ISPRS Journal of Photogrammetry and Remote Sensing 159 (2020) 114–128

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Vehicle global 6-DoF pose estimation under traffic surveillance camera

Shanxin Zhang[a,b], Cheng Wang[a,*], Zijian He[a], Qing Li[a], Xiuhong Lin[a], Xin Li[c], Juyong Zhang[d], Chenhui Yang[a], Jonathan Li[e]

[a] Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Information Science and Engineering, Xiamen University, Xiamen, China
[b] School of Information Science and Engineering, Shandong Normal University, Jinan, China
[c] School of Electrical Engineering and Computer Science, Louisiana State University, Baton Rouge, USA
[d] School of Mathematical Sciences, University of Science and Technology of China, Hefei, China
[e] Department of Geography and Environmental Management University of Waterloo, Waterloo, Canada

ARTICLE INFO

Keywords:
Pose
6-DoF
Surveillance camera
Dynamic 3D reconstruction
Deep learning
Point clouds

ABSTRACT

Accurately sensing the global position and posture of vehicles in traffic surveillance videos is a challenging but valuable issue for future intelligent transportation systems. Although in recent years, deep learning has brought about major breakthroughs in the six degrees of freedom (6-DoF) pose estimation of objects from monocular images, accurate estimation of the geographic 6-DoF poses of vehicles using images from traffic surveillance cameras remains challenging. We present an architecture that computes continuous global 6-DoF poses throughout joint 2D landmark estimation and 3D pose reconstruction. The architecture infers the 6-DoF pose of a vehicle from the appearance of the image of the vehicle and 3D information. The architecture, which does not rely on intrinsic camera parameters, can be applied to all surveillance cameras by a pre-trained model. Also, with the help of 3D information from the point clouds and the 3D model itself, the architecture can predict landmarks with few and/or blurred textures. Moreover, because of the lack of public training datasets, we release a large-scale dataset, ADFSC, that contains 120 K groups of data with random viewing angles. Regarding both 2D and 3D metrics, our architecture outperforms existing state-of-the-art algorithms in vehicle 6-DoF estimation.

1. Introduction

Precise localization is important in many endeavors, such as self-driving cars, Mobile Mapping Systems (MMS), Advanced Driving Assistance Systems (ADAS), and Augmented Reality (AR). The aim of this work is on global 6-DoF vehicle pose estimation from dynamic videos using static point clouds of traffic scenes, which has immediate application for autonomous driving and Internet of Vehicles (IOV).

Existing localization systems can be classified into two types of approaches: local and global. **Local approaches** sense dynamic traffic environments using the perception of each vehicle itself. These approaches are convenient, but have two inherent limitations. First, the drivers in the vehicles, or the sensors mounted on the vehicles, have limited fields of view and can see only nearby vehicles from a parallel perspective. Second, vehicles and objects in complicated traffic environments block each other, further restricting the vision of the drivers/sensors. **Global approaches** estimate a scene through a birds view of the entire scene and greatly avoid the aforementioned limitations in the local approaches. However, such a birds eye view often must originate from an on-site surveillance video of the existing traffic system. But, these videos/images from built-in surveillance infrastructures do not provide the coordinates of individual vehicles. Ideally, installing LiDAR equipment on traffic poles and then combining that equipment with regular cameras may provide a convenient solution for obtaining 6-DoF poses of vehicles. Unfortunately, in practice, LiDAR equipment is very expensive. Also, to obtain the 6-DoF poses of vehicles, the collected data must undergo a nontrivial multi-step pipeline (vehicle detection to distance estimation to calculation of 3D coordinates to orientation/location estimation). Hence, using LiDAR equipment at many traffic intersections for vehicle pose estimation is prohibitive.

It is highly desirable to explore the effective estimation of global 6-DoF poses of vehicles from available traffic surveillance cameras. The critical issue, which remains a challenge in computer vision, is estimating 6-DoF poses of multiple vehicles from a single RGB image. Reasoning 3D poses from 2D landmarks is ill-posed in general. After projection, the possible 3D poses consisting of 2d landmarks are infinite (Tome et al., 2017). Also, from a single RGB image, it is impossible to obtain the 3D yaw angle of a vehicle from its 2D heading, because the 2D heading relies not only on the 3D yaw angle, but also on the view distance and view angle (Mousavian et al., 2017).

* Corresponding author.
E-mail address: cwang@xmu.edu.cn (C. Wang).

https://doi.org/10.1016/j.isprsjprs.2019.11.005
Received 19 April 2019; Received in revised form 5 November 2019; Accepted 6 November 2019

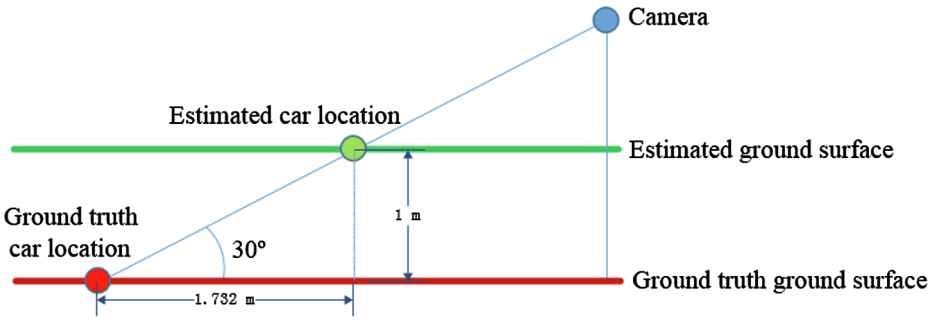


Fig. 1. An example of the error due to incorrectly estimating ground surface.

Traditional image-based 3D reconstruction methods first detect 2D semantic keypoints and then fit 3D keypoints to these 2D keypoints to obtain the 6-DoF pose of a 3D object. These methods provide accurate orientation, but the 3D coordinates of their outputs are relative with a variable scale. Even with a broad assumption of ground surfaces, accuracy in 3D positioning cannot be guaranteed. An example is shown in Fig. 1. When the camera is at an angle of 30° to the ground, for a vertical displacement of the vehicle of 1 m, there is a horizontal displacement of 1.732 m. When extracting the ground from images, the distance of the ground from the camera depends on the intrinsic parameters of the camera, regardless of the method used. With the same 2D-3D correspondences, cameras with different focal length will produce different estimates of the ground. In this work, we reconstruct a 3D dynamic traffic environment for autonomous driving, which requires a true spatiotemporal relationship to reconstruct an authentic traffic environment. To this end, we adopt 3D point clouds to model the ground surface.

With remote sensing, especially Mobile Laser Scanning (MLS), global static high-precision point clouds of a road can be obtained. Such point clouds of traffic scenes, obtained by MLS or Terrestrial Laser Scanner (TLS), can be registered to frames of a video captured from traffic management. This geo-registered image can be used to tackle the ill-posed problem of estimating the geographic location and 3D pose from these video frames, i.e. 2D images. Therefore, unlike autonomous driving systems that require real-time 3D point clouds in the driving process, our proposed architecture requires only pre-scanned point clouds of a scene to estimate the 6-DoF poses of vehicles within a video. Consequently, running our system does not depend upon expensive 3D sensing equipment.

In many existing CNN architectures, the 6-DoF pose of one (and only one) camera from an intact image is regressed (Kendall and Cipolla, 2017; Kendall et al., 2015). The pre-trained network of these approaches is built implicitly on a fixed set of intrinsic parameters, which limits accuracy when this pre-trained network is generalized to handle images captured by different cameras. One example is illustrated in Fig. 2. Except for the focal lengths, the two cameras have the same extrinsic and intrinsic parameters. The focal length of camera 2 is half the focal length of camera 1. Networks trained using images from camera 1 may not effectively estimate images from camera 2. (More results from practical experiments are shown in Table 1.) Therefore, because it is impractical to train a new network for each traffic surveillance camera, the generality of these approaches is limited. In contrast, we propose a CNN architecture to directly estimate the global 6-DoF poses of multiple vehicles from a single image taken by a camera with arbitrary intrinsic parameters.

Table 1
The evaluate results by Kendall and Cipolla (2017) of image (b) and (c) in Fig. 2.

Item	Rotation			Translation		
Ground truth	45°	0°	0°	0 m	−7.07 m	7.07 m
image (b)	47.33°	1.30°	− 1.95°	0.26 m	−6.81 m	7.37 m
image (c)	49.82°	− 8.99°	8.25°	−0.87 m	−9.90 m	11.46 m

To build a network that is independent of camera intrinsic parameters, landmark estimation networks, such as (Wei et al., 2016), can be used to detect 2D landmarks of a vehicle first. Then, corresponding 3D landmarks in the point clouds can be obtained from 2D-3D (image and point clouds) registration. Finally, the 6-DoF pose of the vehicle can be determined by the corresponding 3D landmarks. The accuracy of this approach relies only on the detection of landmarks by the local and global features of vehicles, not camera positions or distances. However, the main shortcoming of these existing landmark estimation networks is their reliance on color and texture information. They do not detect landmarks near regions with little texture or in images with blurred textures. To overcome this limitation, we propose to use the geometric information of the ground and 3D vehicle models to enhance the 2D landmark prediction, making the prediction more robust against shadows, illuminations, and texture variance.

(a) Train image from camera 1 (b) Test image from camera 1 (c) Test Image from camera 2

Fig. 2. Evaluation of 6-DoF pose from cameras with different intrinsic parameters. (a) is an image in training data, which with a random background. (b) and (c) come from different cameras used to test.

Based on the above observations, we propose a novel architecture to perform full global 6-DoF pose estimation of multi-vehicles from an image. By using landmark detection and 2D-3D registration techniques, our predictions are insensitive to the internal parameters of the acquisition camera. This system will then have desirable applications for existing traffic surveillance systems. Our proposed CNN architecture takes advantage of the joint estimation of both 2D and 3D landmark locations. The architecture learns to combine 2D image appearance based on landmark detection, 3D structure information of vehicle CAD models, and ground geometry information provided by point clouds. We compared the estimated poses, both in 2D and 3D, with the results from the state-of-the-art 6-DoF estimation algorithms (Mousavian et al., 2017; Kendall and Cipolla, 2017; Wei et al., 2016; Kendall et al., 2015; Xiang et al., 2016), and found that our approach clearly outperforms all these existing methods.

Another contribution of this work is building new datasets. Datasets with registered pairs of point clouds and images and corresponding 2D-3D landmarks are highly valuable for training pose estimation networks. To the best of our knowledge, no such datasets exist. We construct a large-scale dataset, Autonomous Driving For Smart City (ADFSC), and will release it publicly for comparative study.

