Exploring Intermediate Representation for Monocular Vehicle Pose Estimation

Shichao Li[1],   Zengqiang Yan[1],   Hongyang Li[2],   Kwang-Ting Cheng[1]

[1]The Hong Kong University of Science and Technology,   [2]SenseTime Research

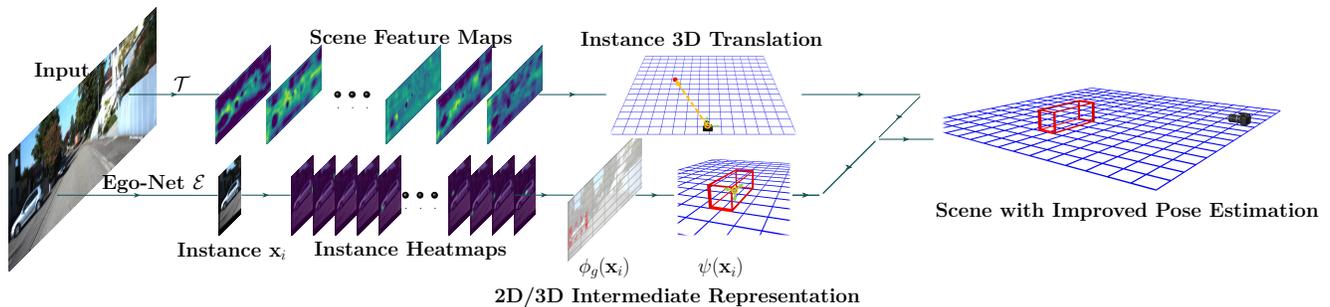slicd@cse.ust.hk,   timcheng@ust.hk

Figure 1: A novel deep model, Ego-Net, is proposed for accurate vehicle orientation estimation in the camera coordinate system via extracting a set of explicit 2D/3D geometrical representations. A 3D vehicle detection system (upper branch) can enjoy improved orientation estimation accuracy by replacing its pose estimation module with Ego-Net (lower branch).

Abstract

*We present a new learning-based framework to recover vehicle pose in SO(3) from a single RGB image. In contrast to previous works that map local appearance to observation angles, we explore a progressive approach by extracting meaningful Intermediate Geometrical Representations (IGRs) to estimate **egocentric** vehicle orientation. This approach features a deep model that transforms perceived intensities to IGRs, which are mapped to a 3D representation encoding object orientation in the camera coordinate system. Core problems are **what** IGRs to use and **how** to learn them more effectively. We answer the former question by designing IGRs based on an interpolated cuboid that derives from primitive 3D annotation readily. The latter question motivates us to incorporate geometry knowledge with a new loss function based on a projective invariant. This loss function allows unlabeled data to be used in the training stage to improve representation learning. Without additional labels, our system outperforms previous monocular RGB-based methods for joint vehicle detection and pose estimation on the KITTI benchmark, achieving performance even comparable to stereo methods. Code and pre-trained models are available at this HTTPS URL[1].*

1. Introduction

> *"The usefulness of a representation depends upon how well suited it is to the purpose for which it is used".* –David Marr [43]

Understanding 3D properties of the surrounding world is critical for vision-based autonomous driving and traffic surveillance systems [15]. Accurate 3D vehicle orientation estimation (VOE) can imply a driver's intent of travel direction, help predicting its position a moment later and help identifying anomalous behaviors. To pursue high accuracy, recent studies have exploited active sensors to obtain point clouds [63, 69, 47] or the use of stereo cameras [32, 55, 46]. The former enjoys the convenience of directly working with 3D data and the latter benefit from multi-view geometry. As downsides, LiDAR sensors cost more than commodity RGB cameras and stereo systems incur extra complexity. Pushing the limit of a single-view RGB-based method is not only cost-effective, but can also complement other configurations through sensor fusion.

However, estimating 3D parameters from a single RGB image is highly challenging due to the lack of depth information and other geometrical cues. Thanks to the strong representation learning power brought by recent advances in deep architectures, state-of-the-art (SOTA) RGB-based methods can regress 3D parameters from visual appearance in an end-to-end manner [45, 3, 52]. In this paradigm, paired images and 3D annotations are specified as inputs and learning targets respectively for supervising a deep

---

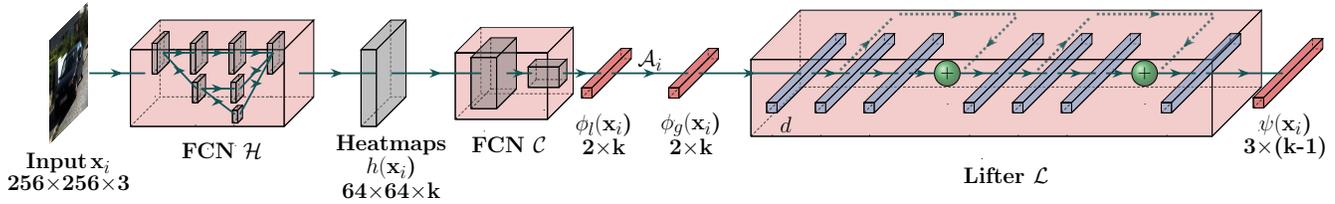[1]https://github.com/Nicholasli1995/EgoNet

Figure 2: Detailed architecture of Ego-Net. Feature maps are first computed with a fully convolution model $\mathcal{H}$ from a detected instance. Heatmaps representing the projection of a 3D cuboid are regressed and mapped to local coordinates with several strided convolution layers. The local coordinates are transformed to screen coordinates $\phi_g(\mathbf{x}_i)$ and mapped to a 3D cuboid representation $\psi(\mathbf{x}_i)$, whose orientation directly represent egocentric pose in the camera coordinate system. k=33 when q=2 as in Sec. 3.

model. No intermediate representation is designed in such end-to-end solutions, which need a large amount of training pairs to approximate the highly non-linear mapping from the pixel space to 3D geometrical quantities.

In contrast to the aforementioned methods, we take a different approach and ask *can a neural network extract explicit geometrical quantities from RGB images and use them for monocular vehicle pose estimation*? Our inspiration is from the representational framework for vision [43], which first extracts geometrically meaningful $2\frac{1}{2}$-D sketch from perceived intensities and converts them to 3D representations. We are also inspired by recent advances in two-stage 3D human pose estimation [36]. Previous approaches [45, 3, 52] specify the start (pixels) and the end (3D pose) of such a process while ignoring the learning of any intermediate representation. Apart from the motivation from the framework [43], geometrical representations have two other benefits: (1) they are easier for human to comprehend than the dense feature maps when debugging safety-critical applications, (2) the mapping from them to 3D pose is easy to learn as a few layers would suffice in our experiments.

