comparing the results with and without depth refining, the importance of our depth refining strategy becomes clear.

4.2. Make3D data set

The Make3D data set consists of 534 images with corresponding depth maps. There are 400 training images and 134 test images. All images were resized to 460×345 pixels. It is worth noting that this data set was published many years ago, the resolution and distance range of the depth image is rather limited (only 55×355). Furthermore, it contains noise in the locations of glass window *etc*. These limitations have some influence on the training stage and the resulting error metrics. Therefore we report errors based on two different criteria as presented in [11]: ($\mathbf{C}_1$) Errors are computed in the regions with ground-truth depth less than 70; ($\mathbf{C}_2$) Errors are computed in the entire image. We compare our method with the state-of-the-art methods such as depth transfer [13], discrete-continuous depth estimation [11]. As illustrated in Table 3, our method clearly outperforms these methods. Furthermore, we present a qualita-

Method	$\delta < 1.25$ $\delta < 1.25^2$ $\delta < 1.25^3$	rel	$\log_{10}$	rms
Depth transfer [13]*	- - -	0.374	0.134	1.12
Liu *et al.* [11]*	- - -	0.335	0.127	1.06
Our method*	**63.95%** **90.03%** **97.41%**	**0.223**	**0.091**	**0.759**
Ladicky *et al.* [18]	54.22% 82.90% 94.09%	-	-	-
Eigen *et al.* [19]	61.1% **88.7%** **97.1%**	**0.215**	**0.094**	0.871
Regression only	59.94% 87.20% 96.30%	0.243	0.098	0.851
Our method	**62.07%** 88.61% 96.78%	0.232	**0.094**	**0.821**

Table 2: *Depth estimation errors on the NYU v2 data set, $*$ means that errors are computed over the non-zero depth in the raw ground truth depth map. "regression only" is our model with no CRF refining.*

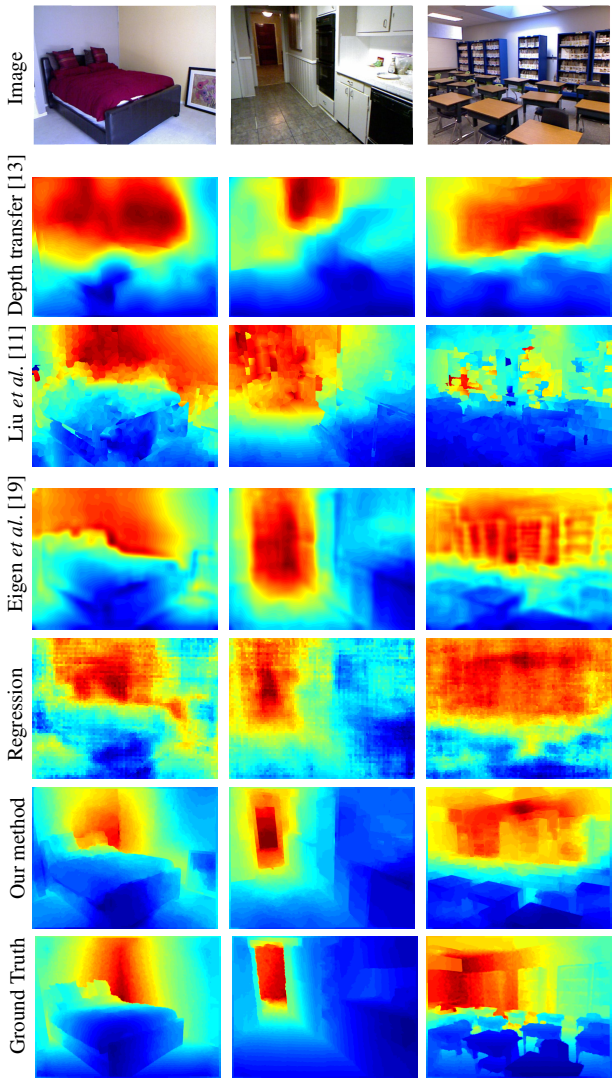

Figure 3: *Qualitative comparison of the estimated depth map on the NYU V2 data set with our method and some state-of-the-art methods. Color indicates depth (red is far, blue is close).*

Method		rel	$\log_{10}$	rms
Depth transfer [13]	C1	0.355	0.127	9.2
	C2	0.361	0.148	15.1
Liu *et al*. [11]	C1	0.335	0.137	9.49
	C2	0.338	0.134	12.6
Regression only	C1	0.283	0.094	7.01
	C2	0.281	0.102	10.74
Our method	C1	**0.278**	**0.092**	**7.188**
	C2	**0.279**	**0.102**	**10.27**

Table 3: *Depth estimation errors on the Make3D data set.*

tive comparison of the depth estimation with these methods on representative images from Make3D data set, which further demonstrates the superior performance of our method.