The two main contributions of this paper are as follows: (1) Images and point clouds are combined to create a new effective CNN architecture to estimate continuous global 6-DoF poses of multiple vehicles from one image. This architecture for estimation outperforms existing state-of-the-art architectures. (2) A large- scale dataset, ADFSC, that can be used as training data and a benchmark for autonomous driving-related tasks, such as 6-DoF pose estimation, semantic segmentation, and dynamic traffic environment reconstruction is presented.

The remainder of the paper is organized as follows: Related work for both 6-DoF pose estimation and datasets with 2D-3D alignment is discussed in Section 2. Our architecture for Global 6-DoF pose estimation is amplified in Section 3. Construction of the ADFSC dataset is presented in Section 4. Experimental results comparing our design with existing state-of-the-art approaches are given in Section 5. Concluding remarks are given in Section 6.

2. Related work

2.1. 6-DoF pose estimation

The 6-DoF pose estimation is a fundamental problem in the computer vision field. We classify the methods to solve this problem into three main categories: perspective-based, template-based, and Deep Convolutional Neural Network (CNN) based.

Perspective-based methods. Perspective-based methods include Perspective-n-Point (PnP) approaches and deformable model approaches. The PnP problem is to estimate the pose of a calibrated camera from n 3D-to-2D corresponding keypoints. Both linear and non-linear methods have been developed to solve PnP problems. Notable linear methods include the following three: N-Point Linear (NPL) method (Ansar and Daniilidis, 2003), Perspective-four-Point (P4P) (Josephson and Byrod, 2009), and Perspective-three-Point (P3P) (Kneip et al., 2011). Effective non-linear methods include the following five: Efficient PnP (EPnP) (Lepetit et al., 2009), Direct Least Squares method (DLS) (Hesch and Roumeliotis, 2011), Unified PnP (UPnP) (Penate-Sanchez et al., 2013), Maximum Likelihood PnP (MLPnP) (Urban et al., 2016), and Fast and Robust PnP (FRPnP) (Cao et al., 2018). The main limitation of PnP based pose estimation is its sensitivity to the prediction of keypoints, which can be imprecise when the background is complicated, or there is occlusion. Furthermore, accurate PnP based estimations rely on the availability of a good instance-specific 3D model for each object being estimated. This limits the general applicability of PnP estimation.

To address the above PnP problems, Miao et al. (2018) and Pavlakos et al. (2017) extended 6-DoF pose estimation from an object to a class of objects by combining deformable models with the camera model. When some of the keypoints are missing or incorrectly detected, these approaches are more robust. Zia et al. (2015) augmented a deformable model to explicitly include vertex-level occlusion and embedded all instances in a common coordinate frame. However, the coordinates they obtained are relative coordinates, and the application in this paper requires 3D coordinates that can be measured in real traffic scenes. Although great progress has been made with perspective-based methods, these approaches rely on the camera imaging principle. Every trained model, which works best for only a set of fixed intrinsic camera parameters, does not generalize well when dealing with images taken from various heterogeneous traffic surveillance cameras.

Template-based methods. Template-based approaches use either holistic or partial templates to estimate poses. Holistic template-based approaches compare an input image with a set of template images of the object to find the 3D pose of the object (Hinterstoisser et al., 2012; Cao et al., 2016). To construct this holistic template, different viewpoints around a model object are used to generate different images. Partial template-based approaches learn/extract features and use their matching to vote for the pose of the object (Fidler et al., 2012; Xiang et al., 2014; Tejani et al., 2018). Zia et al. (2013) designed a detailed 3D geometric object class model for robust model-to-image matching. Given object detection and Structure From Motion (SFM) point tracks, Dhiman et al. (2016) presented a 3D model for occlusion-aware 3D localization in road scenes. Hejrati and Ramanan (2012) presented an approach to detect and analyze the 3D configuration of objects in real-world images with heavy occlusion and clutter. Liebelt and Schmid (2010) proposed a part-based appearance detection method, yielding an approximate 3D pose estimation and an evaluation score for 3D geometric consistent with 2D part detections. Li et al. (2011) proposed a method of randomized subset-based matching to align a shape model. Generally, template-based approaches often perform well for textureless objects and on scenes with occlusions. However, in this paper, we are committed to applying algorithms directly to aid autonomous driving. This application requires that all vehicles have a true size and must be on the ground in the reconstructed traffic environment. Only in this way can we calculate the spatiotemporal relationship between vehicles. When dealing with images from different cameras, template-based approaches that rely solely on images do not guarantee that the predicted vehicle model will be on the ground.

Deep Convolutional Neural Network (CNN)-based methods. CNN based estimations extract features that integrate both local and global appearance information from an image. For example, Su et al. (2015) use features extracted by AlexNet (Krizhevsky et al., 2012) to classify camera rotation. Many approaches (Xiang et al., 2016; Tulsiani and Malik, 2015), which use object detection results to regress camera rotation, do not predict the translation of a camera. Using Szegedy et al. (2015),Kendall et al. (2015) and Kendall and Cipolla (2017) proposed an architecture for full camera 6-DoF pose regression from an intact image. Li et al. (2017, 2018) employed a probabilistic framework to formalize the notion of intermediate concepts, which, compared with the standard end-to-end training, points to better generalization through deep supervision. Tekin et al. (2018) proposed a CNN architecture single-shot 6D pose prediction that naturally extends the single shot 2D object detection paradigm to 6D object detection. Assuming known full-image camera intrinsics, Kundu et al. (2018) presented an inverse-graphics framework to understand instance-level 3D scenes. Leveraging the off-the-shelf 2D object detector, Li et al. (2019) and Manhardt et al. (2019), in the case of knowing the intrinsic camera parameters, proposed approaches to predict the 3D box of a vehicle by the detected corresponding 2D box. Generally, CNN-based methods are based on explicit or implicit use of intrinsic camera parameters. Their predictions cannot be applied well to different traffic surveillance cameras with unknown intrinsic camera parameters. Moreover, their methods are based, not on the global coordinate system, but on the local coordinate system of the camera. This makes the results of their

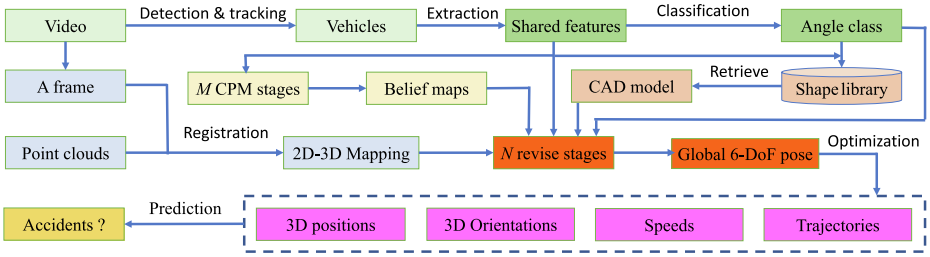


Fig. 3. The architecture of global 6-DoF pose estimation.

algorithms incompatible with high-precision maps. The predicted results of different cameras cannot be combined, thereby further limiting their practical value.

In the field of human pose estimation, Wei et al. (2016) implicitly model long-range dependencies between variables in structured predictions. They take advantage of multi-stage Convolutional Pose Machines (CPM) that directly operate on belief maps from previous stages to deal with vanishing gradients during training, producing increasingly refined estimation results.

Tome et al. (2017) fuse probabilistic knowledge of 3D human poses with CPM and use the knowledge of plausible 3D landmark locations to refine the locations of 2D landmarks. These strategies, using geometric features, can globally optimize the position of landmarks, and therefore, can enhance accuracy and predict the locations of texture-less landmarks. Zamir et al. (2017) established a feedback-based approach in an iterative manner, and achieved competent final results. Inspired by the above two approaches, to directly obtain the global 6-DoF poses of multiple vehicles in an image, we combine CPM with posture information provided by the ground point clouds and the geometric information provided by the vehicle model.

2.2. Datasets with 2D-3D registration

We reviewed three classes of datasets related to 2D-to-3D alignment: image-based, RGB-D image-based, and point clouds-based.

1) Image-based datasets are comprised of images of CAD models from different viewpoints, such as ETH-80 (Leibe and Schiele, 2003), EPFL Car (Ozuysal et al., 2009), ICARO (Lopez-Sastre et al., 2010), IKEA (Lim et al., 2013), Microsoft coco (Lin et al., 2014), PASCAL3D + (Xiang et al., 2014), ShapeNet (Chang et al., 2015), CompCars (Yang et al., 2015), ObjectNet3D (Xiang et al., 2016). These datasets are annotated by a 2D bounding box of the objects, a 6-DoF camera pose, intrinsic camera parameters, etc. They can be used to predict the pose of a camera from an image. However, because they never provide the 3D scene and 2D-3D corresponding landmarks and a 2D pixels to the 3D coordinates mapping table, we cannot use them in our network.
2) RGB-D image-based datasets, such as NYU Depth dataset (Silberman et al., 2012), LabelMe3D (Russell and Torralba, 2009) and SUN RGB-D (Song et al., 2015), provide images combined with 3D depth information that make the 3D detection and pose estimation model more powerful. The 3D depth coordinates in these datasets can be converted to world coordinates. However, these datasets are limited in scale and do not provide 2D-3D corresponding landmarks and 2D pixels to the 3D coordinates mapping table. Thus, they do not meet the needs of our network structure.
3) Point cloud-based datasets can provide a precise and wide range of GPS position and geometric information. NYC3DCars (Matzen and Snavely, 2013) is augmented with detailed geometric and geographic information. This dataset labels full camera poses derived from the method of structure from motion, labels 3D vehicle annotations by manual operation, and obtains geographic information by combining estimated viewpoints. However, this dataset, containing only fewer than 3,000 photos annotated with 3D information and 567 K sparse points, is too small and not suitable as a dataset for deep learning. Furthermore, this dataset does not provide 2D-3D corresponding landmarks and 2D pixels to a 3D coordinate mapping table. KITTI (Geiger et al., 2012) provides ground point clouds, from which partial mapping information between pixels and 3D coordinates can be obtained. Its scale is large enough for deep learning. However, the point clouds in KITTI are acquired by the Velodyne system, and the data is very sparse. The 3D landmarks we used to generate 3D-2D corresponding landmarks cannot be obtained. Besides, the vehicle-mounted camera has a limited perspective of images. The network trained by those images cannot predict a birds-eye-view image.

In summary, to the best of our knowledge, there is no ready-made public dataset available for this particular application, which recovers the global 6-DoF poses of vehicles from traffic surveillance camera videos. Thus, we built a suitable dataset.

