Monocular 3D Localization of Vehicles in Road Scenes

Haotian Zhang, Haorui Ji, Aotian Zheng, Jenq-Neng Hwang
University of Washington
{haotiz, hji2, aotianzheng, hwang}@uw.edu

Ren-Hung Hwang
National Chung Cheng University
rhhwang@csie.io

Abstract

Sensing and perception systems for autonomous driving vehicles in road scenes are composed of three crucial components: 3D-based object detection, tracking, and localization. While all three components are important, most relevant papers tend to only focus on one single component. We propose a monocular vision-based framework for 3D-based detection, tracking, and localization by effectively integrating all three tasks in a complementary manner. Our system contains an RCNN-based Localization Network (LOCNet), which works in concert with fitness evaluation score (FES) based single-frame optimization, to get more accurate and refined 3D vehicle localization. To better utilize the temporal information, we further use a multi-frame optimization technique, taking advantage of camera ego-motion and a 3D TrackletNet Tracker (3D TNT), to improve both accuracy and consistency in our 3D localization results. Our system outperforms state-of-the-art image-based solutions in diverse scenarios and is even comparable with LiDAR-based methods.

1. Introduction

Technological advances have made autonomous driving more and more feasible in common driving scenarios. Many large companies such as Google, Tesla, GM, and Uber have tested their self-driving vehicles with success in limited capacities. These vehicles employ a combination of camera, radar, sonar, and LiDAR sensors. Yet the high cost of LiDAR as well as the unreliability of sonar and radar makes them unsuitable for quick large-scale deployment. On the contrary, camera-based autonomous driving has the potential to be a cheap and reliable alternative through steadily advancing computer vision and deep learning techniques.

A general autonomous driving system incorporates three correlated technologies: 3D-based object detection, tracking, and localization. Currently, these three components are explored separately and work has rarely been done to effectively combine them all, so as to compensate for the individual drawbacks and propose a framework solution to the overall system.

Mainstream approaches to 3D-based object detection implement end-to-end architectures. However, there exists two main problems: 1) End-to-end approaches usually require massive amounts of training data and computation resources. 2) Their results are hard to adapt since they are sensitive to training data and cannot be generalized perfectly to different scenarios. To overcome these problems, we propose an integrated system that effectively combines 3D-based detection, tracking and localization in a complementary manner. The system, as shown in Fig. 1, begins with an easy-to-train RCNN-based Localization Network (LOCNet), which is only trained with limited amounts of training data, to provide reasonable initialization of an object's 3D orientation and distance; Further incorporated with a follow-up single frame optimization method based on the fitness evaluation score (FES) on the 2D raw images, we are able to further improve its 3D localization accuracy in various unreliable detection and localization scenarios.

Frame-by-frame detections are never perfect. Temporal information derived from videos can be employed to associate detections across frames and recover missing or unreliable detections. Traditional tracking methods are usually performed in image coordinates or camera coordinates, which may become problematic for autonomous driving scenarios where the camera encounters translational and rotational movements. To solve this, we take advantage of camera ego-motion to perform tracking in 3D world coordinates. The proposed 3D TrackletNet Tracker (3D TNT) utilizes accurate spatial object information along with discriminative appearance features to achieve better tracking performance. In addition, we exploit the temporal consistency and use a multi-frame optimization technique based on the reliable associations from tracking to obtain the best localization performance.

The main contributions are summarized as follows:

- An RCNN-based LOCNet is proposed to simultaneously regress both the 3D orientation and distance of vehicles, which can serve as a good initialization for follow-up optimizations.

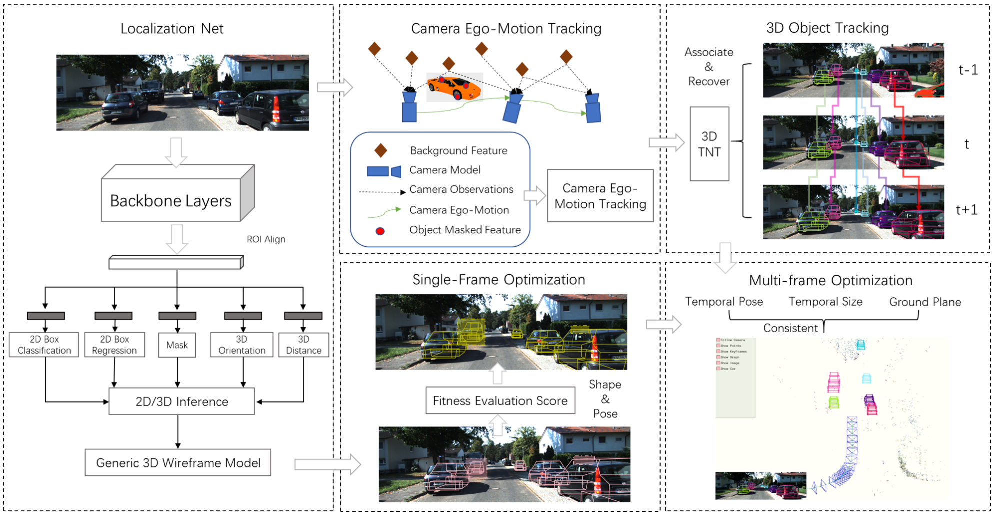


Figure 1. System Overview. The system integrates 3D object detection, single-frame optimization, 3D object tracking and multi-frame optimization to achieve the best localization performance.