Towards this end, we propose a novel VOE approach featuring learning Intermediate Geometrical Representations (IGRs). We first explicitly define a set of IGRs based on existing ground truth without any extra manual annotation. We then design a novel deep model, Ego-Net, which first regresses IGRs and then maps them to the 3D space, achieving SOTA performance. Apart from the high accuracy, we propose a novel loss function for self-supervising the learning of IGRs based on prior knowledge of projective invariants. With a hybrid source of supervision, our model can benefit from unlabeled data for enhanced generalization. Our contributions can be summarized in the following:

- To our best knowledge, our proposed learning-based model, Ego-Net, is the first one which regresses intermediate geometrical representations for *direct* estimation of egocentric vehicle pose.

- Our novel loss function based on the preservation of projective invariants introduces self-supervision for

the learning of such representations.

- Our approach does not require additional labels, and outperforms previous monocular RGB-based VOE methods. Our monocular system achieves performance comparable to some stereo methods for joint vehicle detection and pose estimation on KITTI.

- Ego-Net, a plug-and-play module, can be incorporated into any existing 3D object detection models for improved VOE. Equipped with Ego-Net, a 3D detection system can achieve high 3D location and orientation estimation performance simultaneously, while previous works [3, 13, 4] usually fail to achieve both.

After discussing relevant works in Sec. 2, we detail the design of Ego-Net in Sec. 3. In Sec. 4 we demonstrate how unlabeled data can be utilized to learn IGRs with prior geometry knowledge. Sec. 5 presents experimental results.

2. Related Works

Deep learning-based 3D object detection aims to learn a mapping from sensor input to instance attributes such as 3D orientation and 2D/3D location. Depending on the sensor modality, recent works can be categorized into LiDAR-based approaches [69, 28, 49, 65, 66, 50, 44], RGB-based methods [45, 60, 48, 52, 31, 5], and hybrid ones [61, 57]. Our method falls into the second category, which does not require expensive devices to acquire point clouds. Compared to stereo approaches [32, 31, 62, 11] that need two RGB cameras, our system only takes a *single view* input. For convenience in implementation, many previous works [27, 3, 13, 40, 41, 51] employ a multi-task architecture composed of a backbone for feature extraction and 3D parameter estimation heads. Each head shares similar design and VOE is treated as a sub-task with equal importance as other tasks such as dimension estimation. In contrast, we present a dedicated module for this task which outperforms previous ones.

Pose estimation in 3D object detection systems targets at recovering instance orientation. We focus on the works

that use RGB images and classify them into *learning-based* methods and geometrical *optimization-based* ones.

Learning-based methods learn a function from pixels to orientation. Conventionally, visual features were extracted and fed to a model for discrete pose classification (DPC) or continuous regression. Early works [23, 59] utilized hand-crafted features [14] and boosted trees for DPC. Recent works replace the feature extraction stage with pre-designed deep models. ANN [64] and DAVE [67, 68] classify instance feature maps extracted by CNN into discrete bins. To deal with images containing multiple instances, Fast-RCNN-like architectures were employed in [9, 2, 10, 22, 24] where region-of-interst (ROI) features were used to represent instance appearance and a classification head gives pose prediction. Deep3DBox [45] proposed *Multi-Bin* loss for joint pose classification and residual regression. Wasserstein loss was promoted in [39] for DPC. LSTM was employed in [21] to refine vehicle pose using temporal information. Our work is also a learning-based approach but possesses key differences compared to previous works. Our approach promotes learning explicit IGRs while previous works do not. With IGRs, later part of our model can directly reason in the geometrical space. Such design enables our model to directly estimate global (egocentric) pose in the camera coordinate system while previous works can only estimate relative (allocentric) pose and require a second step to obtain the global pose.

Optimization-based methods first extract 2D evidence from RGB images and fit a 3D model to the evidence for pose estimation. Deepmanta [6] detected pre-annotated semantic object parts and used Epnp [29] for pose optimization. Mono3D++ [20] introduced task priors into a joint energy function and conducted optimization using the Ceres solver [1]. Compared to these works, our approach does not need the construction of a 3D shape model neither any extra manual labeling. Such labeling was required in [6] to train a model for detecting image evidence. Neither does our approach need initialization and iterative optimization for inference. In fact, our approach can serve as the initialization step and be integrated with these works.

Self-supervision from geometry knowledge can be incorporated into deep systems to regularize representation learning. Epipolar geometry was used in [25] to generate supervisory signal for a fully convolutional network. Loop consistency [7] and projective consistency [18] were employed for 3D human pose estimation. Flow fields and disparity maps were extracted in [16] for self-supervised video representation learning. We propose a new self-supervision mechanism by utilizing projective invariants to supervise our intermediate representations. This approach can utilize unlabeled images during training yet does not require multi-view setting as [25]. It is also camera-agnostic and does not require camera intrinsics as [18].

3. Egocentric 3D Vehicle Pose Estimation

Ego-Net $\mathcal{E}$ estimates 3D orientation from cropped instance appearance. It can be combined with a 3D location estimation module $\mathcal{T}$ to form a 3D object detection model $\mathcal{M} = (\mathcal{E}, \mathcal{T})$ as shown in Fig. 1. We detail $\mathcal{E}$ in the following and some examples of composing $\mathcal{M}$ are in our supplementary material.