3. Global 6-DoF pose estimation

Our architecture (shown in Fig. 3) uses an image and pre-scanned geo-registered point clouds of the scene as input and returns the global 6-DoF pose of the vehicle in an end-to-end manner. The network is trained with a 2D loss that fuses multi-tasking, multi-stage, 2D and 3D information. First, point clouds of the scene are registered to a frame from the video to build the 2D-3D mapping. Second, every vehicle in a video is detected and tracked by the off-the-shelf framework based on pre-trained Single Shot Multibox Detection (SSD) (Liu et al., 2016) and a Kalman filter. The image of the tracked vehicle is cropped as the input of our network. Third, to estimate the vehicle 2D orientation in the image, shared features extracted by the VGG network (Simonyan and Zisserman, 2014) are used to classify the heading of the vehicle into sixteen classes. Moreover, the shared features are used to retrieve the 3D vehicle model, proceed CPM stages and revise stages in the left steps. The 2D orientation class is applied to both the retrieved 3D model and revised stages. Fourth and the most important, the classified heading, retrieved 3D model, estimated belief maps from the output of the M CPM stages, and the 2D-3D mapping are used as input in our N revised stages, and then, through the 2D-3D mapping, used as output in the location of 3D landmarks and global 6-DoF pose of vehicles. Finally, the sequence of the global 6-DoF poses of vehicles in different frames, including trajectories, speeds, 3D orientations, and 3D positions, is optimized (Hastie and Stuetzle, 1989) as the output in our architecture, and that output is used to predict traffic accidents.

In summary, our network takes advantage of the stacked structure, which consists of M CPM stages and N revised stages. To preclude an overall network with many layers at the risk of vanishing gradients, by enforcing intermediate supervision periodically through the network, CPM replenishes gradients and guides the network to produce increasingly accurate belief maps. Based on the results of CPM stages, we add the supervision of ground information and the internal geometric relationship of 3D landmarks into our revised stages to further optimize belief maps.

3.1. 2D-3D registration

In a real traffic scene, because correspondences are difficult to establish due to great noise in reflectance values, registering the point clouds with the image is non-trivial. To ensure accuracy, one frame in the video and scanned point clouds are registered together by manually selecting corresponding points, such as the calibration in KITTI (Geiger et al., 2012). Optimization is carried out by the method of Alternating Direction Multipliers (ADMM) (Zhong and Kwok, 2013). After segmenting the ground and non-ground points by a voxel-based upward growing method (Yu et al., 2015), by projecting the ground point clouds to pixel coordinates using the calibrated camera parameters, an incomplete 2D-3D mapping table is obtained.

In this incomplete 2D-3D mapping table, most of the pixels of ground have corresponding 3D coordinates. Most of the ground pixels have 3D points corresponding to them, but some pixels have no corresponding 3D points due to the lack of point clouds in some places or low density of point clouds. To ensure that every pixel in the ground area of an image has corresponding 3D coordinates, we present an interpolation method to complete the 3D points for the pixels, which do not have corresponding 3D coordinates. A 3D coordinate of a pixel is inserted by the method of 2D-3D lifting introduced in Section 3.4.2. An example of 2D-3D registration in a real traffic scene is shown in Fig. 4.

For modeling the road surface in an image, we adopt point clouds captured by MLS/TLS sensors, which allows us to estimate the 6-DoF pose of true dimensional 3D model that falls to the true ground. This facilitates calculating the spatial relationship of the vehicle in autonomous driving. Point clouds have global coordinates. All predicted vehicle 6-DoF pose has global coordinates, all predicted vehicle 6-DoF poses from different cameras are in the global coordinate system.

3.2. Landmarks selection

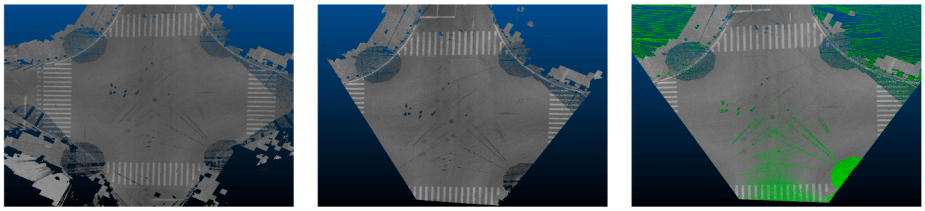
To predict 2D landmarks, the architecture takes advantage of the belief map, the intensity of which indicates the confidence that the respective 2D landmark is located at this position. Five 2D landmarks on the ground that can connect a static scene with dynamic vehicles are selected to build the 2D-3D connection directly. The landmarks include four wheels positioned on the ground (wheel landmarks) and the center point of four wheels positioned on the ground (center landmark). These five 2D landmarks in the image have corresponding 3D landmarks in point clouds. These five landmarks, lying on the ground, annotated in a 3D model are shown in Fig. 5. The orientation of a vehicle is presented in 2D with the two most confident landmarks among the four wheel landmarks. The location of a vehicle is presented in the center 2D landmark. The 3D coordinates of the center 2D landmark and wheel 2D landmarks composed of 2D orientation are obtained by the 2D-3D mapping table. Combining these 3D coordinates with the road point cloud of the traffic environment, the 6-DoF pose of a vehicle can be determined. Its coordinate system is consistent with the point clouds.

The two wheel 2D landmarks that make up the selected direction and the center 2D landmarks are mapped into their 3D landmarks by the 2D-3D mapping table. The three 3D landmarks are put into the point clouds of the traffic environment. On the ground point cloud, the location of the vehicle is determined by the center 3D landmark; the 3D direction of the vehicle is determined by the wheel 3D landmarks. Finally, we obtain the 6-DoF in the point clouds of the traffic environment.

The 3D model provides the spatial positional relationship of the landmarks; the roll and pitch angle of the vehicle with the landmarks on the ground can be inferred. Selecting landmarks in this way allows our architecture to use geometric information from both simultaneously. Unlike with the methods to estimate a camera 6-DoF pose, which requires a complicated coordinate transformation to put the model into the point clouds, the global 6-DoF of a vehicle can be obtained immediately by the 2D-3D mapping table of pixel coordinates and global 3D point clouds.

A problem that merits discussion is whether it is necessary to regard the center landmark on the ground as an additional landmark, because, from the coordinates of the four wheel landmarks, the central location can easily be inferred. Our reasoning is that, suffering from vehicle internal occlusion, the landmarks of the four wheels cannot all be predicted from belief maps, and their average prediction performances are worse than that of the center landmark (See Table 5). Therefore, to predict the center landmark directly is more effective.

(a) A frame from video (b) Color point clouds of traffic evironment



(c) Ground point clouds (d) Registration result (e) Complemention result

Fig. 4. An example of 2D-3D registration for real traffic scene. In (b), the red circle is the placement of traffic surveillance camera. In (e), the green point is the inserted point.

(a) Side view (b) Bottom view

Fig. 5. An example of landmarks selection. The red points in the figure are the selected landmarks.

3.3. Image-based 3D model retrieval

To reduce retrieval space, all of the images in a shape library are classified by their 2D orientation. The retrieval space of an image is the class of the heading nearest the image. For each CAD model, we furnish images every 22.5° in the yaw angle interval [0°, 360°]. Consequently, the shape library is split into sixteen classes. There are m vehicle CAD models in the shape library. For each yaw angle class, a subset $S = \{S_1, ..., S_m\}$ is built by furnishing images with white backgrounds for each vehicle model, $S_i = \{s_{i,1}, ..., s_{i,12}\}$, every 15° in the roll angle interval [30°, 75°], and every 15° in the pitch angle interval [−15°, 15°]. The height from camera to the ground is set to 4.5 m. Thus, if we know the heading of the query image, a total of twelve (four in the roll angle interval [30°, 75°] × three in the pitch angle interval [− 15°, 15°]) are retrieved.

By detecting and tracking vehicles from the video, a sequence of cropped images for each vehicle and their angle classification are obtained. For a vehicle, c images are sampled from the sequence of cropped images, $\mathbf{I} = \{I_1, ..., I_n\}$, at intervals of t frames (here, we set $t = 45$). For a sampled image, a CAD model is retrieved from the class of shape library to which the heading of the model belongs. ObjectNet3D (Xiang et al., 2016) provides a method for retrieving a 3D car model from a 2D image from all of the image shapes. In our architecture, we omit the retrieval space. The retrieved CAD model, i, of an image, I_k, is given as follows:

$$i = \max_{i} \sum_{k=1}^{n} \sum_{j=1}^{12} cos(f(I_k), f(s_{i,j})) \tag{1}$$

where $f(*)$ represents the features extracted from the image. This feature extraction network is the same as the network for extracting shared features.

3.4. 2D landmark detection combined with 3D revision

We believe that more accurate results are obtained by adding, based on CPM, geometric information about the model and the ground to revise 2D landmarks during training. CPM iteratively refines landmark locations with a combination of shared features extracted from an image and the prediction of the previous stage. Using CPM as the backbone, we propose the revised stages as shown in Fig. 6. In this figure, the contents of the left hand side from top to bottom are: (a) CNN networks for shared feature extraction, belief map predictions, and coarse orientation estimation (2D orientation classification). (b) Lift the predicted 2D landmarks into 3D to obtain the 6-DoF pose of the vehicle with the retrieved CAD model. Then project the 3D landmarks onto the image as revised 2D landmarks. (c) Fuse the belief maps generated from the revised 2D landmarks and the predicted belief maps as the output of one stage. These fused belief maps and the shared features are treated as the input of the next stage.

As discussed in the following sections, each revised stage can be divided into three parts: 2D landmarks estimation, 3D revision operation, belief map fusion.

3.4.1. 2D landmarks detection

First, using the CPM stages, we predict the belief maps using crop images as the input. In each stage of CPM except the first, the share feature of image concatenates the features of the belief maps together to predict the belief maps, and output the belief maps to the next stage. Belief map, $b_t^p[u, v]$, at the stage, t and for a landmark, p, represents the degree of confidence that the landmark occurs in any given pixel, (u, v). We transform the belief maps into location L_p simply as follows:

$$L_p = \underset{u,v}{\text{argmax}}\ b_t^p[u, v] \tag{2}$$

To revise the predicted landmarks and achieve better performance, the 2D orientation of the vehicle must be calculated. However, according to the ordered landmarks, orientation can be obtained directly from the belief maps. For example, from belief maps of the left rear wheel $b_t^{lr}[u, v]$ and left front wheel $b_t^{lf}[u, v]$, the locations of the left rear wheel, L_{lr}, and the left front wheel, L_{lf}, are known. In this paper, the orientation is denoted by a vector. Thus, the 2D orientation is $(L_{lf} - L_{lr})$. However, unfortunately, due to the symmetry of the wheels, the results of CNN may not conform to this order restriction, and belief map $b_t^p[u, v]$ may correspond to wheel q where $p \neq q$.

