SINGLE IMAGE 3D VEHICLE POSE ESTIMATION FOR AUGMENTED REALITY

Yawen Lu[*†], *Sophia Kourian*[*†] , *Carl Salvaggio*[†], *Chenliang Xu*[‡], *and Guoyu Lu*[†]

[†]Chester Carlson Center for Imaging Science, Rochester Institute of Technology,
[‡] Department of Computer Science, University of Rochester

ABSTRACT

The intent of this paper is to introduce a novel method for 3D vehicle pose estimation, a critical component of augmented reality. The proposed method is able to recover the location of a specific object from a single image by combining pretrained reliable semantic segmentation and improved single image depth estimation. Our method exploits a novel pose estimation technique by generating new 2D images created from the projections of rotated point clouds. The rotation of the specific object is able to be predicted. Augmented objects can be shifted, rotated and scaled correctly based on the recovered orientation and localization. Through accurate vehicle pose estimation, virtual vehicles are able to be augmented accurately in place of real vehicles. The effectiveness of our method is verified by comparison with other recent pose estimation methods on the challenging KITTI 3D benchmark. Further experiments on the Cityscapes dataset also demonstrates good robustness in the method. Without requiring ground truth 3D vehicle pose labels for training, our model is able to produce competitive and robust performance in 3D vehicle pose estimation.

***Index Terms*—** Vehicle pose estimation, Augmented reality, Depth estimation, Convolutional neural network

1. INTRODUCTION

Estimation of 3D vehicle pose from a monocular image is important in virtual and augmented reality applications for robotic and autonomous driving. Previous works utilize detailed CAD models to explore their instance-level using 6 degree-of-freedom (DoF) pose estimation in 3D space [1] [2] [3] [4]. These methods, however, require known shape information and/or CAD models, which is typically unknown for most objects in real scenes. Recent end-to-end learning-based methods are able to directly regress a 3D bounding box by combining pretrained 3D proposal networks and multiview fusion modules (depth sensors) such as LiDAR and RGB-D cameras. Image-based methods [5] [6] usually generate 3D detection from successful pretrained 2D detection like approaches like RCNN [7] and Fast RCNN [8], however, more often than not the estimated 3D positions do not represent an accurate orientation of objects. [9] proposes to transform a point cloud of the scene into a volumetric grid, and utilizes 3D fully convolutional neural network to generate the object proposals and classification. The computational cost for those methods is usually quite high due to the complicated operations of 3D convolutions and the large 3D search space. Deep sliding shapes [10] takes a 3D volumetric scene from a RGB-D camera as input to learn a 3D convolutional neural network to generate 3D object bounding boxes. Gonzalez *et al.* [11] fuse high-resolution LiDAR and RGB images for 2D detection. These methods are often expensive due to the multiple sensors required and/or the requirement for manually labeling the 3D data. For large-scale data sets such as Cityscapes [12] and SUN RGB-D [13], these official annotations/labels for precise 6 DoF poses are not available.

Fig. 1: We propose the estimate of a pose representation for specific objects. The proposed method is able to place augmented object into a real scene. Here is an example of the virtual object placement on the content of an input RGB image.

To address the limitations presented in these past works, we propose a framework to estimate accurate 3D vehicle pose from a single image and to demonstrate its excellent performance in the application of object placement in augmented reality (AR). The proposed framework is able to establish the rotation and translation of the vehicles simultaneously. The network of 2D detectors and the corresponding segmented masks for objects are obtained using a recent object detection network, Mask-RCNN [14] as shown in Fig. 4. The 3D point cloud is obtained from the intermediate output from the 2D proposal and the improved single image depth estimation network by constraining the 3D boundary tightly to fit in the 2D detection window. To address the great challenge of improving the accuracy of the inferred orientation, we have designed a novel method to rotate the object in point cloud continu-

* indicates equal contribution

ously to capture the final predicted pose and orientation.

Experimental results show that the proposed method outperforms the recent methods in 3D vehicle pose estimation. The quantitative results demonstrate the outstanding visual effect that may be obtained when applied in augmented reality. Our main contributions can be described as 1) A novel method to estimate the vehicle pose in 3D space from a monocular single image. Without the requirement for 3D ground truth labels and multiple sensors, we are able to achieve competitive performance using just a single image compared with other recent models; 2) The proposed method can be directly utilized for object placement in augmented reality application (Fig. 1). We demonstrate the effectiveness and excellent robustness in unseen scenarios, as we do not train on a specific data set like other end-to-end training schemes do. The method can be applied to other objects' pose estimation and augmented reality tasks as well.

2. VEHICLE POSE ESTIMATION FRAMEWORK

We propose a fusion network to effectively combine a 2D detector and predicted depth map to estimate an object's 3D pose. A pose estimation method is designed to obtain accurate object position and orientation from a rotated point cloud.

