Depth is just what we need: Monocular 3D object detection using depth estimation

Chang-Sung Sung[1], Chia-Ming Chang[2], and Shih-Chun Lin[2]

[1]*Data Science Degree Program, National Taiwan University*
[2]*Graduate Institute of Communication Engineering, National Taiwan University*

June 27, 2021

Abstract

In this project, we explore the applicability to detect 3D bounding box from only one color image. We review several previous works regarding depth estimation and 3D object detection for the background knowledge. Our proposed architecture involve two parts. First, we use a monocular depth model to estimate a depth map from a color image. Second, we feed the depth map and the color image to a monocular 3D object detection model to detect 3D bounding boxes.

Contents

1 Introduction

As the 5G and autonomous driving technology develop, understanding the 3D information from the world becomes an influential issue. One task of the issue is monocular 3D object detection that detect 3D bounding box with orientation and position information with only one frame. However, this task is an ill-posed problem due to the uncertainty of the depth and scale information. In this work, we combine depth estimation to make up for the shortcomings of monocular vision.

2 Architecture Overview

Our model involves two parts: monocular depth estimation and 3D object detection. For the depth estimation model, the input is a color image and the output is a depth map. After depth estimation, the color image and the depth map will be feed into the 3D object detection to predict the bounding box.

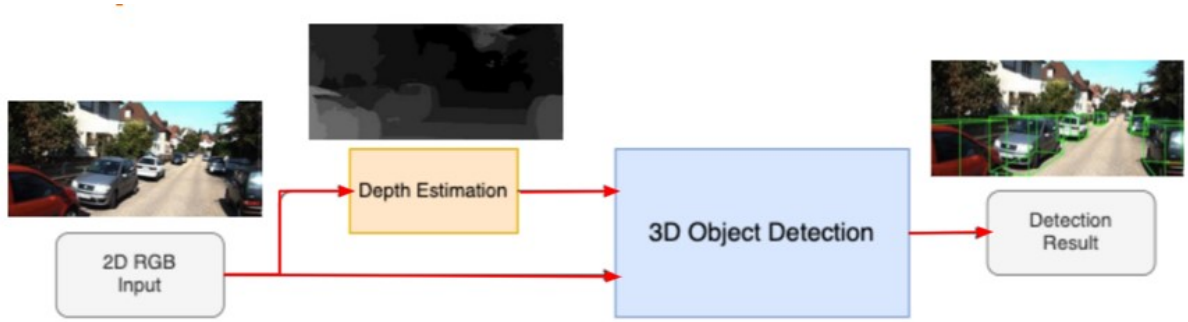


Figure 1: Architecture.

3 Monocular Depth Estimation

Our depth model refers to the Monodepth2 model [1]. The model includes a U-net [2] as a depth network and a pose network to fit the variation between two camera poses. For the depth network, the input is a color image I_t and the output is a depth map D_t. Two successive frames are feed as input for the pose network, and the output is the relative camera pose between the two frames.

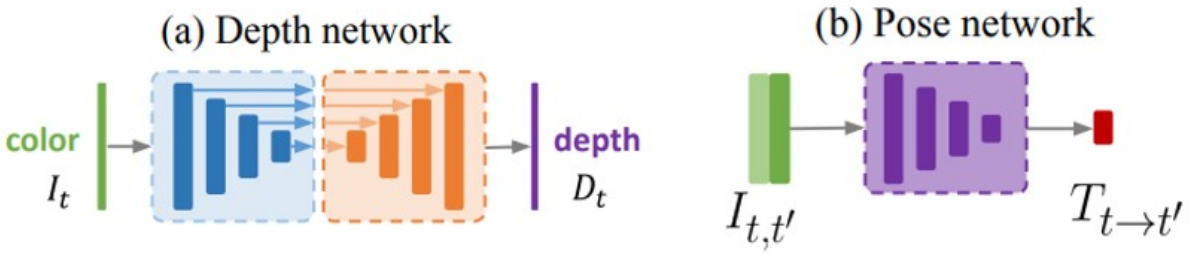


Figure 2: Monocular Depth Estimation Model.