3.1. Single-step Progressive Mapping

For visual perception and traffic surveillance systems, expressing the poses of multiple objects in the *same* coordinate system is favorable and convenient for downstream planing and decision-making. Previous works [45, 27, 3, 13] achieved this by adopting the computational graph in Eq. 1. A CNN-based model $\mathcal{N}$ is used to map local instance appearance $\mathbf{x}_i$ to allocentric pose, i.e., 3D orientation in the object coordinate system $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$, which is later converted to the egocentric pose, i.e., orientation in the camera coordinate system. The difference between these two coordinates systems is shown in Fig. 3. This two-step design is a workaround since an object with the same egocentric pose $\boldsymbol{\theta}_i$ can produce different local appearance depending on its location [27] and learning the mapping from $\mathbf{x}_i$ to $\boldsymbol{\theta}_i$ is an ill-posed problem. In this two-step design, $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ needs to be estimated by another parallel module and $\mathcal{N}$ *can not recover $\boldsymbol{\theta}_i$ independently* without the assist of this module. The error of this module can propagate to the final estimation of egocentric poses. Even with this workaround, the problem remains challenging as the mapping from $\mathbf{x}_i$ to $\boldsymbol{\alpha}_i$ is highly non-linear and difficult to approximate.

$$\mathbf{x}_i \xrightarrow{\mathcal{N}} \begin{matrix} \boldsymbol{\alpha}_i \\ \{\mathbf{i}, \mathbf{j}, \mathbf{k}\} \end{matrix} \begin{matrix} \xrightarrow{convert} \\ \nearrow assist \end{matrix} \boldsymbol{\theta}_i \qquad (1)$$
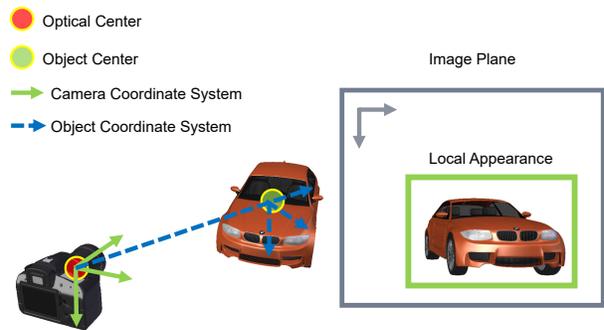

Figure 3: Local appearance cannot uniquely determine egocentric pose. Existing solutions first estimate allocentric pose in the object coordinate system (blue) and convert it to egocentric pose in the camera coordinate system (green) based on the object location.

We believe that such two-step computation is unnecessary and propose to learn a direct mapping $\mathcal{E}$ from visual

information to egocentric pose. However, instead of relying on a black box model to fit such a non-linear mapping, we assume this mapping to be a progressive process, where a series of IGRs are gradually extracted from pixels and eventually lifted to the 3D target. Specifically, we design Ego-Net as a composite function consisting of $\{\mathcal{H}, \mathcal{C}, \mathcal{L}\}$ described as follows. Given the cropped patch of one detected instance $\mathbf{x}_i$, Ego-Net predicts its egocentric pose as $\mathcal{E}(\mathbf{x}_i) = \mathcal{L}(\mathcal{A}_i(\mathcal{C}(\mathcal{H}(\mathbf{x}_i)))) = \boldsymbol{\theta}_i$. Fig. 2 depicts Ego-Net, whose computational graph is shown in Eq. 2. $\mathcal{H}$ is assumed to extract heatmaps $h(\mathbf{x}_i)$ for salient 2D object parts that are mapped by $\mathcal{C}$ to coordinates $\phi_l(\mathbf{x}_i)$ representing their local location within the patch. $\phi_l(\mathbf{x}_i)$ is converted to the global image plane coordinates $\phi_g(\mathbf{x}_i)$ with an affine transformation $\mathcal{A}_i$ parametrized with scaling and 2D translation. $\phi_g(\mathbf{x}_i)$ is further lifted into a 3D representation $\psi(\mathbf{x}_i)$ by $\mathcal{L}$. The final pose prediction derives from $\psi(\mathbf{x}_i)$.

$$\mathbf{x}_i \xrightarrow{\mathcal{H}} h(\mathbf{x}_i) \xrightarrow{\mathcal{C}} \phi_l(\mathbf{x}_i) \xrightarrow{\mathcal{A}_i} \phi_g(\mathbf{x}_i) \xrightarrow{\mathcal{L}} \psi(\mathbf{x}_i) \rightarrow \boldsymbol{\theta}_i \quad (2)$$

3.2. Labor-free Intermediate Representations

The abstraction in Eq. 2 does not mathematically define what geometrical quantity to use as IGRs, here we define them based on three considerations:

Availability: The IGRs should be easily derived from existing ground truth annotations with none or minimum extra manual effort. While using CAD models such as the wireframe model in [6] and 3D voxel pattern in [59] can provide high level-of-detail, such representations require extensive manual effort and are not scalable. A new driving scenario may requires collecting a new training set, and can incur significant burden if the IGRs cannot be readily obtained. In addition, no evidence has shown that detailed object shapes are necessary for orientation estimation.

Discriminative: The IGRs should be indicative for 3D pose estimation, so that it can serve as a good bridge between visual appearance input and the geometrical target.

Transparency: The IGRs should be easy to understand, which makes them debugging-friendly and trustworthy for applications such as autonomous driving. While network visualization techniques [53, 35, 34] can be used to understand learned representations, designing IGRs with explicit meaning offers more convenience in practice.

With the above considerations, we define the 3D representation $\psi(\mathbf{x}_i)$ as sparse 3D point cloud representing an interpolated cuboid. Autonomous driving datasets such as KITTI [17] usually label instance 3D bounding boxes from captured point clouds where an instance $\mathbf{x}_i$ is associated with its centroid location in the camera coordinate system $\mathbf{t}_i = [t_x, t_y, t_z]$, size $[h_i, w_i, l_i]$, and its egocentric pose $\boldsymbol{\theta}_i$. As shown in Fig. 4, denote the 12 lines enclosing the vehicle as $\{\mathbf{l}_j\}_{j=1}^{12}$, where each line is represented by two endpoints (**s**tart and **e**nd) as $\mathbf{l}_j = [\mathbf{p}_j^s; \mathbf{p}_j^e]$. $\mathbf{p}_j^v$ (v is s or e) is a
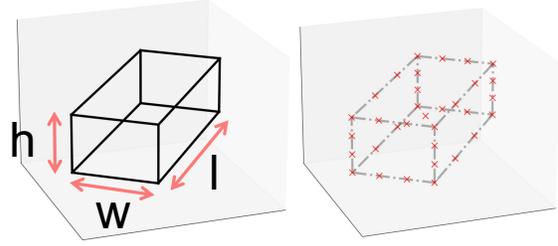

Figure 4: An instance $\mathbf{x}_i$ with primitive 3D bounding box annotation (left) and its 3D point cloud representation $\psi(\mathbf{x}_i)$ (right) in the camera coordinate system as the collection of red crosses.