- A single-frame optimization technique based on the fitness evaluation score (FES) is applied to ensure the object spatial robustness in the 3D localization.
- A 3D TrackletNet Tracker, which takes into account both discriminative CNN appearance features and accurate 3D spatial object information from each frame, is introduced to associate detections across frames.
- A multi-frame optimization technique is incorporated to reduce the impact from unreliable or missing detections and generate more accurate 3D object localization by taking into account temporal consistency.

2. Related Works

We review related works in the context of 3D object detection and localization.

2.1. 3D Object Detection and Localization

3D object detection and localization can be divided into two groups by the use of sensory data: LiDAR-based and Image-based methods.

LiDAR-based method. Researchers have been leveraging the high precision LiDAR point clouds for accurate 3D object detection and the corresponding localization. Works such as [12, 18, 17] show how to directly manipulate point cloud data with neural networks to obtain the state-of-the-art performance. Yet LiDAR has its own drawbacks such as high cost and sensitivity to adverse weather conditions. These limitations suggest that employing LiDAR-based object detection and 3D localization system is unrealistic in practical, day-to-day applications. Conversely, onboard cameras are relatively cheap, ubiquitous, and can potentially be resilient to most environments.

Image-based method. Cameras provide detailed information in the form of pixel intensities, which at a larger scale can reveal shape and texture properties. Recent works have been trying to explore the prospects of 2D RGB images for 3D detection. More specifically, CNNs are used [13, 16] to extract features from the 2D detected bounding boxes to infer orientation and dimension information; 3D localization of objects are then obtained using the geometric constraints between 3D points and 2D box edges. However, by considering geometric projection as the post-processing step, the error from 2D box detection, 3D object orientation and dimension regression can be aggregated in the subsequent distance estimation module. Some other works [4, 2] consider the problem as a purely geometric problem, known as the bundle adjustment problem (BA), where closed-form or iterative solutions can be applied by assuming a robust correspondence between 2D semantic keypoints and a 3D model of the object. However, these 2D keypoints largely depend on the training data and can be easily affected by partial occlusions or truncation. Furthermore, such BA iterations can usually be very time-consuming due to random initialization.

3. 3D Localization Network

3.1. Network Architecture

The proposed Localization Network (LOCNet) is built upon a popular two-stage object detection network, the Mask-RCNN [10]. LOCNet augments the Mask R-CNN model with a unique depth-aware region proposal network (RPN) [19] and additional learning objectives. In the first stage, we extract and score region proposals by means of anchors based on depth-aware RPN, then ROIAlign for feature cropping is deployed. Based on the top scoring propos-

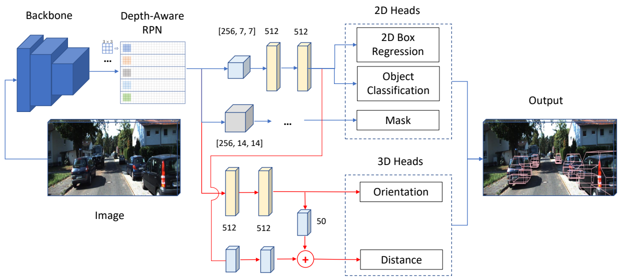


Figure 2. Localization Network (LOCNet). The upper part (in blue) is the typical Mask-RCNN detection framework. The bottom part is the added 3D orientation and distance heads (in red).

als, we use a convolutional encoder to refine the cropped features, then split them up into 5 separate heads. The second stage of the network consists of both classical and customized heads. For the 2D part we use 3 heads for standard multi-class classification, 2D box refinement and (instance segmentation) mask generation respectively. The additional 2 heads are introduced to handle object 3D orientation and distance. The architecture is shown in Fig. 2.

Depth-Aware RPN. ResNet-50 is adopted as a convolution body with a feature pyramid network (FPN) as our detection backbone, which takes a single 2D RGB image to extract feature maps as inputs to a 3D-tailored depth-aware RPN. It has been proven [3] that high-level features related to 3D scene understanding are dependent on depth when a fixed camera is assumed. In this case, we separate the feature map into different row bins and apply individual 2D convolutions for each of them. We believe these depth-aware kernels enable the network to develop location specific features and biases for each bin region. We append a proposal feature extraction layer using depth-aware convolutions to generate features for further processing.