4.3. Performance analysis

We present an analysis over our depth regression and depth refining framework. Formally, we investigate the effect of different sizes of super-pixel, aiming at understanding the trade off between efficiency and effectiveness. Then, we give an illustration of how our framework can be extended to predict depth for images not similar to the training data set, thus demonstrating the generalization capability empirically.

Effect of the size of super-pixels In our depth regression and depth refining framework, depth regression is conducted at the super-pixel level while depth refining is done at the pixel level by inferring with CRF. The size of the super-pixels has an effect on the final depth estimation result. A larger super-pixel size results in a smaller number of regression tasks thus the evaluation is more efficient. However, the depth refining on such a sparse node structure may cause performance deterioration. A smaller super-pixel size can reduce the difficulty in depth refining but increase the CPU time. Meanwhile, if using very small super-pixels or pixel-wise regression at the extreme, it may cause a non-smoothness effect. Therefore, there should be a trade-off in setting the size of super-pixels. Here we present experimental results on the NYU V2 data set by setting different sizes of super-pixels. Results were reported in Table 4. Clearly, performance in depth estimation improves with decreasing the size of super-pixels. However, decreasing the size below 10 does not improve performance further. Therefore, in this paper, we fix the size of super-pixels to 10.

Generalization capability Finally, we present an illustration on how the regression-refining framework can be used to predict depth for images not related to the training data set, thus illustrating the generalization ability of the proposed method in Fig. 5.

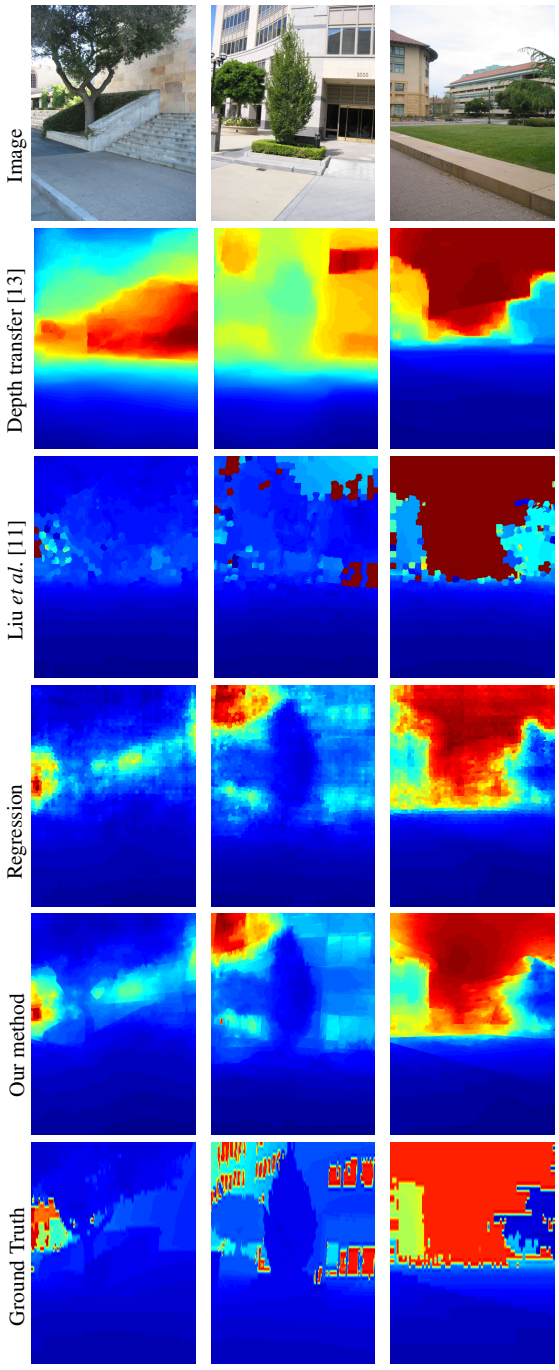

Figure 4: *Qualitative comparison of the depth map estimated by our method and the state-of-the-art [11] and [13]. Color indicates depth (red is far, blue is close).*