2.1. Problem Formulation

As depicted in Fig. 3, a vehicle's pose estimation can be parameterized by the 3D location (X, Y, Z) as well as the angles of roll, pitch and yaw. For an open street scenario, it is assumed that there is no significant roll and pitch, and we only consider the yaw angle θ, as the only significant parameter on the flat street surface. This assumption is strongly supported by the fact that KITTI only provides ground truth for the yaw rotation.

2.2. Point Cloud Generation

In order to obtain an accurate 3D expression from a monocular 2D image, an improved method illustrated as Fig. 2 is utilized to infer a disparity map from single image based on [15]. In [15], Godard *et al.* presented a spatial transformer

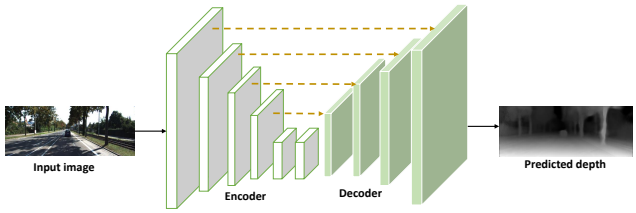

Fig. 2: Overview of our depth estimation network architecture. We employ an encoder-decoder structure with skip connections. The encoder is based on ResNet-50 to extract feature representations from the input RGB image. The decoder part is composed of a series of up-sampling blocks to recover the original spatial size as input.

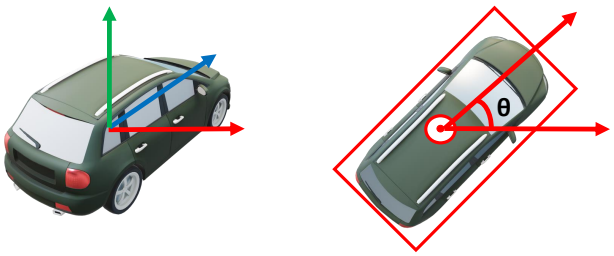

Fig. 3: Explanation of 3D pose (position and orientation). Left: Car position expressed by (X, Y, Z) in meter. Right: Car orientation. Orientation of the car is expressed by the angle from the direction of its headlights to the horizontal direction. Our method is able to get competitive results on both 3D pose estimation.

network to yield a reconstruction cost function. By including a left-right disparity consistency term, their unsupervised method achieved a higher accuracy than the contemporary supervised methods during inference. Our improved method further extend its network to a CNN block. More specifically, we constrain the reconstructed left image $\widetilde{I}^l$ to be the same as the original input left image I^l, and the reconstructed right image $\widetilde{I}^r$ to be the same as the original right input image I^r. After extracting feature representations from the encoder, the decoder is able to up-sample the features to output the predicted disparity map by moving pixels and obtaining correspondences along the epipolar line. The final predicted disparity map $\widetilde{d}$ can be inferred from the following relationship:

$$\widetilde{I} = I(x + \widetilde{d}) \tag{1}$$

where $\widetilde{I}$ is the reconstructed left or right image, I is the original input image, and $\widetilde{d}$ represents the predicted disparity map.

Given the predicted disparity map $\widetilde{d}$, the Z coordinate can be obtained by projecting 2D pixels of the raw image as follows:

$$Z = \frac{f * b}{\widetilde{d}} \tag{2}$$

where f is the focal length of the camera and b is the baseline.

With the depth determined, one can easily derive the X and Y world coordinates from the pixel positions and the camera's intrinsic parameters as

$$X = \frac{Z * (u - c_x)}{f} \tag{3}$$

$$Y = \frac{Z * (v - c_y)}{f} \tag{4}$$

where c_x and c_y are the principal point coordinates in horizontal and vertical directions of the image plane, and (u, v) is the 2D image coordinate.

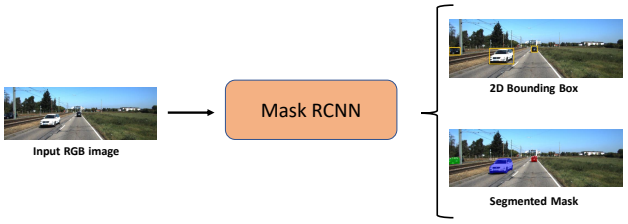

Fig. 4: Illustration of Mask-RCNN. Based on Faster R-CNN that uses a CNN to extract image features and uses CNN region proposal network to generate region of interests (RoIs), it further adds two more convolution layers to construct the object mask. Our framework uses pretrained Mask-RCNN to produce 2D regression.