Orientation Head. The orientation head takes the same depth-aware ROI-Aligned feature maps (256×14×14) as input to generate the 3D orientation output. Due to the periodic nature of orientation, it is harder to regress angles explicitly. Although Euler angles, yaw, $pitch$, $roll$, are easily understandable and interpretable for 3D orientation, they are sensitive to non-injectivity and gimbal lock [9]. Thus, we instead regress the quarternions [28] since they are continuous, which can be easily enforced through back-propagation. For the orientation head, given the ground truth quaternion $q \in R^4$ and the predicted quaternion $\hat{q}$, the orientation loss is defined as:

$$L_{ori}(q, \hat{q}) = \left\| q - \frac{\hat{q}}{\|\hat{q}\|_2} \right\|_2 . \tag{1}$$

Distance Head. The distance head takes a concatenated input, from both depth-aware ROIAligned feature maps (256×14×14) and convolved 512-dim features for bounding-box classification/regression, to form more informative input features for 3D distance. The concatenated features are assumed to implicitly encode the 3D orientation information and pre-defined object size information via the incorporation of the convolved 512-dim features. To generate the ground truth for this distance head, we need to transform the 2D detected objects' box center, height and width (u_p, v_p, h_p, w_p) in 2D image coordinates to their corresponding (u_c, v_c, h_c and w_c) in 3D camera coordinates so that the ground truth 3D distances can be determined.

$$\begin{aligned} u_c &= \frac{(u_p - c_x)z_s}{f_x}, h_c = \frac{h_p}{f_x}, \\ v_c &= \frac{(v_p - c_x)z_s}{f_y}, w_c = \frac{w_p}{f_y}, \end{aligned} \tag{2}$$

where the parameter vector $[f_x, 0, c_x; 0, f_y, c_y; 0, 0, 1]$ stands for the camera intrinsic K, and z_s is the projective distance [27].

Huber loss is adopted to formulate the penalty in distance estimation: given ground truth distance d and the prediction $\hat{d}$, the distance loss is:

$$L_{dis}(d, \hat{d}) = \begin{cases} \frac{1}{2}(d - \hat{d})^2/\delta & if \left| d - \hat{d} \right| < \delta, \\ \left| d - \hat{d} \right| - \frac{1}{2}\delta & \text{otherwise.} \end{cases} \tag{3}$$

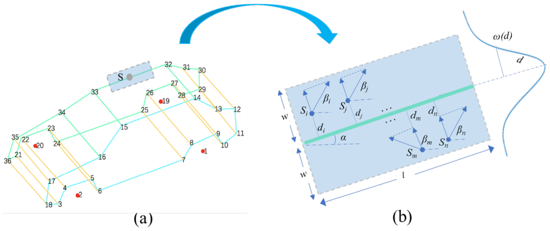
where the hyper-parameter δ controls the boundary of outliers.

3.2. Multi-Task Loss

The following total loss function L_{total} is minimized to train our proposed LOCNet. The first three loss terms are the standard Mask R-CNN multiclass loss L_{cls}, 2D bounding box regression losses L_{box} and mask loss L_{mask}, respectively as defined in [10]. The last two terms are the orientation loss L_{ori} and distance loss L_{dis} respectively, as defined in Eq. (1) and Eq. (3).

$$\begin{aligned} L_{total} =& w_{cls}L_{cls} + w_{box}L_{box} \\ &+ w_{mask}L_{mask} + w_{ori}L_{ori} + w_{dis}L_{dis}. \end{aligned} \tag{4}$$

We show in the later ablation study Sec. 8.4 that our novel formulation for distance regression can produce much more accurate 3D localization estimation compared to methods that treat the distance estimation as a post-processing step [16, 13]. This accurate estimation of both orientation and distance is particularly crucial for the autonomous driving applications, where the location of the objects is of primary importance. Furthermore, the predicted orientation and distance of each object from LOCNet also serve as a good initialization for the subsequent 3D localization optimization part.



(a) (b)

Figure 3. A deformable vehicle model.

4. Single-frame Optimization

Although the orientation and distance estimation results from LOCNet can deal with partial occlusions and truncation cases in most of the time, they are not accurate enough for 3D localization. As you may also notice, LOCNet only focuses on the localization on 2D and 3D without considering the object size, which is an important aspect of 3D detection. In this section, we propose a lightweight optimization pipeline for single-view that refines the initial estimates to ensure localization robustness. Meanwhile, the size of the detected object can also be obtained through this refined optimization. A 3D deformable vehicle model containing 36 shape parameters is set up as prior information and will be described in details in Sec. 4.1. An effective fitness evaluation score (FES) is then used to evaluate the fitness between the 2D projection of the 3D deformable vehicle model and raw image data. Moreover, the fitness evaluation is combined into an optimization framework to select better individuals from the combined parameter space based on an iterative population selection strategy.

4.1. 3D Deformable Vehicle Model

Our deformable model [2] of a vehicle is a 3D wireframe model with 36 shape parameters, which is shown in Fig. 3. The shape parameters have respective changeable values and are interdependent. The pose P of a vehicle can be determined by its position (X, Y, Z) and its orientation θ about the vertical axis of the camera coordinates. The projection relation between each vertex of the 3D car model $V_m = (X_m, Y_m, Z_m)$ in object coordinates and its corresponding point v_m in image coordinates is shown in Eq. (5).

$$v_m = K \cdot P \cdot V_m. \tag{5}$$

With the pose parameters initialized by LOCNet, the 3D vehicle model can then be projected onto the image plane to match with raw image data. An accurate and efficient method is required for fitness evaluation between the projection of 3D vehicle model and image data, which will be described in detail in the next subsection.