To overcome this problem, we propose a mini classification network to make coarse 2D orientation estimations. The feature used for classification is the shared feature for belief map prediction in CPM. For a coarse orientation estimate, we divide 360 degrees into sixteen classes where each class represents an interval. A mini network contains one Spatial Pyramid Pooling (SPP) (He et al., 2015) layer and two fully connected layers. SPP combines the features of multi-scales and generates fixed-length feature representation. Using this representation, which has multi receptive fields, our classification network converges rapidly and performs better on the 2D orientation classification.

Combining the detected belief maps, the 2D orientation classification and the 2D-3D mapping table in each stage, the 2D pose of a vehicle in the image is obtained. Given any two wheels and the center from the belief maps, there are two possible orientations: the direction between the two wheels and the direction between their midpoints and the center, because it is not known if the two wheels are on the same side. These two directions are orthogonal. If one of the directions falls within the angle interval calculated in the 2D orientation classification, this direction is selected as the 2D orientation. The 2D location of the vehicle is determined by the center landmark. We obtain the 2D pose of a vehicle in the image.

3.4.2. 3D Revision

By lifting the predicted 2D landmarks to 3D, the 3D landmarks may contain existing predicted errors and may not meet geometric constraints, such as four wheel 3D landmarks that compose a fixed scale rectangle. The center 3D landmark is the center of the four 3D wheel landmarks. Tome et al. (2017) proved that plausible 3D landmark locations can refine the locations of 2D landmarks in human pose estimation. We adopt a similar strategy to further refine the predicted result. Our 3D revision stages use ground and model geometry information to revise the output of the belief maps from the previous stage and return revised belief maps. The process is divided into two

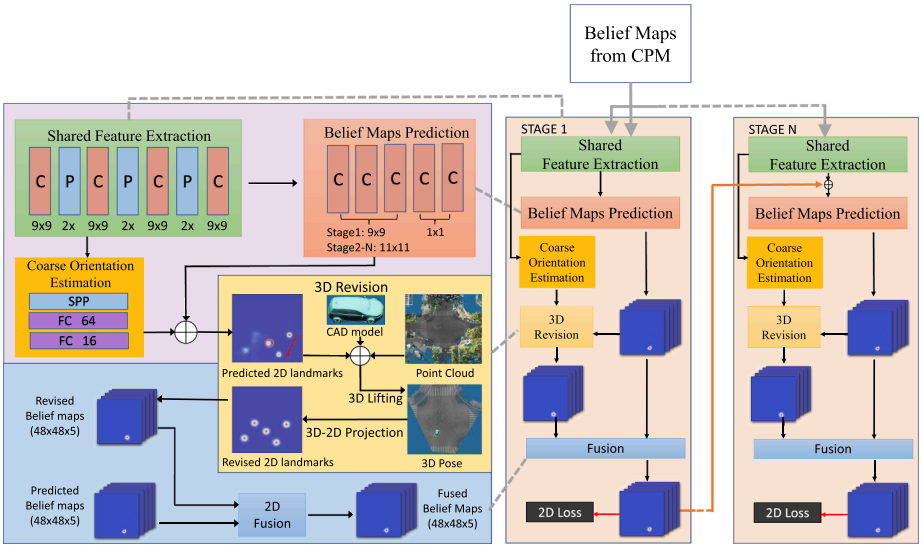


Fig. 6. The illustration of revision stages.

parts: 2D-3D lifting and 3D-2D projection.

2D-3D lifting. In the previous stages, a predicted 2D pose of a vehicle was obtained. The 2D-3D lifting operation lifts the 2D pose of a vehicle to its 6-DoF pose in the point cloud traffic environment and obtains plausible 3D landmarks after posing its 3D model in point clouds.

In theory, given the 2D pose of a vehicle, the corresponding 3D center and 3D orientation can be obtained from the 2D-3D mapping table. However, in fact, there are two obstacles to this approach. Because the 2D-3D mapping table is quite large, continuously searching the 2D-3D mapping table for every training data in every revision stage will greatly reduce the speed. Besides, the cropped image, after detecting and tracking and as the input to our architecture, must be resized to a fixed size, which leads to a change in the correspondence relationship between 2D pixel and 3D point coordinates.

To solve the above problems, the 3D coordinates of the four corners of the cropped image, the center of the original image (principal point), and the position of camera are stored with the cropped vehicle image. Given the following, the problem can be expressed mathematically: (1) a pixel coordinate, (u, v), the height, h, and width, w, 3D coordinates of four corners, of the cropped image; (2) the position, p_c, of the camera; (3) the 3D coordinate of the center of the original image (principal point), p_p. The 3D coordinates of the left upper corner, right upper corner, left lower corner, right lower corner are $p_{ul}, p_{ur}, p_{dl}, p_{dr}$, respectively. The question becomes how to find the corresponding 3D coordinate, p, of pixel coordinate (u, v). A geometrical representation of this problem is given in Fig. 7. Table 2 summarizes inputs and outputs for 2D-3D lifting.

Solution of the lifting problem. In Fig. 7, in addition to known points, the points, p_u, p_d, have the same U coordinate, u, with the point, p, in the image coordinate system. According to the imaging principle of cameras, a plane that passes through p_p and is vertical to the vector between p_c and p_p, is parallel to the actual imaging plane of the camera. This plane is seen as an imaging plane, because it has the same function as the actual imaging plane. The rays from p_c to $p_{ul}, p_u, p_{ur}, p, p_{dl}, p_d, p_{dr}$, intersect the imaging plane at $p'_{ul}, p'_u, p'_{ur}, p', p'_{dl}, p'_d, p'_{dr}$, respectively. Among the intersections, the four corners, $p'_{ul}, p'_{ur}, p'_{dl}, p'_{dr}$, are computed by the intersections between the lines and the imaging plane. Because the cropped image is a square, the quadrilateral, $p'_{ul}p'_{ur}p'_{dr}p'_{dl}$, is a square. In square, $p'_{ul}p'_{ur}p'_{dr}p'_{dl}$, the pixels in the cropped image are evenly distributed. The horizontal and vertical pixel distances of a pixel to pixel, (0, 0), in the cropped image are proportional to the horizontal and vertical 3D distances of the corresponding 3D point to point, p'_{ul}. Therefore, the 3D coordinates of p'_u, p'_d, and p' are computed as follows:

$$\begin{bmatrix} p'_u \\ p'_d \\ p' \end{bmatrix} = \begin{bmatrix} p'_{ul} + (p'_{ur} - p'_{ul}) * (u-1)/(w-1) \\ p'_{dl} + (p'_{dr} - p'_{dl}) * (u-1)/(w-1) \\ p'_u + (p'_d - p'_u) * (v-1) * (h-1) \end{bmatrix} \tag{3}$$

The 3D coordinate, p, of the pixel (u, v) is obtained by computing the intersection between ray, $p_c p'$, and the ground plane determined by points, $p_{ul}, p_u, p_{ur}, p, p_{dl}, p_d, p_{dr}$.

Using the above method, the three landmarks that compose the 2D pose are lifted to their corresponding 3D coordinates. The 3D position and orientation of a vehicle on the ground are obtained. Note that the 3D model of the vehicle is retrieved, and the five 3D landmarks in the model are rotated by quaternion rotation (Vicci, 2001), and then translated into the 3D position. Plausible 3D landmarks of an input image are obtained. The next step is to project those five plausible 3D landmarks onto 2D pixel coordinates.

3D-2D projection. 3D-2D projection is used to project the plausible 3D landmarks onto corresponding 2D landmarks, called plausible 2D landmarks. To maintain the independence of the intrinsic camera parameters, we do not project the 3D coordinates of the revised landmarks onto 2D pixel coordinates by intrinsic and extrinsic camera parameters. As with 2D-3D lifting, we solve this problem geometrically.

Based on the meaning of the symbols in 2D-3D lifting (Fig. 7), the 3D-2D projection, expressed mathematically, is as follows: Given a 3D coordinate, p, how can the pixel coordinate, (u, v), of p be obtained? Table 3 summarizes inputs and outputs for 3D-2D projection.

Solution of the projection problem. The coordinates of p' are obtained by calculating where line, $p_c p$, intersects the imaging plane, $p'_{ul} p'_{ur} p'_{dr} p'_{dl}$. The quadrilateral, $p'_{ul}p'_{ur}p'_{dr}p'_{dl}$, is a square. The distance computed from point, p', to line, $p_{ul}p_{dl}$, is designated as d_u. The distance computed from point, p', to line, $p'_{ul}p'_{ur}$, is designated as d_v. According to the principle of camera imaging, (u, v) is calculated as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \lfloor d_u * (w-1)/|p'_{ul}p'_{ur}| + 0.5 \rfloor \\ \lfloor d_v * (h-1)/|p'_{ul}p'_{dl}| + 0.5 \rfloor \end{bmatrix} \tag{4}$$

Shown in Fig. 8 is the performance of the belief maps of CPM and the belief maps of 3D revised by our method. CPM results are in the first row. Our 3D revision results are in the second row. The white point is the ground truth of a landmark. Red indicates the probability of a landmark at a position. The darker the red, the higher the credibility. From this figure, it is seen that, if a landmark is mis-detected by CPM,

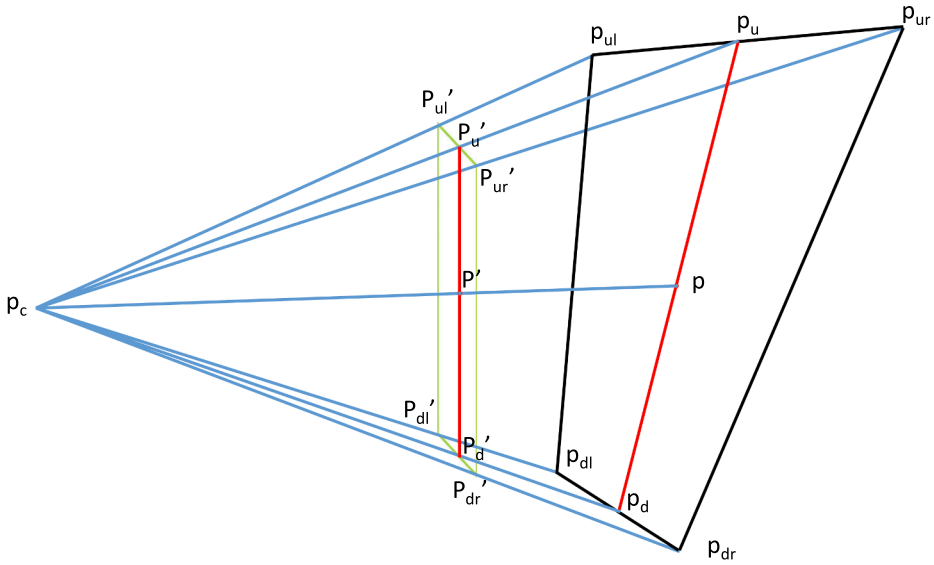


Fig. 7. Diagram to illustrate the 3D geometrical relation in revision.