3-vector (X_j^v, Y_j^v, Z_j^v) representing the point's location in the camera coordinate system. As a complexity-controlling parameter, q more points are derived from each line with a pre-defined interpolation matrix $B_{q \times 2}$ as

$$\begin{bmatrix} \mathbf{p}_j^1 \\ \mathbf{p}_j^2 \\ \dots \\ \mathbf{p}_j^q \end{bmatrix} = B_{q \times 2} \begin{bmatrix} \mathbf{p}_j^s \\ \mathbf{p}_j^e \end{bmatrix} = \begin{bmatrix} \beta_1 & 1 - \beta_1 \\ \beta_2 & 1 - \beta_2 \\ \dots & \dots \\ \beta_q & 1 - \beta_q \end{bmatrix} \begin{bmatrix} \mathbf{p}_j^s \\ \mathbf{p}_j^e \end{bmatrix}. \quad (3)$$

The 8 endpoints, the instance's centroid, and the interpolated points for each of the 12 lines form a set of $9 + 12q$ points. The concatenation of these points forms a $9 + 12q$ by 3 matrix $\tau(\mathbf{x}_i)$. Since we do not need the 3D target $\psi(\mathbf{x}_i)$ to encode location, we deduct the instance translation $\mathbf{t}_i$ from $\tau(\mathbf{x}_i)$ and represent $\psi(\mathbf{x}_i)$ as a set of $8 + 12q$ points representing the shape relative to the centroid

$$\psi(\mathbf{x}_i) = \{(X_j^v - t_x, Y_j^v - t_y, Z_j^v - t_z)\} \quad (4)$$

where $v \in \{s, 1, \dots, q, e\}$ and $j \in \{1, 2, \dots, 12\}$. Larger q provides more cues for inferring pose yet increases complexity. In practice we choose $q = 2$ and the right figure of Fig. 4 shows an example with $B_{2 \times 2} = \begin{bmatrix} \frac{3}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix}$ and 2 points are interpolated for each line.

Serving as the 2D representation to be located by $\mathcal{H}$ and $\mathcal{C}$, $\phi_g(\mathbf{x}_i)$ is defined to be the projected screen coordinates of $\tau(\mathbf{x}_i)$ given camera intrinsics $\mathrm{K}_{3 \times 3}$ as

$$\phi_g(\mathbf{x}_i) = \mathrm{K}_{3 \times 3} \tau(\mathbf{x}_i). \quad (5)$$

$\phi_g(\mathbf{x}_i)$ implicitly encodes instance location on the image plane so that it is not an ill-posed problem to estimate egocentric pose from it directly. In summary, these IGRs can be computed with zero extra manual annotation, are easy to understand and contain rich information for estimating the instance orientation.

4. Learning IGRs With Geometrical Prior

Sec. 3 defines *what* IGRs to learn, here we discuss *how* to learn such representations more effectively. We propose to use geometrical prior knowledge for self-supervision and propose a mixed training strategy to take advantage of cheap unlabeled data for training.
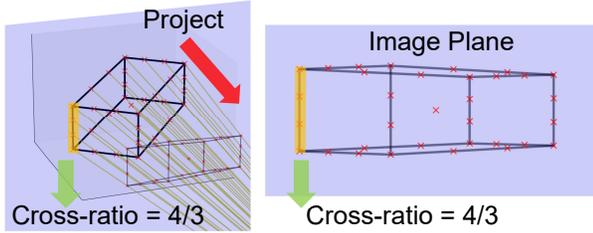

Figure 5: Perspective projection of $\tau(\mathbf{x}_i)$ (left) and the zoomed image plane (right). Since the cross-ratio of every 4 collinear points is invariant after camera projection, the same value should be expected for a line on the screen as that of the corresponding line on the 3D bounding box.

4.1. Self-supervising by Preserving Cross-ratio

The cross-ratio of 4 collinear points are invariant during perspective projection [19]. Once $\psi(\mathbf{x}_i)$ is defined with a fixed interpolation matrix $B_{q \times 2}$, this ratio is fixed and invariant after perspective projection as shown in Fig. 5. If the network $\mathcal{E}$ predicts a different value for such a ratio, one can penalize the network as supervisory signal. For every 4 points in $\phi_l(\mathbf{x}_i)$, we denote their ground truth 2D coordinates as $\phi_l(\mathbf{x}_i) = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4]$ and the ground truth cross-ratio as $cr = \frac{\|\mathbf{v}_3 - \mathbf{v}_1\| \|\mathbf{v}_4 - \mathbf{v}_2\|}{\|\mathbf{v}_3 - \mathbf{v}_2\| \|\mathbf{v}_4 - \mathbf{v}_1\|}$. Denote the predicted coordinates for such 4 points as $[\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3, \hat{\mathbf{v}}_4]$, we propose the cross-ratio loss function as

$$L_{cr} = SmoothL1(cr^2 - \frac{\|\hat{\mathbf{v}}_3 - \hat{\mathbf{v}}_1\|^2 \|\hat{\mathbf{v}}_4 - \hat{\mathbf{v}}_2\|^2}{\|\hat{\mathbf{v}}_3 - \hat{\mathbf{v}}_2\|^2 \|\hat{\mathbf{v}}_4 - \hat{\mathbf{v}}_1\|^2}), \quad (6)$$

where we choose cr^2 as the target because $\|\hat{\mathbf{v}}_p - \hat{\mathbf{v}}_q\|^2$ can be easily computed with vector inner product $< \hat{\mathbf{v}}_p - \hat{\mathbf{v}}_q, \hat{\mathbf{v}}_p - \hat{\mathbf{v}}_q >$ to avoid computing the square-root for $\|\hat{\mathbf{v}}_p - \hat{\mathbf{v}}_q\|$.

4.2. Utilizing Unlabeled Imagery

Evaluating L_{cr} does not require any ground truth information, which means an unlabeled vehicle instance can be used to supervise the learning of $\phi_l(\mathbf{x}_i)$. In addition, the preservation of cross-ratio holds irrespective of the camera intrinsics as long as a center projection model is used. This implies that imagery captured with different cameras at different locations can be used collectively and jointly. In our experiments, we crop unlabeled cars from [54] which were

Figure 6: A random training batch with ground truth $\phi_l(\mathbf{x})$ (blue) and prediction $\hat{\phi}_l(\mathbf{x})$ (red). By introducing the cross-ratio loss function, images with no labels can be employed into the training stage to regularize the learning of $\phi_l(\mathbf{x})$.

captured in a country different from those of KITTI [17] and used cameras with different intrinsics.