4.2. Fitness Evaluation Score

Fitness evaluation between the projected 3D vehicle model and image data is proposed in [31, 30]. Owing to its effective performance, here we adopt it to our deformable-model-based approaches. Most model-based vehicle localization methods require an initialized pose to project. In this work, the pose initialization is provided by the LOCNet, and the wireframe model can be projected onto 2D image coordinates to form a set of projected line segments. Based on the initial orientation θ, we are able to identify which line segments are visible. For every visible projected line segment, whose direction is denoted as α with length l and width $2w$ in image coordinates, we form a $l \times 2w$ virtual rectangle, as shown in Fig. 3. Along the gradient directions of pixels with large gradient magnitude values in the rectangle should coincide with the perpendicular direction of the projected line, if the line fits the image data well. Then, we are able to estimate the fitness score from the gradient information of all pixels within the bounding rectangle. For pixel s_i within the rectangle, we can simply compute its gradient magnitude $m(u, v)$ and gradient angle $a(u, v)$ from pixel differences.

The fitness error score $E(s_i)$ is calculated by the component of its gradient magnitude perpendicular to the direction in Eq. (6). It is also evident that not all pixels in the rectangle have the same weight for fitness evaluation. For those closer to the visible projected line segment, the pixels should contribute more to the FES. In this case, a weight value $\omega(d_i)$ is assigned to every pixel, where d_i is the distance between s_i and projected line segment, and $\omega \sim N(\mu = 0, \sigma = w)$, which is a standard normal distribution. The total FES value, E between the projection of the 3D vehicle model and image data can be obtained from all visible projected line segments, as shown in Eq. (7).

$$E(s_i) = |m(u, v) \cdot \sin(a(u, v) - \alpha)|\,. \tag{6}$$

$$\begin{aligned} E &= \sum_l \log(E_l) \\ &= \sum_l \sum_{s_i} [E(s_i) \cdot \omega(d_i)]. \end{aligned} \tag{7}$$

Our approach performs efficiently and accurately for 3D object localization upon a good pose initialization from LOCNet. FES has several advantages comparing with many other existing methods. Compared to [2], whose pose and shape priors are largely dependent on 2D semantic keypoint trained by a neural network. Though they use an iterative re-weighted optimization scheme to tackle erroneously detected keypoints, we outperform them by using stable and invariant edge information in the local region instead of points, and also by avoiding time-consuming keypoint data labeling and network training. Furthermore, we can also easily handle serious occlusion and truncation cases due to good pose initialization.

5. Camera Ego-Motion and Object Tracking

Our tracking is performed in the world coordinates. To transform from 3D camera coordinates to 3D world coordinates, the feature-based visual odometry [14] is introduced here to recover the camera pose through ORB features [20] extracted in every frame. Since ORB features must be located on the static background scene, instead of on the highly dynamic objects, we utilize the segmentation masks predicted from LOCNet in Sec. 3 to discard those ORB features that are located on the detected objects, keeping those in the static background. We subsequently find correspondence of the background ORB features of the current frame with those of the previous frame. Outliers are further rejected by the RANSAC algorithm [7] as facilitated by the fundamental or homography matrix.

To make it concise for later sections, we define the notations in the following as also shown in Fig. 4. ${}^w(\cdot)$, ${}^c(\cdot)$, and ${}^i(\cdot)$ are used to denote the world, camera and image coordinates respectively. For the k^{th} object at time t, we use ${}^cO_t^k = \{\ {}^cX_t^k,\ {}^cY_t^k,\ {}^cZ_t^k,\ {}^c\theta_t^k,\ {}^cH_t^k,\ {}^cW_t^k,\ {}^cL_t^k\ \}$ to describe its distance, orientation and size, which are obtained from Sec. 3 and Sec. 4. For the camera ego-motion, we use ${}^wc_t = \{\ {}^wT_t,\ {}^wR_t\ \}$ to indicate the camera translation and rotation.

The camera motion is continuously estimated from time 0 to T: ${}^wC = \{{}^wc_t\}_{t=0:T}$. Given the measurements of the n^{th} sparse ORB features, which are anchored on the background: ${}^ip = \{{}^ip_t^n\}_{t=0:T}$ and their corresponding 3D positions: ${}^wP = \{{}^wP_t^n\}_{t=0:T}$. We formulate the camera ego-motion tracking as the following:

$${}^wC, {}^wP = \mathop{\arg\min}_{{}^wC,{}^wP} \sum_{n=0}^{N}\sum_{t=0}^{T} \left\| r_p({}^ip_t^n, {}^wc_t, {}^wP_t^n)\right\|^2_{\Sigma_n^t}, \quad (8)$$

where $\|r_p()\|^2_\Sigma = {r_p}^T\Sigma^{-1}r_p$, the Mahalanobis norm. This is a common visual odometry formulation, where the camera poses are estimated based on a nonlinear least-squared formulation, also referred to bundle adjustment (BA) [25]. After we solve the camera poses, we can simply convert the object measurements from camera coordinates into world coordinates by using:

$${}^wO_t^k = {}^w{C_t}^{-1} \cdot {}^cO_t^k, \quad (9)$$

where the ${}^wO_t^k$ stands for object location (distance), orientation and size in world coordinates.