2.3. Vehicle Pose Estimation

In order to accurately embed virtual objects in a scene, we must understand the vehicle pose relative to the camera; both its position and orientation. In the target data set, the object is defined to have a rotation of $+\pi/2$ degrees if its headlights are parallel to the camera's optical axis and facing the camera. Vehicles that are perfectly perpendicular in either direction relative to the camera's optical axis are defined to have 0 degrees of rotation. Inference of the orientation in our model is performed by minimizing the vehicle's width or maximizing its length based on the yaw rotation. As depicted in Fig. 3, the vehicle orientation can be defined by

$$\theta = \begin{cases} argmin_\theta \mathbf{W}, \ if \ \pi/4 < |\theta| < 3\pi/4, \\ argmax_\theta \mathbf{L}, \ if \ |\theta| < \pi/4 \ or \ 3\pi/4 < |\theta| < \pi \end{cases} \quad (5)$$

where θ represents the yaw orientation angle, and W and L are the vehicle width and length, respectively. We know that by incrementally rotating the cropped point cloud, projecting the point cloud back to the 2D image plane, and comparing the dimension with the minimum width or maximum length, we are able to find the most accurate rotation for a specific object along yaw direction. The rotated point cloud can be expressed by

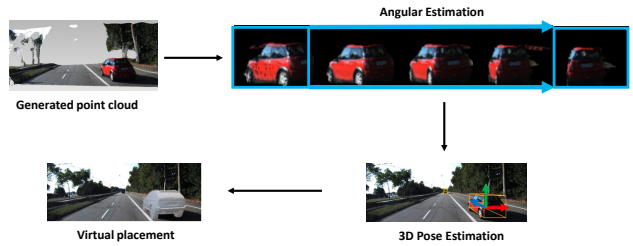

Fig. 5: Framework of our 3D pose estimation. Orientation inference is performed by rotating the generated point cloud to minimize the width or maximize the length along yaw direction. With the proposed method, 3D vehicle pose estimation and augmented scene with virtual object can be realized.

$$P_{rot} = \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix} P_{ori} \quad (6)$$

where P_{ori} and P_{rot} indicate the point cloud before and after rotation. To easily display the estimated 3D orientation in the original 2D image $I(x, y)$, we project the 3D point cloud back to the 2D image plane $[x, y, 1]^T$. This is achieved by multiplying each 3D point $\mathbf{X} = [X, Y, Z, 1]^T$ by the camera's intrinsic matrix. The 2D image pixel coordinate $[x, y, 1]$ can be expressed as

$$[x, y, 1]^T = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} [R \ t][X, Y, Z, 1]^T \quad (7)$$

where R and t are the extrinsic rotation and translation matrices, respectively, (f_x, f_y) are the focal lengths, and (c_x, c_y) are the principal points in the x, y coordinate system.

$$[x, y, 1]^T = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} P_{rot} \quad (8)$$

2.4. Virtual Object Initialization

To place virtual objects in the original scene, we need to ensure that the ground plane of the virtual object be fitted falls "flat" on the ground plane depicted in the target image. This can be realized using least squares. The virtual object also needs to be rescaled to fit consistently with the scale of the target scene. Additionally, the virtual object must be rotated according to our predicted pose angle and translated based on our predicted position. The completed process can be expressed as

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} cos(\theta) & 0 & sin(\theta) & X_{shift} \\ 0 & 1 & 0 & Y_{shift} \\ -sin(\theta) & 0 & cos(\theta) & Z_{shift} \end{bmatrix} LS \begin{pmatrix} X_{init} \\ Y_{init} \\ Z_{init} \\ 1 \end{pmatrix} \quad (9)$$

where $(X_{init}, Y_{init}, Z_{init})$ is the initial coordinate of the virtual object. (X, Y, Z) is final 3D coordinate after transformation. LS stands for the Least Squares method used for plane fitting.

3. EXPERIMENTAL RESULTS

3.1. Implementation Details

In this work, we conducted our experiment using both the KITTI and Cityscapes data sets. First, we trained our learning framework for single image depth estimation on the raw KITTI data with a batch size of 8 and the Adam optimizer [16]. ResNet-50 [17] was used as the encoder structure and

Fig. 6: Explanation of our proposed pose estimation method. Given an original single image with masked object, the geometric method is able to rotate the indicated object in the point cloud constantly along the azimuth direction to obtain its maximum length or minimum width (marked in red box). The estimated orientation is the angle it has rotated from its initial pose.