Ego-Net is supervised by hybrid types of loss functions and Algorithm 1 shows the forward-pass given a data batch consisting of both labeled and unlabeled instances. For the labeled ones, ground truth IGRs are available to evaluate heatmap loss L_{hm} and 2D/3D representation losses L_{2d} and L_{3d}. For the unlabeled images, only the proposed cross-ratio loss L_{cr} is evaluated as an extra supervisory signal. Fig. 6 shows a training batch containing both labeled and unlabeled cars.

Algorithm 1: Mixed training with unlabeled data

Input: Labeled data with ground truth
$D_1 = \{\mathbf{x}_i, h(\mathbf{x}_i), \phi_l(\mathbf{x}_i), \psi(\mathbf{x}_i)\}_{i=1}^{N_1}$, unlabeled image inputs $D_2 = \{\mathbf{x}_j\}_{j=1}^{N_2}$, model $\mathcal{E} = (\mathcal{H}, \mathcal{C}, \mathcal{L})$, loss functions for heatmaps, preserving cross-ratio, 2D and 3D representations as L_{hm}, L_{cr}, L_{2d} and L_{3d}

Output: Total loss L_{total}

for *i=1:N_1* **do**
    $\hat{h}(\mathbf{x}_i) = \mathcal{H}(\mathbf{x}_i)$;
    $\hat{\phi}_l(\mathbf{x}_i) = \mathcal{C}(\hat{h}(\mathbf{x}_i))$;
    $\hat{\psi}(\mathbf{x}_i) = \mathcal{L}(\mathcal{A}_i(\hat{\phi}_l(\mathbf{x}_i)))$;
end

for *j=1:N_2* **do**
    $\hat{\phi}_l(\mathbf{x}_j) = \mathcal{C}(\mathcal{H}(\mathbf{x}_j))$;
end

$L_1 = \frac{1}{N_1} \sum_{i=1}^{N_1} L_{hm}(\hat{h}(\mathbf{x}_i), h(\mathbf{x}_i))$;
$L_2 = \frac{1}{N_1} \sum_{i=1}^{N_1} L_{2d}(\hat{\phi}_l(\mathbf{x}_i), \phi_l(\mathbf{x}_i))$;
$L_3 = \frac{1}{N_1} \sum_{i=1}^{N_1} L_{3d}(\hat{\psi}(\mathbf{x}_i), \psi(\mathbf{x}_i))$;
$L_4 = \frac{1}{N_1+N_2}[\sum_{i=1}^{N_1} L_{cr}(\hat{\phi}_l(\mathbf{x}_i)) + \sum_{j=1}^{N_2} L_{cr}(\hat{\phi}_l(\mathbf{x}_j))]$;
$L_{total} = \sum_{i=1}^{4} L_i$

Method	Reference	Modality	# of Viewpoints	Easy	Moderate	Hard	Average
Mono3D [8]	CVPR' 16	RGB	Monocular	91.01	86.62	76.84	84.82
Deep3DBox [45]	CVPR' 17	RGB	Monocular	**92.90**	**88.75**	76.76	**86.14**
ML-Fusion [60]	CVPR' 18	RGB	Monocular	90.35	87.03	76.37	84.58
FQNet [37]	CVPR' 19	RGB	Monocular	92.58	88.72	76.85	86.05
GS3D [30]	CVPR' 19	RGB	Monocular	85.79	75.63	61.85	74.42
MonoPSR [26]	CVPR' 19	RGB + LiDAR	Monocular	93.29	87.45	72.26	84.33
M3D-RPN [3]	ICCV' 19	RGB	Monocular	88.38	82.81	67.08	79.42
MonoPair [12]	CVPR' 20	RGB	Monocular	91.65	86.11	76.45	84.74
Disp R-CNN [55]	CVPR' 20	RGB + LiDAR	Stereo	93.02	81.70	67.16	80.63
DSGN [11]	CVPR' 20	RGB	Stereo	95.42	86.03	78.27	86.57
D4LCN [13]	CVPR' 20	RGB	Monocular	90.01	82.08	63.98	78.69
RAR-Net [38]	ECCV' 20	RGB	Monocular	88.40	82.63	66.90	79.31
RTM3D [33]	ECCV' 20	RGB	Monocular	91.75	86.73	**77.18**	85.22
Kinematic3D [4]	ECCV' 20	RGB	Monocular	58.33	45.50	34.81	46.21
Ego-Net (Ours)	CVPR' 21	RGB	Monocular	**96.11**$_{\uparrow 3.4\%}$	**91.23**$_{\uparrow 2.8\%}$	**80.96**$_{\uparrow 4.9\%}$	**89.43**$_{\uparrow 3.8\%}$

Table 1: System-level evaluation by comparing Average Orientation Similarity (AOS) with the SOTA learning-based methods for the KITTI test set in the car category. The second-highest performance among the single-view RGB-based approaches is shown in **blue** and our improvement over it follows the ↑ signs. Without using LiDAR data [26] or temporal information [4], our system out-performs previous monocular RGB-based methods and even stereo methods [11, 55].

5. Experiments

We first introduce the evaluation benchmark, followed by an ablation study to demonstrate the effectiveness of IGRs and geometrical self-supervision. Finally we compare the overall system performance with recent SOTA works and demonstrate how Ego-Net can improve other 3D object detection systems.

5.1. Datasets and Evaluation Metrics

Datasets. We employ the KITTI object detection benchmark [17] for evaluation, which contains RGB images captured in outdoor scenes. The dataset is split into 7,481 training images and 7,518 testing images. The ground truth of the testing images is not released and thus all predictions were sent to the official server for evaluation. The training images are further split into a train set and a validation set containing 3,682 and 3,799 images respectively. Our Ego-Net is trained using only the train set and the validation set is used for hyper-parameter tuning. To obtain unlabeled car instances for training, we randomly crop images from Apol-loCar3D [54] dataset.