5.1. 3D TrackletNet Tracker

To take advantage of the temporal consistency for improving the localization performance further, we need tracking to associate corresponding objects along time. The proposed 3D TrackletNet Tracker (3D TNT) takes both discriminative CNN appearance features and accurate object spatial information from each frame to ensure tracking robustness. Inspired by the 2D TNT [26], which builds a graph-based model that takes 2D tracklets as the vertices and use a multi-scale CNN network to measure the connectivity between two tracklets, we further extend the work into 3D tracking scenarios. Our 3D TrackletNet Tracker consists of three key components:

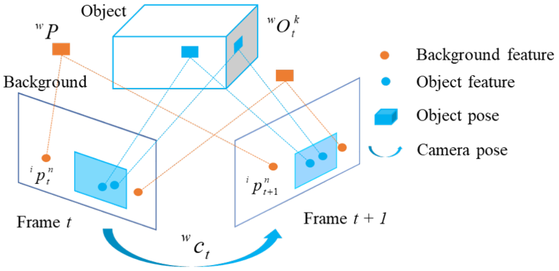


Figure 4. Notation visualization.

Tracklet Generation. Given the refined vehicle localization of each frame (see Sec. 5.1), each tracklet, generated by 2D box appearance similarity based on CNN features derived from FaceNet and 3D intersection-over-union (3D IOU) between adjacent frames, is denoted as a node ($v \in V$) in the graph.

Connectivity Measurement. Between every two tracklets, the connectivity (similarity) $p_e (e \in E)$ is measured and its inverse (dissimilarity) is used as the edge weight in the graph model. To calculate the connectivity, a multi-scale TrackletNet is built as a classifier, which can concatenate both temporal (multi-frame) and appearance features for the likelihood estimation. For each frame t, a vector consisting of the 7-D object measurements ${}^wO_t^k$, concatenated by an 512-D embedding appearance feature extracted from the FaceNet, is used to represent an individual feature of the input frame.

Graph-based Clustering. After the tracklet graph is built, graph partition and clustering techniques, i.e., assign, merge, split, switch, and break operations [24] are iteratively performed to minimize the total cost on the whole graph.

Based on the tracking results from the 3D TNT, we are not only able to associate every object across frames, but also can deal with errors caused by the occlusions and missing detections. This information will be used in the subsequent multi-frame optimization part to further improve the localization performance.

6. Multi-frame Optimization

In the context of autonomous driving, the temporal information can be readily exploited to obtain better localization predictions. Based on the 3D object measurements within each frame from Sec. 4 and tracking results across frames from Sec. 5.1, several temporal consistency constraints can be further imposed to refine the localization results, which are introduced by the following:

Temporal Location and Orientation Consistency. The object location and orientation cannot have a very abrupt change between two adjacent frames, as reflected in the location and orientation consistency regularizer L_P. Here we further denote k^{th} ($k \in K$) object location in frame t as ${}^{w}l_t^k = \left\{\ {}^{w}X_t^k,\ \ {}^{w}Y_t^k,\ \ {}^{w}Z_t^k\ \right\}$, and object orientation as ${}^{w}\theta_t^k$,

$$\mathcal{L}_P = \sum_{t=0}^{T-1}\sum_{k=1}^{K}\left(\left\|{}^{w}l_{t+1}^{k} - {}^{w}l_{t}^{k}\right\|^2 + \left\|{}^{w}\theta_{t+1}^{k} - {}^{w}\theta_{t}^{k}\right\|^2\right). \tag{10}$$

Temporal Size Consistency. Since the vehicle object of interest is considered as a rigid body, its size (height, width and length) in the 3D world coordinates is supposed to remain the same along time. Here we further denote ${}^{w}s_t^k = \left\{\ {}^{w}H_t^k,\ \ {}^{w}W_t^k,\ \ {}^{w}L_t^k\ \right\}$ as the object size.

$$\mathcal{L}_S = \sum_{t=0}^{T-1}\sum_{k=1}^{K}(\left\|{}^{w}s_{t+1}^{k} - {}^{w}s_{t}^{k}\right\|^2). \tag{11}$$

Ground Plane Consistency. Assume all the observed objects are residing on the same plane, which is usually the case for autonomous driving scenarios. A base plane n_b can be formed by the roof surface of the 3D car model and its normal vector should have the same direction as the ground plane normal vector n_g computed in [15]. We use the dot product $(\cdot)$ to measure the similarity between two vectors.

$$\mathcal{L}_N = \sum_{t=0}^{T-1}\sum_{k=1}^{K}\left\|\left(n_g\right)_t \cdot \left(n_b\right)_t^k\right\|. \tag{12}$$

Total Optimization Loss. The overall optimization loss $\mathcal{L}_{total}$ consisting all the terms Eq. (10), (11), (12) can be written as

$$\min_{l,\theta} \mathcal{L}_{total} = \omega_P\mathcal{L}_P + \omega_S\mathcal{L}_S + \omega_N\mathcal{L}_N. \tag{13}$$

Here $\omega_P, \omega_S, \omega_N$ are the weights to adjust the relative importance for the loss terms. In practice, the loss terms are defined with Huber loss function to avoid the effect of outliers. The above problem can also be minimized using Ceres Solver with a Levenberg-Marquardt optimization method and Iterative Schur as the linear solver. After the multi-frame optimization is performed in world coordinates, we transform the adjusted measurements back to camera coordinates to compare the localization performance.