Table 2
Notation of inputs and outputs for 2D-3D lifting.

2D-3D lifting	Symbol	Description
Input	(u, v)	A 2D landmark
	h	Height of the cropped image
	w	Width of the cropped image
	$p_{ul}, p_{ur}, p_{dl}, p_{dr}$	Four corners of the cropped image
	p_c	Camera position
	p_p	Principal point
Output	p	The corresponding 3D landmark

Table 3
Notation of inputs and outputs for 3D-2D projection.

3D-2D projection	Symbol	Description
Input	p	A 3D landmark
	h	Height of the cropped image
	w	Width of the cropped image
	$p_{ul}, p_{ur}, p_{dl}, p_{dr}$	Four corners of the cropped image
	p_c	Camera position
	p_p	Principal point
Output	(u, v)	The corresponding 2D landmark

our revision completes it; if there is a large error in predicting the position of a landmark, our revision corrects it.

3.4.3. Belief map fusion

Given 2D revised landmarks projected from the 3D space, we initialize a belief map where the value of a pixel is "one" at the landmark position and "zero" at other positions. Then, the pixels are convolved to form an initial belief map, upon which a revised belief map is generated using Gaussian filters.

The output of one stage is the fusion between the belief maps predicted by the CNN-based model b_t^p and the revised belief maps $\hat{b}_t^p$ according to the following equation (Tome et al., 2017):

$$f_t^p = w_{\_t} * b_t^p + (1 - w_{\_t}) * \hat{b}_t^p, \tag{5}$$

where $w_{\_t}$ is the weight to control the degree of revision. If $w_{\_t}$ is too small, it is difficult for the whole architecture to converge and offer precise results. In contrast, if $w_{\_t}$ is too large, the revision is rendered meaningless. To achieve a balance, $w_{\_t}$ is set to be a trainable variable.

The fused belief maps, integrated with the shared features, are passed on to the next stage. For the last stage, we output the Global 6-DoF pose in the approach introduced in 2D-3D lifting.

3.5. The loss function

The loss function, c_t, minimized at each stage, is the l_2 distance

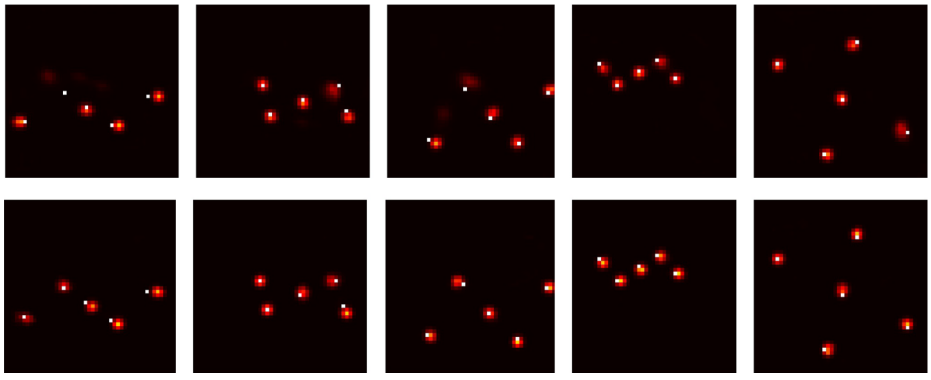


Fig. 8. Performance of the belief maps of CPM and the belief maps of 3D revised by our method.

between fusion belief maps, f_t^p, and ground truth belief maps, b_*^p. The loss function at stage t (the same as that of Wei et al. (2016)) is as follows:

$$c_t = \sum_{p=1}^{6} \sum_{z \in Z} \|f_t^p(z) - b_*^p(z)\|_2^2 \tag{6}$$

where z is a pixel coordinate in the belief map of a landmark, Z is a set includes the coordinates of all pixels. There are a total of six trust maps, five of which are used to represent landmark positions and one to represent background information.

For end-to-end training, the total loss is the sum over the loss of all stages. For the architecture of T stages, the total loss function is given as follows:

$$C = \sum_{t=1}^{T} c_t \tag{7}$$

4. Constructing ADFSC dataset

In traffic scenes, a camera is mounted fixed to a shelf, and the vehicles move in different positions on the ground. Because of symmetry, keeping the vehicles still, we can obtain the same image by taking images with cameras around the vehicles. Using the camera model in Fig. 10, we built a dataset to simulate real traffic scenarios.

In the ADFSC dataset, a total of 120,796 groups of data are generated at present. Each group of data includes an image, 2D and 3D landmarks, 2D and 3D bounding box, 2D-3D mapping table, intrinsic and extrinsic camera parameters, pixel labels to distinguish among cars and shadows and backgrounds. Fig. 9 shows some examples from the ADFSC dataset. In this figure, the 3D models and their 3D landmarks (red points) are shown in the first row. The second row shows the corresponding rendered images and 2D bounding boxes of 3D models in the first row. We add shadow and random background for each image. The third row shows the corresponding semantic annotations for vehicle, shadow, and background for images in the second row. The last row shows the corresponding generated point clouds of the ground, and the 3D bounding box of the vehicles. The ADFSC dataset is built in the following steps: 3D CAD model acquisition and 3D landmarks detection, 2D image rendering and 2D landmarks annotation, 2D-3D registration generation.

4.1. 3D shape acquisition and 3D landmarks detection

The 3D vehicle models we used are from ShapeNet (Savva et al., 2015). We obtained 3D landmarks and 3D bounding boxes through the following steps: First, the size of the model in ShapeNet is not the real size; therefore, we searched the Internet for the real length for each model and then zoomed in the size of the model based on the ratio of the real and model lengths. Second, we rotated and translated a vehicle model in ShapeNet to place each model at the origin in the *xoy* plane and the heading direction parallel to the x-axis. Third, in two simple steps, we located four wheel landmarks, found the points with the smallest z coordinate, and selected the wheel point with the largest absolute x coordinate in each quadrant. Finally, traversing the vertexes of the model, we calculated the 3D bounding box of the model.

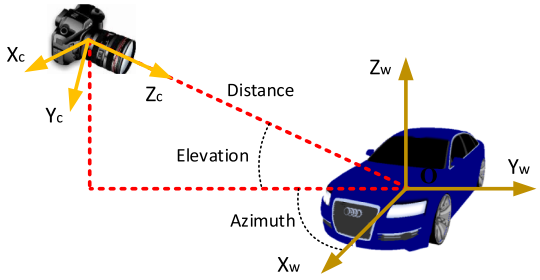


Fig. 10. The camera model of rendering images. X_c, Y_c, Z_c are axes of the camera coordinate system. X_w, Y_w, Z_w are axes of the world coordinate system.

4.2. 2D image rendering and 2D landmarks annotation

We propose a new automatic tool that provides all kinds of vehicle models from any perspective and view distance and simultaneously generates annotation of 2D landmarks. This tool, with a light source of random position and intensity, simulates illumination from the sun to produce vehicle shadows that avoid affecting the network. With this tool, rendered images, intrinsic and extrinsic camera parameters, 2D landmarks projected by 3D landmarks, 2D bounding boxes, and pixel categories are generated in the following ranges for azimuth/yaw, elevation/roll, and in-plane rotation/pitch angles: 0°~360°, 10°~85°, −30°~30°, respectively.

The camera model used in the tool is shown in Fig. 10. To minimize the gap between rendered and real test images, using the tracing algorithm (Cook et al., 1984), we added shadow to vehicles and added backgrounds to rendered images as did Su et al. (2015).

4.3. Generating 2D-3D registration

The steps used to simulate real traffic scenes in 2D-3D registration are as follows: In the *xoy* plane, with the origin as the center and one hundred meters as the side length, a uniform point distribution was used to generate a square point cloud. The distance between the points in the generated point cloud is one cm. Our tool automatically

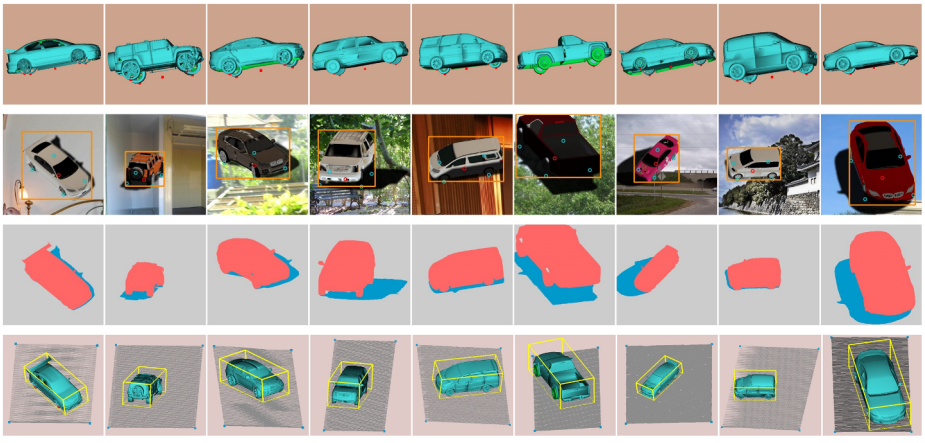


Fig. 9. The data in ADFSC dataset.

generates these point clouds and projects those points onto the rendered image by camera imaging principle to build the 2D-3D mapping table. Projection is followed by interpolation to complete the point cloud for the pixel that has no corresponding 3D coordinate. The interpolation algorithm is the same as 2D-3D lifting.

5. Experimental results

In this section, we first describe how to train our architecture. Then we compare our architecture with the state-of-the-art. Finally, we display performance and an analysis of the runtime of our architecture.

5.1. Training details

To hasten the prediction of landmarks, the number of feature maps in each layer is half that of the corresponding maps in the original CPM network. Before training our revised stages in an end-to-end manner, we independently pre-trained the stages of the CPM network on our dataset. Then, we restored the parameters in our architecture. To increase global contrast and lead to a better view of vehicles in an image, we resized each input image to 384×384, converted them to grayscale images, and processed them using a histogram equalization method. For predicting real traffic images, Adaptive Batch Normalization (ABN) (Li et al., 2018) replaces the mean and variance of the BN layer learned in the training set to diminish domain shift.

The number of CPM stages and revised stages are determined by experiments. We selected at random 30,000 training images and 6,000 validation images from our ADFSC dataset and compared the performance across different combinations between the CPM stages and revision stages (See Table 4). Four metrics are used for evaluation: 2D location error of center (pixel), 2D location error of wheels (pixels), 3D orientation error (degree), and 3D Intersection over Union (IoU). Location error is measured by $l2$ distance on image and point clouds. When calculating 3D IoU, for simplicity, we consider only the surface on the road plane of the 3D bounding box. As shown in Table 4, $M = 4$ and $N = 2$ have the best results for 2D wheels location and 3D orientation; $M = 6$ and $N = 3$ have the best results for 2D central location and 3D IoU. Because 3D IoU is superior in autonomous driving, we finally selected $M = 6$ and $N = 3$ for comparison with the state-of-the-art methods.