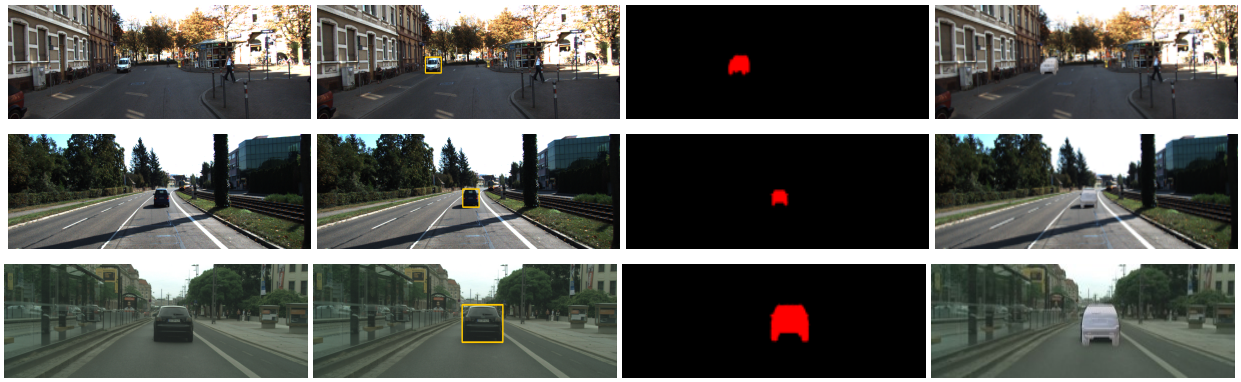

Fig. 7: Qualitative illustration of the 2D detection boxes, the estimated virtual object masks and augmented scene with on given examples (First two rows from KITTI and third row from Cityscapes). First column: Original input image; Second column: 2D detection boxes; Third column: Estimated virtual object masks; Fourth column: Augmented scenes with virtual object.

the Rectified Linear Unit (ReLU) was selected as the activation functions for each convolutional layer. The decoder consisted of a series of upsampling layers to output the depth image the same as the original input. The 3D vehicle pose was able to be estimated directly without training requiring 3D labels. With the predicted disparity map and the pretrained Mask-RCNN network, our method is able to be evaluated using the KITTI benchmark for 3D vehicle pose estimation, where we split the provided images into training and testing set following the guidance provided by [6].

We compared our framework with other state-of-the-art 3D vehicle pose estimation methods that utilized a single image [5] [6]. The average orientation similarity (AOS) and mean orientation error were evaluated on the segregated testing set.

Method	Orientation error (rad)	AOS (%)	
		Visible	Occluded
3DOP [6]	0.580	91.58	85.80
Mono3D [5]	0.558	91.90	86.28
Ours	0.243	94.20	86.37

Table 1: Mean errors in orientation estimation and average orientation similarity (AOS) compared with other methods (Mono3D [5] and 3DOP [6]).

3.2. Comparison with other Methods

Fig. 6 explains the process that our method uses to estimate the vehicle orientation from the azimuth direction given a single monocular image as the input. It can be seen that our approach is able to recover an accurate orientation from the rotated point cloud approach. Table 1 demonstrates the pose estimation and AOS results. Our approach demonstrates a better capability to estimate the vehicle orientation compared with Mono3D [5] and 3DOP [6], all without using any 3D training labels as required by these referenced methods. For orientation, our method can achieve a 0.243 radian mean error, which is 0.315 and 0.337 radians less than the supervised methods, Mono3D [5] and 3DOP [6]. Fig. 7 displays the visual results representing the generation of 2D and 3D bounding box labels for vehicles and augmented scene results on different standard data sets (KITTI and Cityscapes). These results demonstrate the practicality and excellent robustness of our method on different and unseen scenarios.

4. CONCLUSION

We have presented a method that estimates the 3D object pose and embeds the virtual object into a real, target scene. By formulating the 3D vehicle pose estimation task as a learning-based geometric problem, we make use of the 2D seman-

tic information and depth constraints simultaneously. Unlike previous learning-based networks, our model does not require 3D labels to train. The proposed framework has demonstrated its efficiency and effectiveness compared with other methods on both the standard KITTI and Cityscapes data set. Our framework is easily implemented for real applications.

5. REFERENCES

[1] Bugra Tekin, Sudipta N Sinha, and Pascal Fua, "Real-time seamless single shot 6d object pose prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 292–301.

[2] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[3] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, et al., "Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3364–3372.

[4] Francisco Massa, Bryan C Russell, and Mathieu Aubry, "Deep exemplar 2d-3d detection by adapting from real to rendered views," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 6024–6033.

[5] 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*, 2016, pp. 2147–2156.

[6] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Huimin Ma, Sanja Fidler, and Raquel Urtasun, "3d object proposals using stereo imagery for accurate object class detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1259–1272, 2017.

[7] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[8] Ross Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[9] Bo Li, "3d fully convolutional network for vehicle detection in point cloud," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1513–1518.

[10] Shuran Song and Jianxiong Xiao, "Deep sliding shapes for amodal 3d object detection in rgb-d images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 808–816.

[11] Alejandro González, David Vázquez, Antonio M López, and Jaume Amores, "On-board object detection: Multicue, multimodal, and multiview random forest of local experts," *IEEE transactions on cybernetics*, vol. 47, no. 11, pp. 3980–3990, 2016.

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[13] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 567–576.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[15] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017.

[16] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.