Region-Support Depth Inference from a Single Image

Anonymous CVPR submission

Paper ID ****

Abstract

Depth inference from a single image is a long-standing challenge in the computer vision community. It is technically ill-posed since monocular cues like the scale ratio or the objects variance are ambiguous for inference of depth. In this paper, we propose the region support as the inference guidance to resolve the ambiguity. The region support is the regional segmentations each of which consists of pixels at similar depths. It formulates the depth inference working in two steps: first inferring the regional depth and then refining the regional depth to pixel-wise depth. The region-support depth inference is realized a novel network consisting of three modules: generation of region support, computation of regional depth and regional refinement to pixel-wise depth. We first obtain the region support by a novel clustering-based segmentation module and then use the obtained region support as additional channel to infer the initial pixel-wise depth, where the two modules share the same multi-scale feature from a pyramid unit. Then we use the region support as masks on the initial pixel-wise depth to compute the regional depth. The regional refinement to pixel-wise depth separately works on the initial pixel-wise depth map by the target refinement and on the regional depth map by the variance refinement. The target refinement makes the depth among each region close to its stable mean depth while the variance depth models the variance on the basis of the mean depth of each region. The final refined depth map fuses the output of both refinements. From the experiments on the NYU and KITTI datasets, we can see both the regional depth and the two-steps regional refinement can remarkably reduce the ambiguity and raise the inference accuracy.

1. Introduction

The depth estiimation methods can be widely used in robotics, autonomous vehicles, recognition tasks, visual localization and scene analysis [11, 4, 1, 5]. As monocular images are the most readily available data, the depth inference from a single image has attracted considerable attention in the past decades [15, 3, 18]. It mostly relies on monocular cues like the scale ratio, feature variance of objects and so on [3, 9], but these cues are ambiguous to guide the depth inference. Many strategies have been proposed to resolve the ambiguous problem, for example, using the additional supportive guidance like semantic information to guide the inference [17, 10, 20], discretizing the continuous depth into interval values to regularize the inference space [14], using the coarse-to-fine framework to arrange the inference and so on. In this paper, we propose the region support as the guidance for the depth inference and design a novel region-support depth inference network which realizes the depth inference by two stages: infering the regional depth and refining the regional depth to the pixel-wise depth.

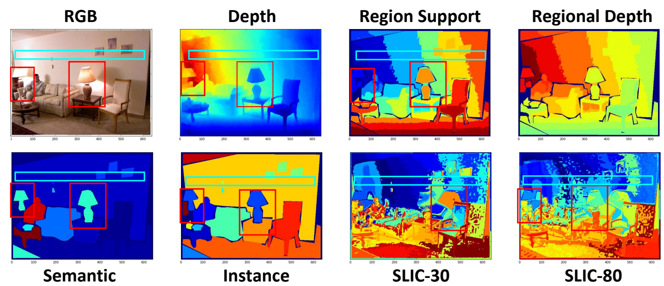

Figure 1. The comparison of regional guidances. In the red boxes, we can see the semantic guidance has conflicts with the different depth of the 'light'. In the blue box, we can see both the instance and the semantic guidance cannot offer supportive guide of the 'wall' which varies a lot on the depth. The SLIC guidance over-segments objects too much, which makes the regional guidance too sensitive to adapt different scenes. As for our region support, we can see it ensures the supportive information from the semantic and instance guidance meanwhile resolves the ambiguitous situation beween different depth but same labels.

The region support is a special kind of segmentation of regions each of which consists of pixels at similar depths. Like many regional guidances, it guides the inference by the assumption that with the same regional label, the variance of depth should be stable and continuous. However, most regional guidance can not truely realize this assumption, as