5.2. Comparison with state-of-the-art

We compared our architecture with the following five state-of-the-art methods: CPM (Wei et al., 2016), objectNet3D (Xiang et al., 2016), PoseNet1 (Kendall et al., 2015), PoseNet2 (Kendall and Cipolla, 2017), 3D-Deepbox (Mousavian et al., 2017) based on deep learning to estimate 6-DoF poses. Among them, the following three were used to estimate the 6-DoF poses of the camera: objectNet3D (Xiang et al., 2016), PoseNet1 (Kendall et al., 2015), PoseNet2 (Kendall and Cipolla, 2017). For a fixed vehicle 3D 6-DoF pose, different intrinsic parameters of the cameras produce different images, which, conversely, leads to different 6-DoF poses of the camera. Therefore, these methods are affected by the intrinsic parameters of the camera, but our method is not. To verify the effects the intrinsic parameters have on different methods, we divided the experiment into two parts: comparisons under the same camera intrinsic parameters and comparisons under different camera intrinsic parameters. In a comparison, if a method does not estimate the full pose of a vehicle, we use ground truth data to replace the missing value.

For comparison, through a series of operations, the 6-DoF pose of the camera is transferred to the global 6-DoF pose of the vehicle. The four steps in these operations (Fig. 11) are as follows: (1) Transform the 3D landmarks of the 3D model to camera 1 coordinate space by the estimated 6-DoF of the camera. (2) Align the 6-DoF pose of Camera 1 with the 6-DoF pose of Camera 2 in the real scene to transform the 3D landmarks from Camera 1 space to Camera 2 space. (3) Transform the 3D landmarks in Camera 2 space to the global coordinate system with the 6-DoF pose of Camera 2. After this operation, the vehicle may be off the ground because of the deviation of the estimated 6-DoF of Camera 1. (4) Project the 3D landmarks onto the global coordinate system to the ground by the Camera 2 position and point clouds of the ground.

Because the methods PoseNet1 (Kendall et al., 2015), PoseNet2 (Kendall and Cipolla, 2017) require integral images for training and evaluation, we randomly generated 36,000 images (30,000 for training; 6,000 for evaluating) with sizes of 384×384 related to 602 vehicle models with the same intrinsic parameters for the experiment under the same camera intrinsic parameters. Moreover, we generated another 6,000 evaluation images, with changed intrinsic camera parameters, from among the same car models, for experiment under different camera intrinsic parameters. Specifically, the camera focus was changed from (α_x, α_y) to $(\alpha_x * 1.1, \alpha_y * 1.1)$, and the principal point was changed from (u_0, v_0) to $(u_0 + 10, v_0 + 10)$.

We compared the performance of the pre-trained networks with that of competitors by evaluating data that have the same intrinsic camera parameters as the training data. Fig. 12 shows the 2D and 3D comparison results at different tolerances. As seen in Fig. 12a and b, our architecture achieves performance comparable to that of state-of-the-art methods for detecting the 2D landmarks of center and wheel of vehicles. For wheels with a tolerance distance of two pixels, our method scores 92.84%, which is 7.27% higher than that of the closest competitor. As Fig. 12c and d shows, our approach outperforms all other methods under both 3D orientation and 3D-IoU metrics, proving that our 3D revision also benefits the 3D pose estimation. This result shows that our 3D revision, by combining appearance and geometric properties, effectively improves both the prediction of 2D landmarks and 3D pose estimation.

The average results with the evaluation set (See Table 5) show more vividly the improvement of our architecture. Under all four metrics, our method outperforms all other competitors.

We compare the performance of the pre-trained networks with that of competitors by evaluating data that have different intrinsic camera parameters as the training data. As shown in Fig. 13, intrinsic camera parameters greatly affect the methods that estimate directly the 6-DoF poses (objectNet3D, PoseNet1, and PoseNet2). Our methods and CPM are independent of those parameters. Again, our architecture achieves the best performance under all four metrics.

5.3. Performance display and runtime analysis

Performance display. The estimated results on 2D landmarks and a practical application in a real traffic environment are displayed.

Fig. 14 shows the comparison of our method against competitors

Table 4
Performance comparison with different CPM stages and revision stages.

M	4	4	4	5	5	6	6	6
N	2	3	4	3	4	2	3	4
2D center	0.8355	0.8301	0.8594	0.8440	0.8781	0.8269	**0.7741**	0.8871
2D wheels	**1.0720**	1.1178	1.1354	1.1089	1.1284	1.1634	1.1359	1.2251
3D orientation	**3.9952**	4.4334	4.5349	4.0653	4.4175	4.7712	4.7662	5.1484
3D IoU	0.8170	0.8160	0.8137	0.8175	0.8144	0.8112	**0.8209**	0.8089

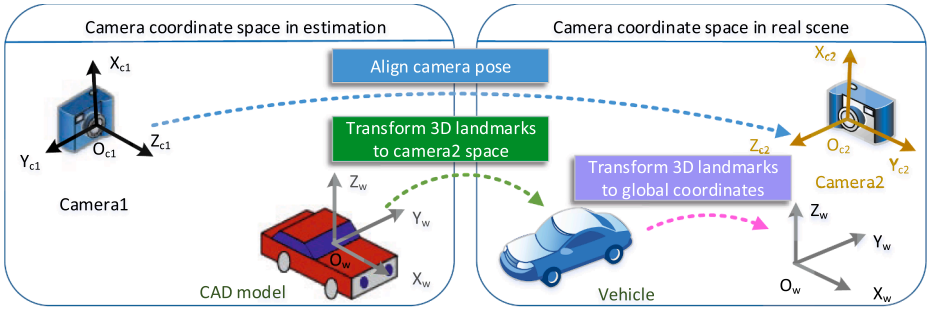


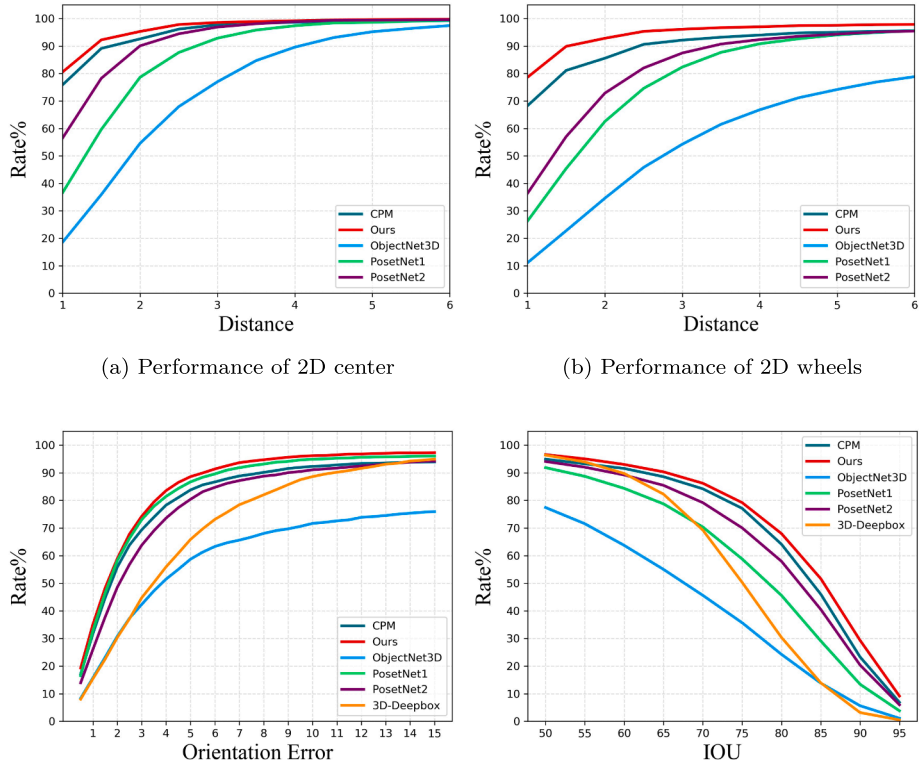
Fig. 11. The transformation from camera 6-DoF pose to global vehicle 6-DoF pose.

with identical intrinsic camera parameters. As the comparison shows, among the generated images, our method achieves the best results. Another performance of our architecture in detecting 2D landmarks is shown in Fig. 15. In this figure, the first row is the estimated 2D landmarks on ADFSC dataset. The last row is the estimated 2D landmarks on a real video. The circle is the estimated landmarks. The yellow arrow is the vehicle heading. It turns out that our architecture, trained with the ADFSC dataset, can effectively estimate the 6-DoF poses of vehicles in a real scene. Furthermore, our architecture has definite robustness to change shadows and resolutions.

Given a frame captured by a traffic surveillance camera, every vehicle in the frame is detected by pre-trained Single Shot Multibox Detection (SSD) (Huang et al., 2017) and tracked by Kalman filter. To ensure that the aspect ratio of a vehicle remains constant, an image is cropped (i.e. the aspect ratio is maintained) if the tracked bounding box is larger than 80×80 pixels and resized to 384×384. This image is the input for our architecture and the architecture outputs 6-DoF pose of the vehicle. To further improve accuracy, a principal curvature optimization algorithm (Hastie and Stuetzle, 1989) is used to optimize the 6-DoF pose of the vehicle within 30 frames by using the inter-frame relationship of the video. Examples are shown in Fig. 16, where, given a video from a traffic surveillance camera and the static point clouds of the traffic environment, the 6-DoF pose for every vehicle in each frame is estimated and the 3D scene reconstructed. Fig. 16a shows the estimated results of a frame. The estimated results of the positional relationship, historical trajectory, vehicle driving speed, and traffic accident prediction are marked, respectively, with yellow 3D bounding boxes, cyan lines, green lines, fuchsia lines, and two red colliding

Table 5
The average performance comparisons against competitors under the same camera intrinsic parameters.