4.4. Estimation of surface normals

We now report the results of surface normal estimation. Table 5 compares the performance of our method against a few recent methods. As we can see, our method compares on par with the best results. Note that we have directly

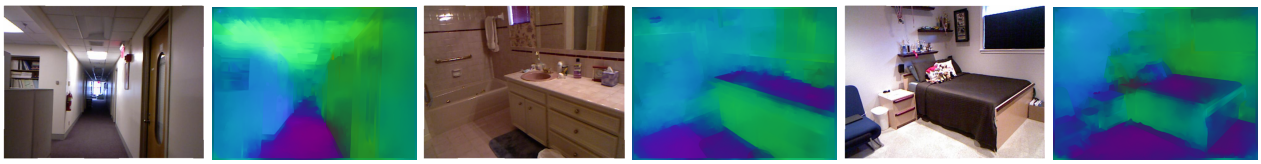

Figure 6: *Qualitative results to show the surface normal estimation of our method on the NYU V2 data set. Our method successfully captures the layout of the indoor scenes.*

SLIC size	$\delta < 1.25$ $\delta < 1.25^2$ $\delta < 1.25^3$	rel	$\log_{10}$	rms
7	61.82% 88.63% 96.82%	0.232	0.094	0.825
10	62.07% 88.61% 96.78%	0.232	0.094	0.821
15	59.80% 87.74% 96.57%	0.2410	0.0979	0.859
20	56.37% 85.73% 95.63%	0.2574	0.1045	0.9245
30	49.21% 80.07% 92.86%	0.3003	0.1217	1.0738

Table 4: *Depth estimation results on the NYU V2 data set with varying sizes of super-pixels.*

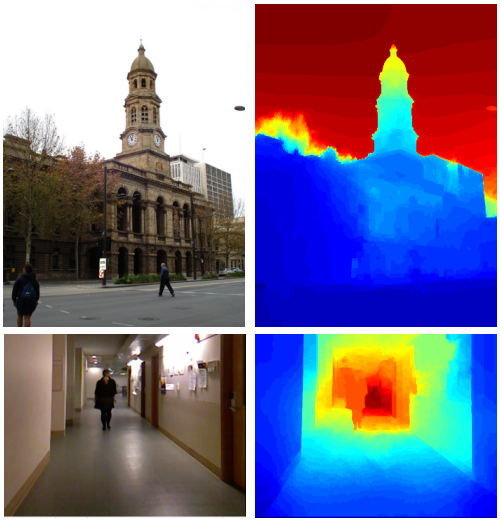

Figure 5: *Demonstration of the generalization capability of our method, where we estimate depth map for images not in the NYU V2 or Make3D data set.*

trained a regression model for this surface normal estimation task. It is expected that following the idea of converting surface normal regression into classification (triangular coding), better performance can be achieved. We here do not pursue this strategy to show the simplicity and versatility of our framework.

We also demonstrate some qualitative results in Fig. 6. One can see that our method can successfully capture the overall layout of the indoor scenes.

Method	mean err (°)	median (°)	% 11.25°	22.5°	30°
[6]	35.1	**19.2**	37.6	53.3	58.9
[12]	32.5	22.4	27.4	50.2	60.2
[30]	34.2	30.0	18.6	38.6	49.9
Ours	**30.6**	27.8	19.6	40.6	53.7

Table 5: *Surface normal estimation results on the NYU V2 data set. The results are evaluated on valid pixels. The last three columns show the percentages of "good pixels" against three thresholds.*

5. Conclusions

In this paper, we have presented a new and common framework for depth and surface normal estimation from single monocular images, which consists of regression using deep CNNs and refining via a hierarchical CRF. With this simple framework, we have achieved promising results for both tasks of depth and surface normal estimation.

In the future, we plan to investigate different data augmentation methods to improve the performance in handling real-world image transformations. Furthermore, we plan to explore the use of deeper CNNs. Our preliminary results show that improved depth estimation can be obtained with VGGNet, compared with AlexNet. In addition, the effect of joint depth and semantic class estimation with deep CNN features also deserves attention.