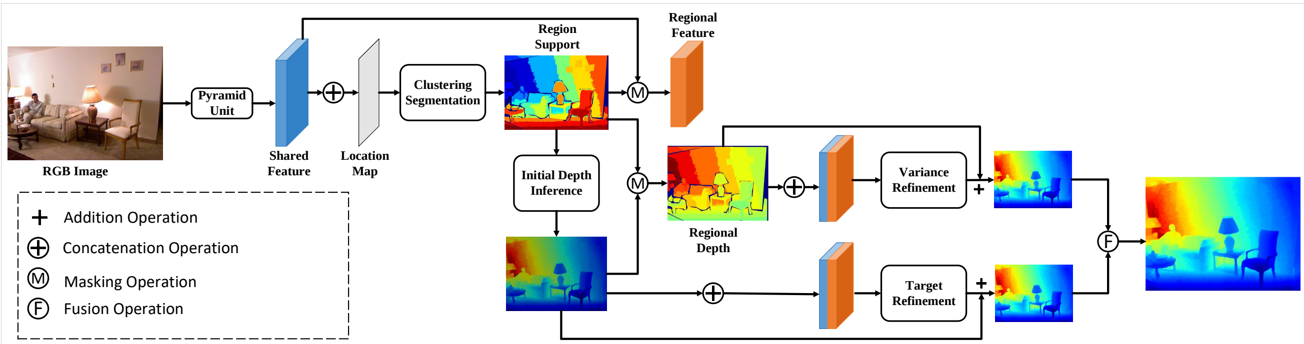

Figure 2. The region-support depth inference network. The network consists of three module: generation of region support, computation of the regional depth and the regional refinement. A pyramid unit provides the multi-scale feature for all of the three modules. We first use the clustering-based segmentation module to obtain the region support. Then we use the region support as additional information to support the inference of the initial pixel-wise depth. After that, we can simply compute regional depth and regional feature using the region support as masks on the initial pixel-wise depth and shared feature maps. The regional refinment module seperately works on the regional depth and the initial depth map by the variance and the target refinement. The variance refinement infers the depth by computing the variance based on the mean value of each region while the target module computes the variance of depth with the target of the mean value.

shown in Figure 1, the semantic guidance has problem in the red box where the 'lights' are at different depth but has the same semantic label. Besides, in the blue box, objects like 'wall' cross a long range of depth but the label is the same for the region. As for the super-pixel based approaches, the division of regions is too sensitive to adapt for different scenes. Compared with these guidance, the region support can effectively handle the above ambiguious situations.

We use the region support to formulate the depth inference working in two steps: infering the regional depth and refining to the pixel-wise depth. As shown in Figure 2, we design the region-support depth inference network to carry out this two-steps formulation, which consists of three modules: generation of the region support, computation of the regional depth and regional refinement to the pixel-wise depth. To generate the region support, we design a novel clustering-based segmentation module which consists of the pyramid feature extraction unit and the learnable clustering segmentation unit. The pyramid unit provides the multi-scale feature as the representation and then the clustering unit takes into the multi-scale feature to generate the region support by the clustering-based segmentation like [2, 13]. Then we combine the shared feature with the region support to infer the pixel-wise initial depth through several convolutional layers. The region support serves as masks for the initial depth to compute the mean depth of each region, meanwhile the masks are applied to the shared feature to compute the regional feature map.

The regional refinement is conducted by the variance refinement and target refinement. The variance refinement refines the regional depth by learning the variance on the mean depth of each region while the target refinement refines the initial depth to close to its mean depth of each region. The outputs of the two refinement sub-modules fuses with each other to form the final depth map. A unified loss function is proposed to ensure the end-to-end training, which consists of the discriminative loss for the clustering segmentation and berhu loss for the refinement module. For the data without regional labels, we design a novel semi-supervised loss which only uses the depth as supervision to train the clustering module.

The region-support depth inference method reaches state-of-the-art performance on the NYUv2 [12, 16] and KITTI [5] datasets. From the alation analysis, we can see both the formulation of the regional depth and regional refinement can effectively resolve the ambiguities and constraint the depth inference space. Our contribution can be summarized in three folds:

- We propose the region support as the guidance for the depth inference from a single image, which formulates the depth inference as the infering of regional depth and refining the regional depth to pixel-wise depth.
- We design a novel region-support depth inference network with three modules: the generation of region support, the computation of the regional depth and regional refinement to pixel-wise depth. A unified loss for each module is designed for the end-to-end training with a novel discriminative loss for the clustering-based segmentation.
- The region-support depth inference reaches the state-of-the-art performance on challenging datasets. Both the formulation of the regional depth and regional refinement are proven to be effective to simplify the depth inference.

2. Related Work

There are many strategies to resolve the ambiguities during the depth inference while the most related three strategies to our method is using the semantic guidance, coarse-to-fine formulation and discretizing the continuous depth into intermediate depth. Many works found that combining the semantic segmentation task with the depth inference can improve the performance for both tasks. Mostly the intergration of the semantic segmentation and depth inference are realized by the concatenation operation which treats the semantic segmentation as additional guidance for the inference. Then the two tasks could be jointly optimized by the end-to-end training with the multi-task loss. This kind of strategy can reach sufficient performance benefiting from the learnable usage of guidance from the joint optimization, however, the semantic guidance has its natural conflicts with depth. Therefore, we propose the region support to release the conflicts between regional guidance and true depth. In addition, we form a much more explicit usage of regional guidance by using the region support to form the regional depth and regional refinement.