Methods	Center (pixel)	Wheels (pixel)	3D orientation (°)	3D IoU (%)
Ours	**0.7741**	**1.1359**	**4.7662**	**82.09**
CPM	0.9238	1.8349	7.5393	80.58
PoseNet1	1.4632	2.3964	5.6238	74.59
PoseNet2	1.0865	2.1752	7.7169	78.61
ObjectNet3D	2.2061	4.8364	18.5911	62.41
3D-Deepbox	–	–	8.0749	73.52



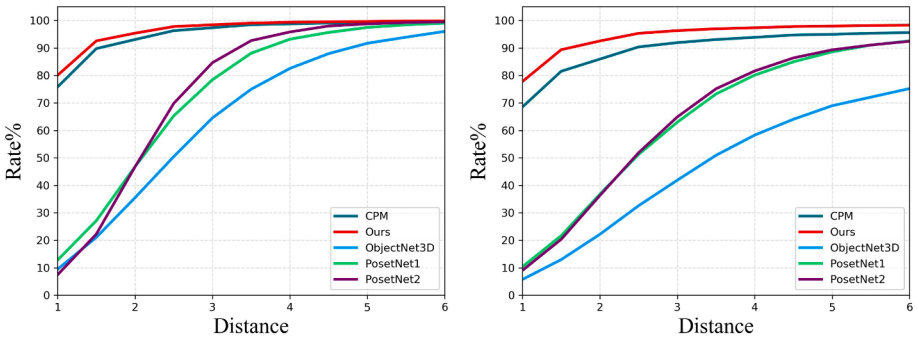
(a) Performance of 2D center

(b) Performance of 2D wheels

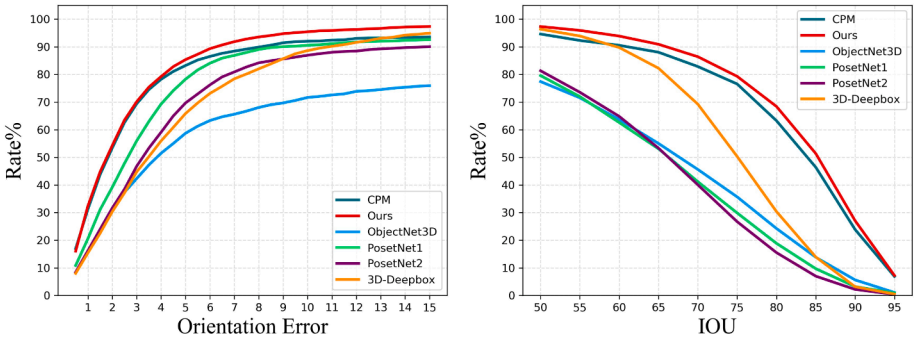
(c) Performance of 3D oreintation

(d) Performance of 3D IoU

Fig. 12. The comparison against competitors under correct camera intrinsic parameters.



(a) Performance of 2D center (b) Performance of 2D wheels



(c) Performance of 3D oreintation (d) Performance of 3D IoU

Fig. 13. The comparison against competitors under different camera intrinsic parameters.

rectangles. Fig. 16b shows the 3D scene of the frame (colored point clouds) reconstructed from Fig. 16a; the camera is highlighted at the bottom in an orange ellipse. Fig. 16c shows an accident predicted from Fig. 16a that happened in a future frame. Fig. 16d shows retrieved 3D vehicle models, which, in Fig. 16e, are then overlaid on the frame using estimated poses.

Given a frame in KITTI captured by the autonomous driving platform Annieway, we first fit the ground of its point clouds with a plane and sample the dense points on this plane. (The error depends on whether the ground is flat.) Then, using the provided transform matrix of KITTI, the sampled points are projected onto the corresponding image to obtain a 2D-3D mapping table. Because the traffic surveillance camera looks down upon the vehicle, the KITTI camera is almost parallel to the vehicle. Therefore, it is easier to see the contact point

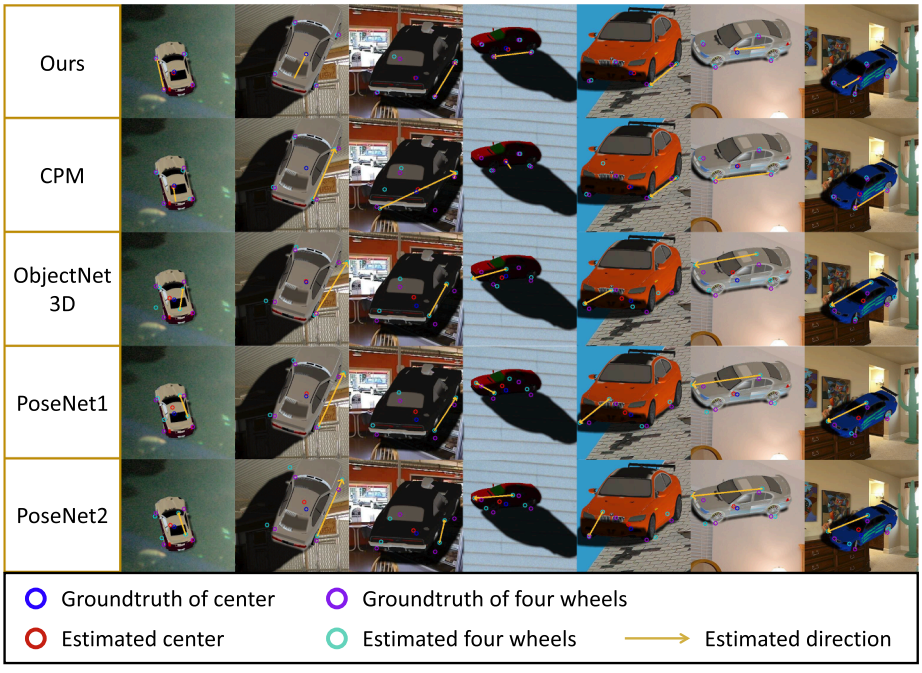


Fig. 14. The performance comparisons against competitors in the image under the same camera intrinsic parameters.

Fig. 15. The performance of our architecture in detecting 2D landmarks.

between the wheels and the ground in the KITTI data than it is in the traffic surveillance camera data. Also, there is a greater possibility of occlusion by other vehicles. For vehicles that are not occluded, our algorithm runs in the KITTI dataset to estimate vehicle poses more easily than in traffic surveillance camera datasets. We do not consider occluded vehicles in this paper. The performance of our algorithm on a frame without occlusion in KITTI is shown in Fig. 17. As seen in this figure, our algorithm performs well when the vehicles are on flat ground and there is no occlusion. The average orientation error and 3D IoU are 1.48°, 79.59% respectively. The gray points in Fig. 17c are the fitted ground.

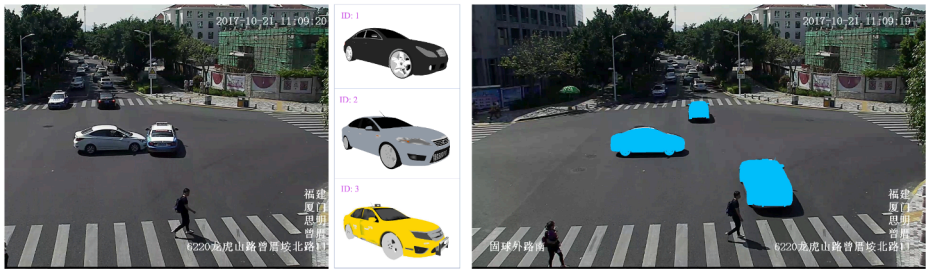
Runtime analysis. For an application of autonomous driving or IoV, the 2D-3D mapping table and the shape library can be written into the hardware in advance. Furthermore, because of the similarity in the appearance and size of the vehicles, it is unnecessary to add every car to the library. Here, we chose only 602 vehicle models to generate the shape library. When analyzing the runtime of our method, we collected videos with a varying number of vehicles. The frame size was 1088 × 1920, and the runtime analysis was performed on a desktop with one NVIDIA GeForce GTX-1080 TI550 GPU. The runtime was split into two parts: detection/tracking and 6-DoF pose estimation. For each vehicle in a frame, detection and tracking took 55 ms; whereas, 6-DoF pose estimation took an average of 44 ms. Because the runtime of the 6-DoF pose estimation in every frame increases linearly with the number of vehicles, our method achieves a speed of 4 fps with 5 vehicles.

If we sacrifice some accuracy, we can hastenthe pose estimation network. We resize each input image from 384 × 384 to 96 × 96, replace the convolution block in the feature extraction part (Fig. 6) with split-shuffle-non-bottleneck (SS-nbt) (Ma et al., 2018), and reduce the CPM stages and revision stages to 2 and 2, respectively. The performance of lightweight architecture with different stages is shown in Table 6. The speed of 6-DoF pose estimation for one vehicle is accelerated from 44 ms to 8.57 ms. The average value of the 3D IoU metric is reduced to 77.84%. Finally, with five vehicles, our method achieves a speed of 10 fps.

6. Conclusions

Dynamic 3D reconstruction of traffic on road is a challenging but valuable task in smart transportation and autonomous driving. In this work, we proposed a near real-time architecture to estimate the global 6-DoF poses of multiple vehicles in a frame from a traffic surveillance camera with support from static 3D laser scanning point cloud of the same scene. This architecture, independent of the intrinsic and extrinsic parameters of the camera, benefits on 3D geometric information to refine results. Moreover, we contributed a dataset used for 6-DoF pose estimation, segmentation, classification, dynamic traffic accident 3D reconstruction, etc.

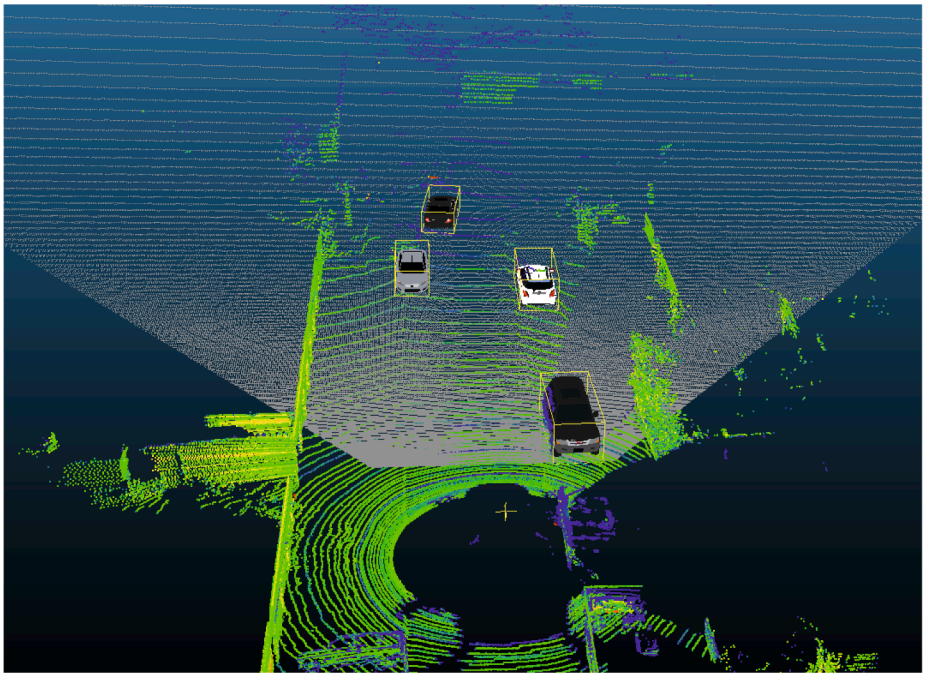
(a) Estimated results in a 2D frame (b) Reconstructed 3D scenario



(c) Predicted accident (d) Models (e) Overlaid image

Fig. 16. The performance of our architecture used in an actual traffic environment.