7. Implementation Details

We implement our LOCNet framework using mmdetection [5]. The hyperparameters w_{ori} and w_{dis} in Eq. (4) is set to 1.0, 0.1 to scale the loss accordingly. In order to decrease the distance outlier penalty and stabilize the training, the δ in Eq. (3) is set to 1.5 meters for KITTI and 2.8 meters for ApolloCar3D individually. The base learning rate starts from 1e-3 and the models are trained up to 3e4 iterations for both models.

The 3D TNT is also implemented in Pytorch and purely trained on KITTI tracking dataset. The extracted appearance features have 512 dimensions and object measurements have 7 dimensions. The time window is set to 64. Adam optimizer is adopted with a learning rate of 1e-3 at the beginning. We decrease the learning rate by 10 times for every 2,000 steps until it reaches 1e-5.

The FES and Multi-frame Optimization framework are optimized using the estimation of distribution algorithm (EDA) [23] and the Ceres Solver [1] respectively. In order to avoid scale ambiguity caused by the monocular systems, we also slightly modify the monocular ORB-SLAM2 [14] based on [29] to avoid the scale ambiguity.

8. Experiments

8.1. Dataset

Evaluations are performed on various autonomous driving datasets:

- KITTI [8]: KITTI multi-object tracking dataset contains 20 video sequences for training and 28 sequences for testing. In terms of the data split, we follow [21] and use 1, 6, 8, 10, 12, 13, 14, 15, 16, 18, 19 as the *val* set and other sequences as the *train* set, through our LOCNet training.
- ApolloCar3D [22]: This dataset contains 5,277 driving images with over 60K car instances, aiming at localizing 3D objects in single images.

8.2. Qualitative Results Under Diverse Scenarios

We demonstrate the system performance on different datasets under various driving scenarios, which include object far distance estimation, occlusion, truncation, and complex road conditions. Some examples of the reprojected images and their corresponding 3D views are shown in Fig. 8.1. We use different colors to represent different vehicles. All the observed cars are visualized in both camera (left side of each column) and world (right side of each column) coordinates for ApolloCar3D and KITTI tracking dataset.

8.3. Quantitative Evaluation

For KITTI, we define the true positive of the object 3D localization results if the 3D IOU is greater than 0.5 against

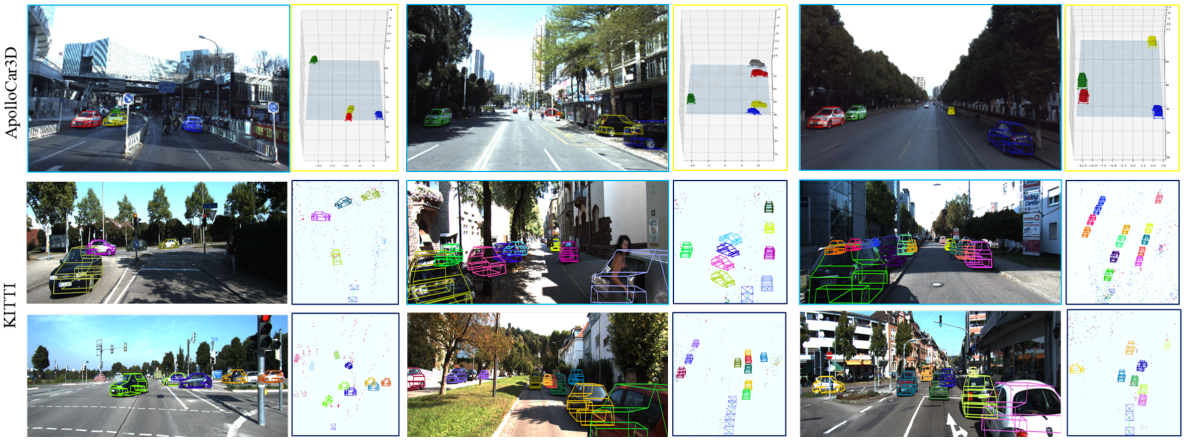


Figure 5. Qualitative examples under diverse scenarios. The top row are the results on the ApolloCar3D instances, and the bottom 2 rows show the results on some image frames of the KITTI tracking dataset. The system is capable of estimating the shape and pose (without scale-factor ambiguity) of objects. The images of the scenes contain the projection of the estimated shapes of cars.

the ground truth, as this IoU threshold is widely used and rather strict for image-based methods. For ApolloCar3D, we adopt the official 3D overlap criteria. The quantitative performance are shown in Table 1 and 2.