The coarse-to-fine strategy is a widely used strategy for depth inference task, which seperates the inference process into two steps: computing the coarse depth and refining the coarse depth to the fine. Many methods built a two-stages network to realize this strategy with different approaches to compute the coarse depth or refine the coarse depth. Some works contributed on the computation of coarse depth by designing a more effective network to extract the multi-scale feature, while others foucs on the post-processing like the conditional random field (CRF) and iterative optimization to enhance the refinement module. Compared with these methods, we define the coarse depth as the regional depth and design the variance and target refinement to obtain the pixel-wise depth from the regional depth. Current, discretizating the continious depth into discrete intermediate depth shows ability to resolve the ambiguities by transforming the regression of depth into the classification of discrete ranges of depth. The regional depth formulation can be deemed as a special discretization which is more adaptive than the constant discretized ranges. Since the discrete ranges changes for different scenes, we use the region support to adaptively discrete the continuous depth into the discrete regional depth and then refine this discrete depth to continuous depth.

3. Region-Support Depth Inference

As shown in Figure 2, the region-support depth inference network has three modules: the generation of region support, the computation of regional depth and the regional refinement to pixel-wise depth. All of these modules share the multi-scale feature from a spatial pyramid unit which consists of 34 residual layers and a spatial pooling layer. The generation of region support module uses $2D$ convolutional layers to generate the clustering feature from the shared feature and uses a clustering-based segmentation approach to generate the region support. The computation of regional depth module first combines the shared feature with the obtained region support to infer the initial depth and then use the obtained region support as masks to compute the regional coarse depth. The regional refinement module seperately works as the variance refinement on the regional depth and the target refinement on the initial depth. We fuse the outputs of these two refinements to form the final depth map. All of the three modules can be trained end-to-end by a unified loss function for the clustering segmentation and the regional refinement.

3.1. Generation of Region Support

3.1.1 Spatial Pyramind Unit

We adopt a spatial pyramid architecture to utilize multi-scale features for the determination of the region support. A standard ResNet [7] with $4, 10, 5, 5$ basic residual blocks is applied as the feature extraction. We conduct the sub-sampling on the second and third block, so we get a sub-sampled feature map with the $1/4$ resolution size after the residual layers. Then we employ the spatial pyramid pooling unit [19] to enhance the multi-scale features by pooling the feature map into four lower scales, i.e., $1/8, 1/20, 1/80$ and $1/240$. The sub-sampled feature maps respectively pass through one convolutional layer with $1/4$ input channel. After that, the sub-sampled feature maps are resized into the $1/4$ resolution size by bilinear interpolation and concatenated together with the $1/4$ resolution residual feature map. To fuse the multi-scale feature, we adopt two de-convolutional units each of which has three layers: one convolutional layer before the de-convolutional layer and one after the de-convolutional layer. The concatenation skip connection is used between the up-sampled feature map and the previous feature map from the output of the first and second residual block.

3.1.2 Clustering Segmentation

The region support is determined by the depth distribution of the pixels, so it is quite different from the semantic or instance segmentation. Besides, the label for each region can not be strictly corresponding to a constant depth value because the direct inference of the absolute depth for each region is as difficult as infering the pixel-wise depth. So the label for the region support can only reveal whether the pixels are at similar depth but do not represent the real depth. The common detection based approaches like R-CNN or Mask-RCNN is not useful for this task since the labels vary dynamically for different scenes. Inspired by the Braban-

Algorithm 1: Clustering Segmentation

Input: Feature Map F
Output: Region Support C

$n = 0$
1. Iteratively random select the seed $p = (x, y)$ from R, until all $C(p)$ labeled to a certain region
2. Compute the related pixels $P = (|F - F(p)|^2 < \delta_{\text{var}})$
3. Compute the clustering center $c = mean(F(P))$
4. Use c to replace $F(p)$ and repeat step 2,3 twice.
5. Get the region P
6. Fuse the P with C, if the overlap between P and $C > 70\%$, $C(P) = n$, else the non-overlapped $C(P)$=c+1
7. $c = c + 1$
8. If $c > 80$ return C

δ_{var} is empirically set to 0.16 at the begining and changed dynamically during training. The final stable value is used for test.

der et al. [2] and Novotny et al. [13], we find out that the clustering-based segmentation can satisfy the dynamical labeling task since the label can be determined by the distribution of the feature itself.