(a) Estimated results in a 2D frame (b) Overlaid image



(c) Reconstructed 3D scenario

Fig. 17. The performance of our architecture used in KITTI data.

Table 6
Performance of lightweight architecture with different stages.

M	N	2D center (pixel)	2D wheels (pixel)	3D orientation (°)	3D IoU (%)	Time (ms/car)
2	**2**	**1.042**	**1.503**	**6.023**	**77.84**	**8.57**
3	2	0.993	1.428	5.638	78.54	9.20
4	2	0.981	1.409	5.354	79.12	9.61
4	3	0.871	1.307	4.943	81.01	12.65
5	3	0.883	1.289	4.930	81.58	12.90

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is partially supported by the Natural Science Foundation of China (No. U1605254 and No. 61728206), and the Collaborative Innovation Center of Haixi Government Affairs Big Data Sharing and Cloud Services.

References

Ansar, A., Daniilidis, K., 2003. Linear pose estimation from points or lines. IEEE Trans. Pattern Anal. Mach. Intell. 25 (5), 578–589.
Cao, Z., Sheikh, Y., Banerjee, N.K., 2016. Real-time scalable 6dof pose estimation for textureless objects. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 2441–2448.
Cao, M.W., Jia, W., Zhao, Y., Li, S.J., Liu, X.P., 2018. Fast and robust absolute camera pose estimation with known focal length. Neural Comput. Appl. 29 (5), 1383–1398.
Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al., 2015. Shapenet: An information-rich 3d model repository, arXiv preprint arXiv:1512.03012.
Cook, R.L., Porter, T., Carpenter, L., 1984. Distributed ray tracing. In: ACM SIGGRAPH computer graphics, vol. 18. ACM, pp. 137–145.
Dhiman, V., Tran, Q.-H., Corso, J.J., Chandraker, M., 2016. A continuous occlusion model for road scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4331–4339.
Fidler, S., Dickinson, S., Urtasun, R., 2012. 3d object detection and viewpoint estimation with a deformable 3d cuboid model. Adv. Neural Informat. Process. Syst. 611–619.
Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 3354–3361.
Hastie, T., Stuetzle, W., 1989. Principal curves. J. Am. Stat. Assoc. 84 (406), 502–516.
He, K., Zhang, X., Ren, S., Sun, J., 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Machine Intell. 37 (9), 1904–1916.
Hejrati, M., Ramanan, D., 2012. Analyzing 3d objects in cluttered images. In: Advances in Neural Information Processing Systems, 2012, pp. 593–601.
Hesch, J.A., Roumeliotis, S.I., 2011. A direct least-squares (dls) method for pnp. In: 2011 International Conference on Computer Vision. IEEE, pp. 383–390.
Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., Lepetit, V., 2012. Gradient response maps for real-time detection of textureless objects. IEEE Trans. Pattern Anal. Mach. Intell. 34 (5), 876–888.
Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al., 2017. Speed/accuracy trade-offs for modern convolutional object detectors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7310–7311.
Josephson, K., Byrod, M., 2009. Pose estimation with radial distortion and unknown focal length. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 2419–2426.
Kendall, A., Cipolla, R., 2017. Geometric loss functions for camera pose regression with deep learning, arXiv preprint arXiv:1704.00390.
Kendall, A., Grimes, M., Cipolla, R., 2015. Posenet: A convolutional network for real-time 6-dof camera relocalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2938–2946.
Kneip, L., Scaramuzza, D., Siegwart, R., 2011. A novel parametrization of the perspective-

three-point problem for a direct computation of absolute camera position and orientation. In: CVPR 2011. IEEE, pp. 2969–2976.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Informat. Process. Syst. 1097–1105.
Kundu, A., Li, Y., Rehg, J.M., 2018. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3559–3568.
Leibe, B., Schiele, B., 2003. Analyzing appearance and contour based methods for object categorization. In: Proceedings. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003, vol. 2 IEEE pp. II–409.
Lepetit, V., Moreno-Noguer, F., Fua, P., 2009. Epnp: Efficient perspective-n-point camera pose estimation. Int. J. Comput. Vis. 81, 155–166.
Liebelt, J., Schmid, C., 2010. Multi-view object class detection with a 3d geometric model. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, pp. 1688–1695.
Li, Y., Gu, L., Kanade, T., 2011. Robustly aligning a shape model and its application to car alignment of unknown pose. IEEE Trans. Pattern Anal. Machine Intell. 33 (9), 1860–1876.
Li, C., Zeeshan Zia, M., Tran, Q.-H., Yu, X., Hager, G.D., Chandraker, M., 2017. Deep supervision with shape concepts for occlusion-aware 3d object parsing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5465–5474.
Li, C., Zia, M.Z., Tran, Q.-H., Yu, X., Hager, G.D., Chandraker, M., 2018. Deep supervision with intermediate concepts. IEEE Trans. Pattern Anal. Machine Intell.
Li, Y., Wang, N., Shi, J., Hou, X., Liu, J., 2018. Adaptive batch normalization for practical domain adaptation. Pattern Recogn. 80, 109–117.
Li, B., Ouyang, W., Sheng, L., Zeng, X., Wang, X., 2019. Gs3d: An efficient 3d object detection framework for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1019–1028.
Lim, J.J., Pirsiavash, H., Torralba, A., 2013. Parsing ikea objects: Fine pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2992–2999.
Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context. In: European Conference on Computer Vision. Springer, pp. 740–755.
Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. Ssd: Single shot multibox detector. In: European Conference on Computer Vision. Springer, pp. 21–37.
Lopez-Sastre, R., Redondo-Cabrera, C., Gil-Jimenez, P., Maldonado-Bascon S., 2010. Icaro: image collection of annotated real-world objects.
Ma, N., Zhang, X., Zheng, H.-T., Sun, J., 2018. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 116–131.
Manhardt, F., Kehl, W., Gaidon, A., 2019. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2069–2078.
Matzen, K., Snavely, N., 2013. Nyc3dcars: A dataset of 3d vehicles in geographic context. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 761–768.
Miao, Y., Tao, X., Lu, J., 2018. Robust monocular 3d car shape estimation from 2d landmarks. IEEE Trans. Circuits Syst. Video Technol. 28 (3), 652–663.
Mousavian, A., Anguelov, D., Flynn, J., Košecká, J., 2017. 3d bounding box estimation using deep learning and geometry. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 5632–5640.
Ozuysal, M., Lepetit, V., Fua, P., 2009. Pose estimation for category specific multiview object localization. In: CVPR 2009. IEEE Conference on Computer Vision and Pattern Recognition, 2009. IEEE, pp. 778–785.
Pavlakos, G., Zhou, X., Chan, A., Derpanis, K.G., Daniilidis, K., 2017. 6-dof object pose from semantic keypoints. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 2011–2018.
Penate-Sanchez, A., Andrade-Cetto, J., Moreno-Noguer, F., 2013. Exhaustive linearization for robust camera pose and focal length estimation. IEEE Trans. Pattern Anal. Machine Intell. 35 (10), 2387–2400.
Russell, B.C., Torralba, A., 2009. Building a database of 3d scenes from user annotations. In: CVPR 2009. IEEE Conference on Computer Vision and Pattern Recognition, 2009. IEEE, pp. 2711–2718.
Savva, M., Chang, A.X., Hanrahan, P., 2015. Semantically-enriched 3d models for common-sense knowledge. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 24–31.
Silberman, N., Hoiem, D., Kohli, P., Fergus, R., 2012. Indoor segmentation and support inference from rgbd images. Comput. Vision-ECCV 2012, 746–760.
Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.
Song, S., Lichtenberg, S.P., Xiao, J., 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 567–576.
Su, H., Qi, C.R., Li, Y., Guibas, L.J., 2015. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2686–2694.
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.
Tejani, A., Kouskouridas, R., Doumanoglou, A., Tang, D., Kim, T.-K., 2018. Latent-class hough forests for 6 dof object pose estimation. IEEE Trans. Pattern Anal. Machine Intell. 40 (1), 119–132.
Tekin, B., Sinha, S.N., Fua, P., 2018. Real-time seamless single shot 6d object pose prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 292–301.
Tome, D., Russell, C., Agapito, L., 2017. Lifting from the deep: Convolutional 3d pose estimation from a single image. In: CVPR 2017 Proceedings, pp. 2500–2509.
Tulsiani, S., Malik, J., 2015. Viewpoints and keypoints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1510–1519.
Urban, S., Leitloff, J., Hinz, S., 2016. Mlpnp-a real-time maximum likelihood solution to the perspective-n-point problem, arXiv preprint arXiv:1607.08112.
Vicci, L., 2001. Quaternions and rotations in 3-space: The algebra and its geometric interpretation. TR01-014, 1–11.
Wei, S.-E., Ramakrishna, V., Kanade, T., Sheikh, Y., 2016. Convolutional pose machines. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4724–4732.
Xiang, Y., Mottaghi, R., Savarese, S., 2014. Beyond pascal: A benchmark for 3d object detection in the wild. In: 2014 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, pp. 75–82.
Xiang, Y., Kim, W., Chen, W., Ji, J., Choy, C., Su, H., Mottaghi, R., Guibas, L., Savarese, S., 2016. Objectnet3d: A large scale database for 3d object recognition. In: European Conference on Computer Vision. Springer, pp. 160–176.
Yang, L., Luo, P., Change Loy, C., Tang, X., 2015. A large-scale car dataset for fine-grained categorization and verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3973–3981.
Yu, Y., Li, J., Guan, H., Wang, C., Yu, J., 2015. Semiautomated extraction of street light poles from mobile lidar point-clouds. IEEE Trans. Geosci. Remote Sens. 53 (3), 1374–1386.
Zamir, A.R., Wu, T.-L., Sun, L., Shen, W.B., Shi, B.E., Malik, J., Savarese, S., 2017. Feedback networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1308–1317.
Zhong, W., Kwok, J.T., 2013. Accurate probability calibration for multiple classifiers. In: IJCAI, pp. 1939–1945.
Zia, M.Z., Stark, M., Schiele, B., Schindler, K., 2013. Detailed 3d representations for object recognition and modeling. IEEE Trans. Pattern Anal. Machine Intell. 35 (11), 2608–2623.
Zia, M.Z., Stark, M., Schindler, K., 2015. Towards scene understanding with detailed 3d object representations. Int. J. Comput. Vis. 112 (2), 188–203.