Evaluation Metrics. We use the standard metric, *Average Orientation Similarity* (AOS), to assess the overall system performance for joint vehicle detection and pose estimation. AOS is defined as $AOS = \frac{1}{41} \sum_{r \in \{0, 0.025,...,1\}} \max_{\tilde{r}:\tilde{r} \geq r} s(\tilde{r})$ where r is the detection recall and $s(r) \in [0,1]$ is the orientation similarity (OS) at recall level r. OS is defined as $s(r) = \frac{1}{|\mathcal{D}(r)|} \sum_{i \in \mathcal{D}(r)} \frac{1+\cos\Delta_{\theta}^{(i)}}{2} \delta_i$, where $\mathcal{D}(r)$ denotes the set of all object detections at recall rate r and $\Delta_{\theta}^{(i)}$ is the difference in angle between estimated and ground truth vehicle pose of detection i. For the AOS evaluation, vehicle in-

stances are grouped based on three difficulty levels: easy, moderate and hard. A smaller instance with more severe occlusion and truncation is considered to be more difficult.

5.2. Implementation Details

A detected instance is cropped and resized to a 256 by 256 patch as input to Ego-Net. HRNet-w48 [56, 58] is used as the fully-convolutional sub-model $\mathcal{H}$ to regress heatmaps of size 64 by 64. $L2$ loss is used for L_{hm} and L_{3d}, while $L1$ loss is used for L_{2d}. More training details are included in our supplementary material.

5.3. Ablation Study

Direct regression vs. learning IGRs To evaluate the effectiveness of our proposed IGRs, we compare the pose estimation accuracy with learned IGRs to that of a baseline which directly regresses pose angles from the instance feature maps. AOS is influenced by the recall of car detection and is upper-bounded by 2D Average Precision (AP_{2D}) of the detector [17]. To eliminate the impact of the used object detector, we compare AOS on all annotated vehicle instances in the validation set. This is equivalent to measuring AOS with $AP_{2D} = 1$ so that the orientation estimation accuracy becomes the only focus. The comparison is summarized in Tab. 2. Note learning IGRs outperforms the baseline by a significant margin. Deep3DBox [45] is another popular architecture[3] that performs direct angle regression, but utilizes a complex *MultiBin* loss function for joint classification and regression. Without designing such loss functions, our approach outperforms it with the novel design of IGRs and does not need discrete pose classification. Quali-

---

[3]https://github.com/smallcorgi/3D-Deepbox

Figure 7: Qualitative results on the KITTI validation set where the red arrows indicate the vehicle poses. More examples can be found in our supplementary material.

tative visualization of pose predictions on the validation set is shown in Fig. 7.

Method	AOS when $AP_{2D} = 100.00$		
	Easy	Moderate	Hard
B	95.22	92.14	88.37
Deep3DBox [45]	98.48	95.81	91.98
B+IGRs (Ego-Net)	$99.58_{\uparrow 1.1\%}$	$99.06_{\uparrow 3.4\%}$	$96.55_{\uparrow 5.0\%}$

Table 2: Module-level evaluation assuming perfect object detection ($AP_{2D} = 100.00$) in the KITTI validation set. B: baseline using direct regression. +IGR: adding intermediate geometric representations. Improvements compared with [45] follow the $\uparrow$ signs.

Effect of the mixed training strategy. To study whether the introduction of L_{cr} and unlabeled data can improve the learning of $\phi_g(\mathbf{x}_i)$ and result in improved accuracy of locating $\phi_g(\mathbf{x}_i)$, we re-train Ego-Net without using the cross-ratio component and unlabeled data. The quantitative comparison is shown in Tab. 3. *Percentage of Correct Keypoints* (PCK@X) gives the ratio of correctly located joints that have pixel error smaller than X times a pre-defined threshold. This threshold is different for each instance and depends on the instance size so that PCK is not influenced by car size variation. We define the threshold as one third of the bounding box height for each instance. Tab. 3 shows that the accuracy of locating screen coordinates improves after adding unlabeled data for training. Despite the unlabeled images do not have any 3D annotation and were taken in a different country with different camera intrinsics, they benefit model generalization. We believe the reason is that the introduction of these unlabeled images mitigates the appearance bias towards the training data.

Method	PCK@0.1	PCK@0.2	PCK@0.3
w/o mixed training	31.3	64.1	79.3
w/ mixed training	31.9	64.8	79.6

Table 3: Comparison of coordinate localization performance on KITTI validation set with and without using unlabeled data for mixed training. Higher PCK is better.

5.4. Comparison with SOTA works

Joint vehicle detection and pose estimation performance is measured by AOS. Tab. 1 compares the AOS of our system using Ego-Net with other SOTA approaches on KITTI test set. Among the single-view RGB-based approaches, Ego-Net outperforms all other deep models by a clear margin. Ego-Net even outperforms DSGN [11] and Disp R-CNN [55] that use stereo cameras. In addition, our approach using a single image outperforms Kinematic3D [4] which exploits temporal information using RGB video. This result indicates that vehicle pose can be reliably estimated with a single frame. In fact, our approach can also serve for per-frame initialization of temporal models.

Method	$AOS \& AP_{2D}$, $AOS \le AP_{2D}$		
	Easy	Moderate	Hard
	AP_{2D}		
M3D-RPN [3]	90.28	83.75	67.72
D4LCN [13]	92.80	84.43	67.89
Method	AOS		
M3D-RPN [3]	88.79	81.25	65.37
M3D-RPN + Ego-Net	$90.20_{\uparrow 1.6\%}$	$83.60_{\uparrow 2.9\%}$	$67.53_{\uparrow 3.3\%}$
D4LCN [13]	91.74	82.96	66.45
D4LCN + Ego-Net	$92.62_{\uparrow 1.0\%}$	$84.25_{\uparrow 1.6\%}$	$67.60_{\uparrow 1.7\%}$

Table 4: AOS evaluation on KITTI validation set. After employing Ego-Net, the vehicle pose estimation accuracy of SOTA 3D object detection systems can be improved. The space for AOS improvement is upper-bounded by AP_{2D}.

Method	AP_{BEV}		
	Easy	Moderate	Hard
ROI-10D [42]	14.04	3.69	3.56
Mono3D++ [20]	16.70	11.50	10.10
MonoDIS [52]	24.26	18.43	16.95
D4LCN [13]	31.53	22.58	17.87
D4LCN + Ego-Net (Ours)	**33.60**	**25.38**	**22.80**

Table 5: AP_{BEV} evaluated on KITTI validation set. Ego-Net can correct the erroneous pose predictions from [13] as shown in Fig. 9.
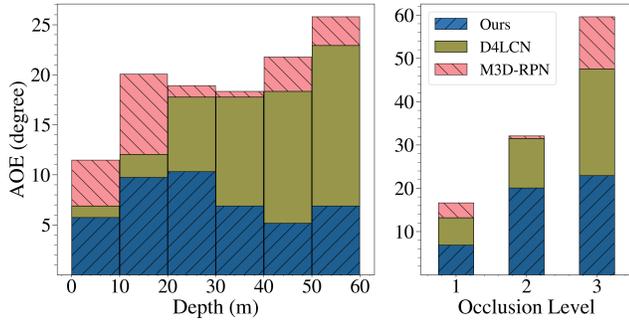
Figure 8: Average orientation error (AOE) on KITTI validation set in different depth ranges and occlusion levels. Our approach is robust to distant and partially occluded instances.