We apply the Resnet18 pretrained model as the encoder of U-net and pose net, sharing the same model. The decode involves four prediction layers and four upsampling layers. During training, the encoder forwards four features with different resolutions to the decoder. Next, the decoder predicts four depth maps with different resolutions and then upsamples to the original input size. Using the four depth maps to calculate loss provides the scale-invariant property.

Because monocular depth estimation cannot obtain the absolute depth map, we also input $t-1$ and $t+1$ frames for calculating the reprojection loss at time t during the training process. The pose network first estimates the relative camera pose between t and t', where t' can be $t-1$ or $t+1$. Then, we can reconstruct the color image at t from the color image at t' by the depth map D_t, the relative pose $T_{t\to t'}$, and the intrinsic matrix K.

$$I_{t'\to t} = I_{t'}(projection(D_t, T_{t\to t'}, K)) \tag{1}$$

As we have the original image I_t and reconstruction image $I_{t'\to t}$, we can calculate the reprojection loss. The reprojection loss adopts the combination of structure similarity index (SSIM) [3] and L1-norm. Since we have four different scales and two pairs of (t, t'), we can obtain 8 reprojection losses for each data instance. For each scale, we calculate the per-pixel minimum reprojection loss between the two pairs $(I_t, I_{t-1\to t})$ and $(I_t, I_{t+1\to t})$, and average the losses in four different scales as the final reprojection loss L_r.

$$ReprojectionLoss(I_t, I_{t'\to t}) = \frac{\alpha}{2}(1 - SSIM(I_t, I_{t'\to t})) + (1-\alpha)L_1(I_t, I_{t'\to t}) \tag{2}$$

$$L_r = \sum^{4} \min_{t'}(ReprojectionLoss(I_t, I_{t'\to t})) \tag{3}$$

Besides the per-pixel minimum reprojection loss, we also adopt disparity smoothness loss in [4]. Strong image gradients often reflect the depth discontinuities [5], so we calculate the terms $e^{-|\partial_x I_t|}$ and $e^{-|\partial_y I_t|}$, weighted by the gradient of normalized depth d_t. Finally, we combine the per-pixel reprojection loss and the disparity smoothness loss into the overall loss.

$$L_s = |\partial_x d_t| e^{-|\partial_x I_t|} + |\partial_y d_t| e^{-|\partial_y I_t|} \tag{4}$$

$$L = w_1 L_r + w_2 L_s \tag{5}$$

4 Monocular 3D Object Detection

For the monocular 3D object detection, we propose two methods to deal with the additional depth information. Both of our methods are based on SMOKE [6]. Therefore, we'll name the two methods SMOKED1 and SMOKED2 in the following sections.

4.1 SMOKE

SMOKE is a simple single-stage monocular 3D object detection model. It consists of a pretrained feature extractor, a keypoints classification branch and a 3D box regression branch. The information flow of SMOKE is shown in Figure 3.

For the keypoint branch, the authors define the keypoint estimation network such that each object is represented by one specific keypoint. Instead of identifying the center of a 2D bounding box, the keypoint is defined as the projected 3D center of the object on the image plane. For each ground truth keypoint, its corresponding downsampled location on the feature map is computed and distributed using a Gaussian Kernel following [7]. For the 3D box regression branch, the regression head predicts the essential variables to construct the 3D bounding box for each keypoint. Similar to other monocular 3D object detection method, the 3D information is encoded as an 8-tuple $\tau = [\delta_z, \delta_{x_c}, \delta_{y_c}, \delta_h, \delta_w, \delta_l, \sin\alpha, \cos\alpha]$. For more details, please refer to the original paper.