Acknowledgements

B. Li's contribution was made when he was a visiting student at the University of Adelaide, sponsored by the Chinese Scholarship Council.

This work was also in part supported by ARC Grants (FT120100969, DE140100180), National Natural Science Foundation of China (61420106007), and the Data to Decisions Cooperative Research Centre, Australia.

References

[1] X. Ren, L. Bo, and D. Fox, "RDB-D scene labeling: Features and algorithms," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2012, pp. 2759–2766. 1

[2] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013. 1

[3] S. R. Fanello, C. Keskin, S. Izadi, P. Kohli, D. Kim, D. Sweeney, A. Criminisi, J. Shotton, S. B. Kang, and T. Paek, "Learning to be a depth camera for close-range human capture and interaction," *ACM T. Graphics*, vol. 33, no. 4, pp. 86, 2014. 1

[4] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Proc. Eur. Conf. Comp. Vis.*, pp. 746–760. Springer, 2012. 1, 4

[5] A. Gupta, A. Efros, and M. Hebert, "Blocks world revisited: Image understanding using qualitative geometry and mechanics," in *Proc. Eur. Conf. Comp. Vis.*, pp. 482–496. 2010. 1

[6] D. Fouhey, A. Gupta, and M. Hebert, "Unfolding an indoor origami world," in *Proc. Eur. Conf. Comp. Vis.*, pp. 687–702. 2014. 1, 8

[7] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, "Shape-from-shading: a survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 8, pp. 690–706, 1999. 1

[8] C. Wu, J.-M. Frahm, and M. Pollefeys, "Repetition-based dense single-view reconstruction," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2011, pp. 3113–3120. 1

[9] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–840, 2009. 1, 2, 5

[10] A. Saxena, S. Chung, and A. Ng, "3-d depth reconstruction from a single still image," *Int. J. Comp. Vis.*, vol. 76, no. 1, pp. 53–69, 2008. 1, 2, 3

[11] M. Liu, M. Salzmann, and X. He, "Discrete-continuous depth estimation from a single image," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2014, pp. 716–723. 1, 2, 5, 6, 7

[12] L. Ladick, B. Zeisl, and M. Pollefeys, "Discriminatively trained dense surface normal estimation," in *Proc. Eur. Conf. Comp. Vis.*, pp. 468–484. 2014. 1, 2, 8

[13] K. Karsch, C. Liu, and S. B. Kang, "Depth extraction from video using non-parametric sampling," in *Proc. Eur. Conf. Comp. Vis.*, pp. 775–788. Springer, 2012. 1, 2, 6, 7

[14] J. Konrad, M. Wang, and P. Ishwar, "2d-to-3d image conversion by learning depth from examples," in *Proc. IEEE Conf. Computer Vis. & Pattern Recogn. Workshops*. IEEE, 2012, pp. 16–22. 1, 2

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105. 2, 3, 5

[16] A. Saxena, J. Schulte, and A. Y. Ng, "Depth estimation using monocular and stereo cues," in *Proc. IEEE Int. Joint Conf. Artificial Intell.*, 2007, vol. 7. 2

[17] B. Liu, S. Gould, and D. Koller, "Single image depth estimation from predicted semantic labels," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2010, pp. 1253–1260. 2, 5

[18] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2014, pp. 89–96. 2, 5, 6

[19] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014. 2, 5, 6

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2014. 2

[21] M. Oquab, L. Bottou, I. Laptev, J. Sivic, et al., "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2013. 2

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learning Representations*. 2015. 3, 5

[23] T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales," *J. Applied Statistics*, vol. 21, no. 2, pp. 224–270, 1994. 3

[24] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2009, pp. 670–677. 4

[25] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM T. Graphics*, 2004, vol. 23, pp. 689–694. 4

[26] J. Diebel and S. Thrun, "An application of markov random fields to range sensing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 291–298. 4

[27] O. Mac Aodha, N. D. Campbell, A. Nair, and G. J. Brostow, "Patch based synthesis for single depth image super-resolution," in *Proc. Eur. Conf. Comp. Vis.*, pp. 71–84. Springer, 2012. 4

[28] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012. 5

[29] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678. 5

[30] D. F. Fouhey, A. Gupta, and M. Hebert, "Data-driven 3D primitives for single image understanding," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2013. 8