Ego-Net as a module can be used as the pose estimation branch of other 3D object detection systems. To assess if Ego-Net can help improve the pose estimation accuracy of other 3D object detection systems, compared to their original pose estimation branch, we fuse Ego-Net with some recent works with open-source implementations. We took the detected instances produced by other approaches and compared the estimated poses by Ego-Net with those produced by their original pose estimation modules. The result is summarized in Tab. 4. While AOS depends on the detection performance of these methods, fusing Ego-Net consistently improves the pose estimation accuracy of these approaches. This indicates that Ego-Net is robust despite the performance of a vehicle detector varies with different recall levels. For these detected instances (true positives) we plot the distribution of orientation estimation error versus different depth ranges and occlusion levels[4] in Fig. 8. The error in each bin is the averaged egocentric pose error for those instances that fall into it. While the performance of M3D-RPN [3] and D4LCN [13] degrades significantly for distant and occluded cars, the errors of our approach increase gracefully. We believe that explicitly learning the vehicle parts makes our model more robust to occlusion as the visible parts can provide rich information for pose estimation.

Bird's eye view (BEV) evaluation compared with D4LCN [13] is visualized in Fig. 9. The egocentric pose predictions are shown under bird's eye view and the arrows point to the heading direction of those vehicles. From the local appearance it can be hard to tell whether certain cars head towards or away from the camera, as validated by the erroneous predictions of D4LCN [13] since it regresses pose from local features. In comparison, our approach gives accurate egocentric pose predictions for these instances and others that are partially occluded. Quantitative comparison

---

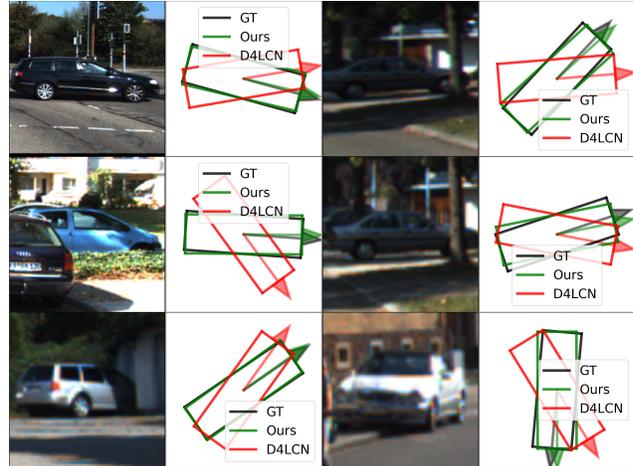[4]The occlusion levels are annotated for each instance in KITTI.

Figure 9: Detected instances in KITTI validation set along with the comparison of the predicted egocentric poses in bird's eye view.

with several 3D object detection systems on the KITTI validation set is shown in Tab. 5. Note that utilizing Ego-Net can correct wrong pose predictions especially for difficult instances, which leads to improved 3D IoU and results in significantly improved AP_{BEV}.

6. Conclusion

We propose a new deep model for estimating 3D vehicle pose in the camera coordinate system. The heatmaps of object parts are first extracted from local object appearance, which are mapped to the screen coordinates and then further lifted to a 3D object pose. A novel loss function that encourages the preservation of cross-ratio is proposed to improve the estimation accuracy of the screen coordinates. Based on the definition of intermediate representations and the cross-ratio loss, unlabeled instances can be used jointly with labeled objects to improve representation learning. Our system hits a record high for the vehicle detection and pose estimation task on KITTI among the monocular RGB-based deep models.

Several future directions remain to be explored. For example, the definition of intermediate representations could consider instance point-clouds if LiDAR data is available. More geometrical prior knowledge apart from preserving cross-ratio can be incorporated. Apart from unlabeled instances, graphics-rendered cars can also be incorporated whose 3D ground truth are known.

References

[1] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. http://ceres-solver.org.

[2] Markus Braun, Qing Rao, Yikang Wang, and Fabian Flohr. Pose-rcnn: Joint object detection and pose estimation using 3d object proposals. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1546–1551. IEEE, 2016.

[3] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *ICCV*, pages 9287–9296, 2019.

[4] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. In *In Proceeding of European Conference on Computer Vision*, Virtual, August 2020.

[5] Yingjie Cai, Buyu Li, Zeyu Jiao, Hongsheng Li, Xingyu Zeng, and Xiaogang Wang. Monocular 3d object detection with decoupled structured polygon estimation and height-guided depth estimation. In *AAAI*, pages 10478–10485, 2020.

[6] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teulière, and Thierry Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *CVPR*, pages 2040–2049, 2017.

[7] Ching-Hang Chen, Ambrish Tyagi, Amit Agrawal, Dylan Drover, Stefan Stojanov, and James M Rehg. Unsupervised 3d pose estimation with geometric self-supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5714–5724, 2019.

[8] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *CVPR*, pages 2147–2156, 2016.

[9] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In *NeurIPS*, pages 424–432, 2015.

[10] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals using stereo imagery for accurate object class detection. *TPAMI*, 40(5):1259–1272, 2017.

[11] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Dsgn: Deep stereo geometry network for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12536–12545, 2020.

[12] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12093–12102, 2020.

[13] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[14] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1532–1545, 2014.

[15] James M Ferryman, Stephen J Maybank, and Anthony D Worrall. Visual surveillance for moving vehicles. *International Journal of Computer Vision*, 37(2):187–197, 2000.

[16] Chuang Gan, Boqing Gong, Kun Liu, Hao Su, and Leonidas J Guibas. Geometry guided convolutional neural networks for self-supervised video representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5589–5597, 2018.

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012.

[18] Ikhsanul Habibie, Weipeng Xu, Dushyant Mehta, Gerard Pons-Moll, and Christian Theobalt. In the wild human pose estimation using explicit 2d features and intermediate 3d representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10905–10914, 2019.