KITTI. As the KITTI tracking *test* set ground truth is not released to users, we have to use the KITTI *val* set for 3D evaluation. Our framework is evaluated on both AP_{BEV} and AP_{3D} metrics and the Car class is split into 3 difficulties: *Easy*, *Moderate* and *Hard*. For 3D localization performance based on single frame images, we compare our LOCNet with/without FES optimization with monocular 3D object detection methods [3, 16]. It can be seen that our method using only LOCNet can achieve 36.06%, 25.44% and 24.19% respectively on AP_{3D}. By adding the FES optimization, we observe significant gains with 48.40% (↑ 12.34%), 38.59% (↑ 13.15%) and 32.69% (↑ 8.5%) on AP_{3D}. Furthermore, by considering the temporal information when dealing with video sequences, we compare our overall system with [6, 11] by adding the proposed 3D TrackletNet and multi-frame optimization methods. We further achieve more gains with 56.54% (↑ 8.14%), 44.23% (↑ 7.1%) and 36.91% (↑ 4.22%) on AP_{3D} and outperform the state-of-the-art image-based methods. Considering the best 3D localization performance, our overall system is even comparable with LiDAR-based methods [12] with reasonable margins ($\sim 4-6\%$).

ApolloCar3D. The 2D evaluation metrics for ApolloCar3D follow similar instance mean AP as the MS-COCO. Instead of using 2D mask IoU to define a true positive, the 3D metric contains the perspective of shape, 3D distance and orientation. Since there are no available published methods that we can compare with, we only show the performance of baseline and our LOCNet with/without FES optimization. We first provide the 2D evaluation metrics (AP) as shown in Table 2. We achieve an mAP of 13.3 by using LOCNet only and we also find that small objects are harder to detect, which commonly indicates the object longitudinal axis distance is far away from the camera. The accurate estimation of large transnational distance value is thus more important. Still, for the 3D evaluation metrics, with the help of FES optimization, the shape similarity, distance and orientation scores are improved by 0.03, 0.04m, 0.6° respectively. Besides, the 2D mAP also increases to 14.1% (↑ 0.8%).

Although we claim that it is not a complete fair comparison between our method and the state-of-the-art image-based 3D object detection methods due to our use of temporal optimization via 3D tracking. However, we stress that our approach only uses a monocular camera and can accurately and efficiently localize the 3D objects with spatial robustness and temporal consistency, which is essential for continuous perception in autonomous driving.

8.4. Ablation Study

We perform the ablation study on our LOCNet and the overall system.

Localization Network. To explicitly show the effectiveness of our proposed LOCNet, we perform the ablation study on depth-aware RPN (D-RPN), orientation head (O-H) and distance head (D-H) for both KITTI and ApolloCar3D validation set. O-H+D-H represents we use the features from original RPN in Mask-RCNN to regress the distance and orientation. D-RPN+O-H indicates that the network only regresses the orientation, then the distance is obtained by a post-processing stage [13]. D-RPN+O-H+D-H represents both the distance and orientation are regressed simultaneously from the network, where the distance head uses the concatenated features. As seen in Table 3, by incorporating both depth-aware RPN and the distance head, the network can achieve the distance and orientation errors

Table 1. Performance of 3D localization methods using different modality on KITTI *val* set.

Method	Modality	Type	AP_{BEV} (IoU>=0.5)			AP_{3D} (IoU>=0.5)		
			Easy	Mod	Hard	Easy	Mod	Hard
M3D-RPN [3]	Image	Mono	41.53	31.02	26.65	37.41	27.11	23.73
Shift-RCNN [16]	Image	Mono	39.64	30.33	25.90	31.48	24.04	23.60
LOCNet (Ours)	Image	Mono	42.86	30.43	26.35	36.06	25.44	24.19
LOCNet+FES (Ours)	Image	Mono	**50.69**	**36.17**	**31.97**	**48.40**	**38.59**	**32.69**
3DOP [6]	Image	Stereo	54.83	43.36	37.15	53.73	42.27	35.87
Li et al. [11]	Video	Stereo	58.52	46.17	43.97	48.51	37.13	34.54
LOCNet+FES+ **3D TNT+Multi. Opt (Ours)**	Video	Mono	**60.37**	**48.49**	**44.36**	**56.54**	**44.23**	**36.91**
Point-RCNN [12]	LiDAR	Pointcloud	66.89	54.91	47.13	62.76	49.13	42.43

Table 2. Performance of 3D localization methods on ApolloCar3D *val* set.

Method	Modality	2D Evaluation Metrics				3D Evaluation Metrics		
		AP_S	AP_M	AP_L	mAP	shape sim	dist. error (m)	ori. error (°)
LOCNet (Ours)	Image	11.3	12.6	29.7	13.3	0.88	1.13	6.7
LOCNet+FES (Ours)	Image	**11.6**	**13.8**	**33.1**	**14.1**	**0.91**	**1.09**	**6.1**

Table 3. Ablation on LOCNet on KITTI and ApolloCar3D *val* set.