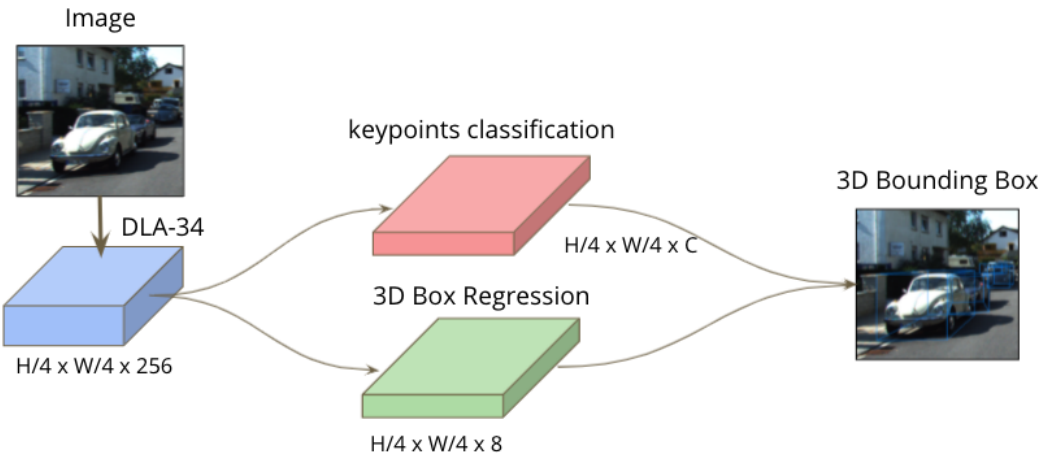


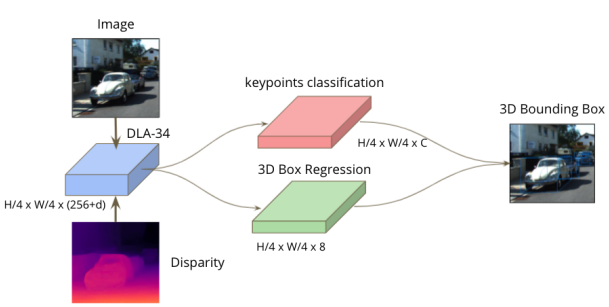
Figure 3: SMOKE

4.2 SMOKED1

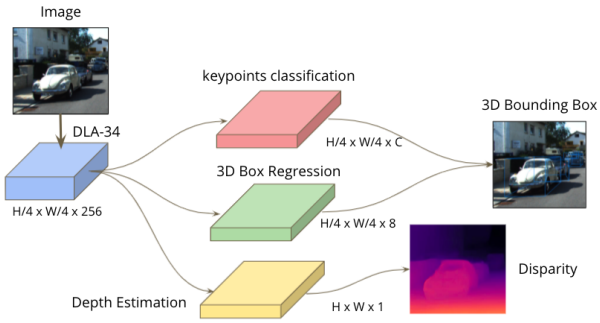
SMOKED1 is an extended version of SMOKE. As shown in Figure 4a, we simply add the depth information estimated by Monodepth2 to SMOKE by concatenating the extracted feature by DLA-34 with the downsampled depth map. We expect that with the help of the estimated depth information, keypoint classification and 3D box regression can achieve better performance.

4.3 SMOKED2

The main idea of SMOKED2 is based on knowledge distillation [8]. As shown in Figure 4b, we add an additional branch to SMOKE. The estimated depth map play the role as the teacher value of the depth estimation branch. We assume that by adding the depth estimation branch, the feature extractor will also learn some knowledge from the pretrained Monodepth2 depth estimation model. As such, the original two branches are probably able to gain more depth-related information during training in contrast to SMOKE.



(a) SMOKED1



(b) SMOKED2

Figure 4: Monocular 3D Object Detection Model

5 Experiment and Result

5.1 Experimental Settings

We use Kitti Eigen split dataset [9] for the monocular depth estimation. After removing the static frame, the dataset includes 39810 images for training and 4424 images for validation. For monocular 3D object detection, we use Kitti cars dataset. To ensure that the scene in training set would not appear in validation set. We follows [10] settings, the training set is split into 3712 training examples and 3769 validation examples.

We first train the depth model on the Kitti Eigen split dataset. Then, we utilize the depth model to predict the depth maps on the Kitti cars dataset. Thus, we obtain the depth maps for all images in the Kitti cars dataset so that We can train the object detection model with the color images and the corresponding depth maps as input.

5.2 Monocular Depth Estimation

Figure 5 shows the result of the monocular depth estimation. The quality of sharp edge can be attributed to the edge-aware smoothness loss mentioned before.