The shared multi-scale feature map first concatenates with the two-channel location map to form the fused feature, which is different from the add operation in [13]. Then four convolutional layers are used to refine the fused map and adjust the channel to 16 to release the computational resource for the clustering. We design a modified mean-shift clustering unit to obtain the segmentation from the refined feature map which is shown in the Algorithm 1. Compared with the common mean shift approach, we only shift the clustering center for twice to save the time for segmentation and add the fusion part to get a more stable division. Although this clustering unit is fully differencial, it is not involved into back-propogation because it will cost too much memory due to the iterative clustering. To make this part trainable, we propose the supervised and semi-supervised loss.

3.1.3 Supervised and Semi-supervised Discriminative Loss

The discriminative loss function for the clustering-based segementation is formulated as the pull and push forces between and within clusters [2]. The pull force is realized by penelizing the intra-cluster variance, which can be indicated like

$$L_{var} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{p=1}^{N_c}\lceil \|\mu_c - F(p)\| - \delta_v \rceil_+^2 \tag{1}$$

Here, C is the number of regions, N_c is the number of pixels in region c, $\mu_c$is the mean feature of region c and F is the fused feature map. $\|\cdot\|$ is the L_1 distance and $\lceil x \rceil_+$ is the $max(x, 0)$ function. The variance loss makes the embedding of same region close the mean feature of this region while the δ_v is the maxinum variance for each cluster. The push force is realized by the distance loss which is shown as

$$L_{dist} = \frac{1}{C(C-1)}\sum_{\substack{c_A=1 \\ c_A \neq c_B}}^{C}\sum_{c_B=1}^{C}\lceil 4\delta_v - \|\mu_{c_A} - \mu_{c_B}\| \rceil_+^2 \tag{2}$$

The distance loss pushes the clusters away from each other and at least has a distance of $4\delta_v$ between two cluster centers. The final discriminative loss $L = L_{var} + L_{dist}$ is similar to the loss function for general instance segmentation [2, 13]. Novotny et al. [13] found directly minimizing the inner variance can reach better performance, but we need the variance δ_v to employ the Algorithm 1, so we still use the clap version as [2] but with a dynamical δ_v. In our experiments, the δ_v is first set by 0.16 and the updated after each training step as $\delta_v = \lambda \times \delta_v + (1 - \lambda) \times \sum_{c=1}^{C} \delta_c$, where δ_c is the variance in region c and λ is set by 0.99 in our experiments.

For the data only having depth supervision D, we propose the semi-supervised loss. We first use the Algorithm 1 to obtain the region support and then compute the mean depth for each region. After that, we can get the variance for each pixel. We assume that if the variance is more than δ_d which is empirically set as the $\frac{\max(D)-\min(D)}{20}$, this pixel is clustered in the wrong area. So the regions can be divided as two sub-sets R and W which are the right division and wrong division. For the right division, we want the feature of the pixels are more close to the cluster center, while for the wrong division, we want these pixel move away from the cluster center. We design the L_{wr} to realize the pull and push force which can be expressed as

$$L_{rw} = L_r + L_w, \tag{3}$$

$$L_r = \frac{1}{R}\sum_{r=1}^{R}\frac{1}{N_r}\sum_{p=1}^{N_r}\lceil \|\mu_r - F(p)\| - \delta_v \rceil_+^2, \tag{4}$$

$$L_w = \frac{1}{W}\sum_{w=1}^{W}\frac{1}{N_w}\sum_{p=1}^{N_w}\lceil 2\delta_v - \|\mu_r - F(p)\| \rceil_+^2. \tag{5}$$

When the L_{rw} becomes zeros, all of the pixels are within a distance of δ_v to its cluster center while at least $2\delta_v$ from the other cluster center. So we can simply use the Algorithm 1 to get the region support.

3.2. Computation of Regional Depth

The computation of regional depth works in two steps where we first compute the initial pixel-wise depth and then compute the mean depth for each region as the regional depth. We adopt four residual units to reduce the channel of the shared feature and then use three common convolutional units to infer the initial depth for each pixel, where the final inference layer output the one channel depth map. Then we use the obtained region support as masks to compute the mean depth for each region. Since there are unlabeled pixel if the number of regions becomes more than 80, these unlabed pixels can not be used to compute the regional depth, so we only use the initial depth as the regional depth for there unlabeled pixels. At the same time, we also use the region support as masks on the shared feature to compute the regional feature as the supportive information for the refinement module

3.3. Regional Refinement

The regional refinement works as the variance refinement and the target refinement. The variance refinement computes the variance of the mean depth value to infer the exact depth for each pixel. We combine the regional depth with the shared feature and the regional feature map as the input for the variance refinement. The variance refinement module consists of five convolutional units where the last unit removes the Relu and the Normalization layer. Then the regional depth adds the output to form the refined regional depth. As for the target refinement, it shares the structure of the variance refinement but with the combination of initial depth, regional feature and shared feature as input. The output adds the initial depth map to form the refined result. The target refinement computes the variance of the initial depth for each pixel with regarding to the mean dpeth of its region. The final depth map fuses the two refined depth where we use a very simple average fusion in this paper.