[19] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[20] Tong He and Stefano Soatto. Mono3d++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8409–8416, 2019.

[21] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krahenbuhl, Trevor Darrell, and Fisher Yu. Joint monocular 3d vehicle detection and tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 5390–5399, 2019.

[22] Siyuan Huang, Yixin Chen, Tao Yuan, Siyuan Qi, Yixin Zhu, and Song-Chun Zhu. Perspectivenet: 3d object detection from a single rgb image via perspective points. *arXiv preprint arXiv:1912.07744*, 2019.

[23] Roman Juranek, Adam Herout, Markéta Dubská, and Pavel Zemcik. Real-time pose estimation piggybacked on object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2381–2389, 2015.

[24] Lei Ke, Shichao Li, Yanan Sun, Yu-Wing Tai, and Chi-Keung Tang. Gsnet: Joint vehicle pose and shape reconstruction with geometrical and scene-aware supervision. In *The European Conference on Computer Vision (ECCV)*, 2020.

[25] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3d human pose using multi-view geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1077–1086, 2019.

[26] Jason Ku, Alex D Pon, and Steven L Waslander. Monocular 3d object detection leveraging accurate proposals and shape reconstruction. In *CVPR*, pages 11867–11876, 2019.

[27] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *CVPR*, pages 3559–3568, 2018.

[28] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders

for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019.

[29] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009.

[30] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *CVPR*, pages 1019–1028, 2019.

[31] Peiliang Li, Xiaozhi Chen, and Shaojie Shen. Stereo r-cnn based 3d object detection for autonomous driving. In *CVPR*, pages 7644–7652, 2019.

[32] Peiliang Li, Tong Qin, et al. Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 646–661, 2018.

[33] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[34] Shichao Li and Kwang-Ting Cheng. Facial age estimation by deep residual decision making. *arXiv preprint arXiv:1908.10737*, 2019.

[35] Shichao Li and Kwang-Ting Cheng. Visualizing the decision-making process in deep neural decision forest. *arXiv preprint arXiv:1904.09201*, 2019.

[36] Shichao Li, Lei Ke, Kevin Pratama, Yu-Wing Tai, Chi-Keung Tang, and Kwang-Ting Cheng. Cascaded deep monocular 3d human pose estimation with evolutionary training data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6173–6183, 2020.

[37] Lijie Liu, Jiwen Lu, Chunjing Xu, Qi Tian, and Jie Zhou. Deep fitting degree scoring network for monocular 3d object detection. In *CVPR*, pages 1057–1066, 2019.

[38] Lijie Liu, Chufan Wu, Jiwen Lu, Lingxi Xie, Jie Zhou, and Qi Tian. Reinforced axial refinement network for monocular 3d object detection. *arXiv preprint arXiv:2008.13748*, 2020.

[39] Xiaofeng Liu, Yang Zou, Tong Che, Peng Ding, Ping Jia, Jane You, and BVK Kumar. Conservative wasserstein training for pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8262–8272, 2019.

[40] 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.

[41] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[42] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In *CVPR*, pages 2069–2078, 2019.

[43] David Marr. *Vision: A computational investigation into the human representation and processing of visual information*, pages 32–33. MIT press, 2010.

[44] Qinghao Meng, Wenguan Wang, Tianfei Zhou, Jianbing Shen, Luc Van Gool, and Dengxin Dai. Weakly supervised 3d object detection from lidar point cloud. *arXiv preprint arXiv:2007.11901*, 2020.

[45] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *CVPR*, pages 7074–7082, 2017.

[46] Wanli Peng, Hao Pan, He Liu, and Yi Sun. Ida-3d: Instance-depth-aware 3d object detection from stereo vision for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13015–13024, 2020.

[47] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9277–9286, 2019.

[48] T Roddick, A Kendall, and R Cipolla. Orthographic feature transform for monocular 3d object detection. In *30th British Machine Vision Conference 2019, BMVC 2019*, 2019.

[49] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019.

[50] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1711–1719, 2020.

[51] Xuepeng Shi, Zhixiang Chen, and Tae-Kyun Kim. Distance-normalized unified representation for monocular 3d object detection. In *European Conference on Computer Vision*, pages 91–107. Springer, 2020.

[52] Andrea Simonelli, Samuel Rota Rota Bulò, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. *arXiv preprint arXiv:1905.12365*, 2019.

[53] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[54] Xibin Song, Peng Wang, Dingfu Zhou, Rui Zhu, Chenye Guan, Yuchao Dai, Hao Su, Hongdong Li, and Ruigang Yang. Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5452–5462, 2019.

[55] Jiaming Sun, Linghao Chen, Yiming Xie, Siyu Zhang, Qinhong Jiang, Xiaowei Zhou, and Hujun Bao. Disp r-cnn: Stereo 3d object detection via shape prior guided instance disparity estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10548–10557, 2020.

[56] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5693–5703, 2019.

[57] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detec-

tion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4604–4612, 2020.

[58] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[59] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. Data-driven 3d voxel patterns for object category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1903–1911, 2015.

[60] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *CVPR*, pages 2345–2353, 2018.

[61] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2018.

[62] Zhenbo Xu, Wei Zhang, Xiaoqing Ye, Xiao Tan, Wei Yang, Shilei Wen, Errui Ding, Ajin Meng, and Liusheng Huang. Zoomnet: Part-aware adaptive zooming neural network for 3d object detection. In *AAAI*, pages 12557–12564, 2020.

[63] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[64] Linjie Yang, Jianzhuang Liu, and Xiaoou Tang. Object detection and viewpoint estimation with auto-masking neural network. In *European conference on computer vision*, pages 441–455. Springer, 2014.

[65] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1951–1960, 2019.

[66] Xin Zhao, Zhe Liu, Ruolan Hu, and Kaiqi Huang. 3d object detection using scale invariant and feature reweighting networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9267–9274, 2019.

[67] Yi Zhou, Li Liu, Ling Shao, and Matt Mellor. Dave: A unified framework for fast vehicle detection and annotation. In *European Conference on Computer Vision*, pages 278–293. Springer, 2016.

[68] Yi Zhou, Li Liu, Ling Shao, and Matt Mellor. Fast automatic vehicle annotation for urban traffic surveillance. *IEEE Transactions on Intelligent Transportation Systems*, 19(6):1973–1984, 2017.

[69] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.