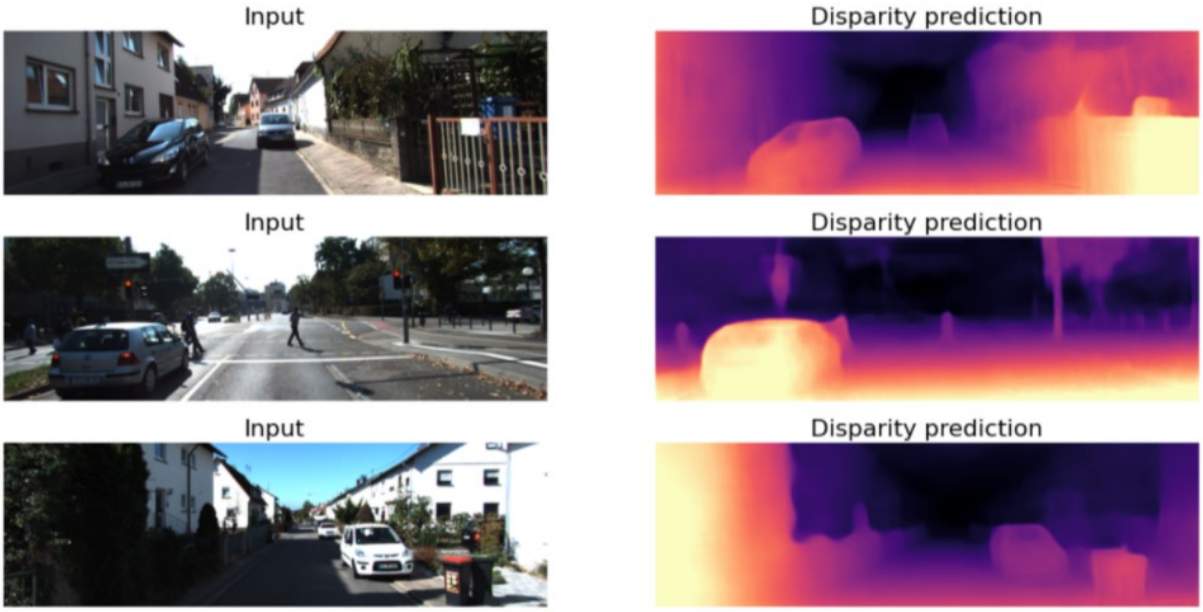


Figure 5: Depth Estimation

5.3 Monocular 3D Object Detection

Table 1 shows the comparison of the baseline model: SMOKE and the models we proposed: SMOKED1, SMOKED2. 3D object detection performance w.r.t. the car class on the official KITTI data set using the val split. Both metrics are evaluated by $AP|R_{11}$ at 0.7 IoU threshold. Quantitative Results on test video are displayed in Figure 6.

Figure 6: 3D Object Detection. Top: SMOKE. Middle: SMOKED1. Bottom: SMOKED2.

Method	KITTI car		
	Easy	Moderate	Hard
SMOKE	10.5	**8.1**	**6.8**
SMOKED1	6.2	3.7	3.7
SMOKED2	**11.1**	6.8	6.8

Table 1: Validation set performance.

6 Conclusion

In this work, we implement a monocular 3D Object Detection Model using depth estimation named SMOKED1 and SMOKED2. To extract depth map, we also formulate the depth estimation network Monodepth2 which aims to make more accurate prediction. In comparison, our SMOKED2 shows slightly better performance than SMOKE on KITTI (Easy).

7 Work Division

Monocular Depth Estimation: Shih-Chun Lin
Monocular 3D Object Detection: Chia-Ming Chang, Chang-Sung Sung

References

[1] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. October 2019.

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[3] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[4] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.

[5] Philipp Heise, Sebastian Klose, Brian Jensen, and Alois Knoll. Pm-huber: Patchmatch with huber regularization for stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2360–2367, 2013.

[6] Zechen Liu, Zizhang Wu, and Roland Tóth. Smoke: single-stage monocular 3d object detection via keypoint estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 996–997, 2020.

[7] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[9] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[10] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.