A behu loss [8] is adopted to train the refinement module which can be indicated as

$$L_d = \begin{cases} \left|\log(d) - \log(d^{'})\right| & \left|\log(d) - \log(d^{'})\right| \leq \delta_d \\ \frac{\left\|\log(d) - \log(d^{'})\right\| + \delta_d^2}{2\delta_d} & \left|\log(d) - \log(d^{'})\right| > \delta_d \end{cases}. \tag{6}$$

Here, the d is the ground truth depth, $d^{'}$ is the predicted depth, $|\cdot|$ is the L_1 distance, $\|\cdot\|$ is the L_2 distance and $\delta_d = \frac{1}{5} max(log(d_p) - log(d_p^{'}))$.

3.4. Implementation Details

All of the convolutional and residual layers are followed by the GroupNorm with group num 1 and ReLU unit except for the last layers of the two refinement sub-modules. We adopt the duplication padding for all of the convolutional operation. We use the nearest interpolation for the spatial pyramid pooling unit. The kernel size for the last three layers of the two refinement and the 14 layer is set to 1×1 while the deconvolutional layers have the 4×4 kernel size.

3.5. Experiment

Table 1. The Region-support Depth Inference Network

unit	index	stride	input	output
Generation of Region Support				
Pyramid Feature Extraction				
Conv	1-2	1	3	64
Residual	3-6	1	64	128
Residual	7-16	2	128	128
Residual	17-22	2	128	256
Residual	23-27	1	256	256
SPP	28-30	2,5,20,60	256	512
conv	31-32	1	512	128
deconv	33	2	128	128
concatenate 33 with output of 16				
conv	34	1	256	192
deconv	35	2	192	192
concatenate 35 with output of 6				
conv	36	1	256	256
Clustering Segmentation				
concatenate 36 with location map (X,Y)				
conv	37-38	1	258	64
conv without norm	39-40	1	64	32
conv without norm, relu	41	1	32	16
cluster segmentation	42	1	16	1
Initial Depth Inference				
concatenate 42 with 36				
residual	42-45	1	257	128
conv	46-47	1	128	32
conv without norm	48	1	32	1
42 as masks on 48	49	1	1	1
42 as masks on 36	50	1	256	256
Variance Refinement				
concatenate 49,36,50				
conv	52-55	1	513	16
conv without norm, relu	56	1	16	1
add 56 with 49				
Variance Refinement				
concatenate 48,36,50				
conv	59-62	1	513	16
conv without norm, relu	63	1	16	1
add 48	64	1	2	1
add 64 and 58, divide by 2				

References

[1] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *Advances in Neural Information Processing Systems*, pages 730–738, 2016. 1

[2] B. De Brabandere, D. Neven, and L. Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017. 2, 4

[3] D. Eigen, C. Puhrsch, and R. 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. 1

[4] D. A. Forsyth and J. Ponce. *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002. 1

[5] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012. 1, 2

[6] J. J. Gibson. The perception of the visual world. 1950.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE con-*

ference on computer vision and pattern recognition, pages 770–778, 2016. 3

[8] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 239–248. IEEE, 2016. 5

[9] T.-s. Lee and B. R. Potetz. Scaling laws in natural scenes and the inference of 3d shape. In *Advances in Neural Information Processing Systems*, pages 1089–1096, 2006. 1

[10] B. 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*, pages 1253–1260. IEEE, 2010. 1

[11] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proc. R. Soc. Lond. B*, 204(1156):301–328, 1979. 1

[12] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 2

[13] D. Novotny, S. Albanie, D. Larlus, A. Vedaldi, A. Nagrani, S. Albanie, A. Zisserman, T.-H. Oh, R. Jaroensri, C. Kim, et al. Semi-convolutional operators for instance segmentation. 2, 4

[14] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5506–5514, 2016. 1

[15] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in neural information processing systems*, pages 1161–1168, 2006. 1

[16] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *Proceedings of the International Conference on Computer Vision - Workshop on 3D Representation and Recognition*, 2011. 2

[17] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille. Towards unified depth and semantic prediction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2800–2809, 2015. 1

[18] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of CVPR*, 2017. 1

[19] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017. 3

[20] W. Zhuo, M. Salzmann, X. He, and M. Liu. Indoor scene structure analysis for single image depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 614–622, 2015. 1