Dataset	D-RPN	O-H	D-H	shape sim.	trans dist	rot dist
KITTI		✓	✓	0.93	2.06	6.6
	✓	✓		0.88	5.89	9.2
	✓	✓	✓	**0.94**	**0.98**	**4.3**
ApolloCar3D		✓	✓	0.84	3.67	10.4
	✓	✓		0.78	10.23	12.8
	✓	✓	✓	**0.88**	**1.13**	**6.7**

Table 4. Ablation on overall system on KITTI *val* set. (Average precision of bird eye's view and 3D boxes comparison.)

Module	AP_{BEV} (IoU>=0.5)			AP_{3D} (IoU>=0.5)			Time (ms)
	Easy	Mod	Hard	Easy	Mod	Hard	
L	42.86	30.43	26.35	36.06	25.44	24.19	143
L+T	46.89	35.11	28.43	44.22	30.48	27.92	407
L+S	50.69	36.17	31.97	48.40	38.59	32.69	197
L+S+T	57.16	44.72	38.29	54.34	42.88	35.94	457
L+S+T+M	**60.37**	**48.49**	**44.36**	**56.54**	**44.23**	**36.91**	795

within 0.98m and 4.3° for KITTI and 1.13m and 6.7° for ApolloCar3D respectively, which means it is able to exploit the implicit information that is shared between the orientation and distance heads.

Overall System. To see how different modules of our proposed system can contribute to the localization performance, we further conduct some experiments on KITTI validation set to highlight how they can impact the final results. We use L, S, T, M to represent LOCNet, single frame optimization with FES measure, TrackletNet Tracker, and multi-frame optimization respectively. As shown in Table 4, compared to LOCNet-only (L) results, the L+T improves both AP_{BEV} and AP_{3D} by a large margin, which shows that incorporating the temporal information from tracking is helpful to localization accuracy since it can deal with errors caused by occlusions and missing detections. By adding the single-frame FES optimization further brings an improvement of 10.12% and 12.4%, 8.02% respectively. Employing the multi-frame optimization further achieves the best AP_{3D} of 56.54% (↑ 2.2%), 44.23% (↑ 1.35%) and 36.91% (↑ 0.97%). The runtime of the system is also provided based on 8 Core i7-7700k CPUs (S, M) and 2 NVIDIA Titan Xp GPUs (L, T).

9. Conclusions

In this paper, we propose a monocular vision based autonomous driving framework to perform 3D detection, tracking and localization by effectively integrating all three tasks in a complementary manner. Our LOCNet and FES based single frame optimization provide accurate localization results, which are further refined with the help of the 3D TrackletNet Tracker to eventually achieve performance comparable to LiDAR-based localization methods. Quantitative experiments have shown that our system can achieve high accuracy in localization and outperform the state-of-the-art methods. Demonstrations on different datasets also show that our system is robust under different autonomous driving scenarios.

In the future works, we also plan to distinguish between static and dynamic observed vehicles through analysis on their distance and speed. Based on this, we will be able to better select background ORB features for camera ego-motion. Furthermore, the camera pose can also be integrated into our optimization framework such that the estimation for both camera and observed objects can also benefit from each other.

References

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

[2] Junaid Ahmed Ansari, Sarthak Sharma, Anshuman Majumdar, J Krishna Murthy, and K Madhava Krishna. The earth ain't flat: Monocular reconstruction of vehicles on steep and graded roads from a moving camera. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8404–8410. IEEE, 2018.

[3] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9287–9296, 2019.

[4] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teuliere, and Thierry Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2040–2049, 2017.

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[6] 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 *Advances in Neural Information Processing Systems*, pages 424–432, 2015.

[7] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[9] Andrew J Hanson. Visualizing quaternions. In *ACM SIGGRAPH 2005 Courses*, pages 1–es. 2005.

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

[11] 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.

[12] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*, pages 820–830, 2018.

[13] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7074–7082, 2017.

[14] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[15] J Krishna Murthy, Sarthak Sharma, and K Madhava Krishna. Shape priors for real-time monocular object localization in dynamic environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1768–1774. IEEE, 2017.

[16] Vlad Paunescu, Andretti Naiden, Gyeongmo Kim, Byeong-Moon Jeon, and Marius Leordeanu. Shift r-cnn: Deep monocular 3d object detection with closed-form geometric constraints. 2019.

[17] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[18] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[20] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[21] Samuel Scheidegger, Joachim Benjaminsson, Emil Rosenberg, Amrit Krishnan, and Karl Granström. Mono-camera 3d multi-object tracking using deep learning detections and pmbm filtering. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 433–440. IEEE, 2018.

[22] 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.

[23] Jianyong Sun, Qingfu Zhang, and Edward PK Tsang. De/eda: A new evolutionary algorithm for global optimization. *Information Sciences*, 169(3-4):249–262, 2005.

[24] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8797–8806, 2019.

[25] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

[26] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 482–490, 2019.

[27] Di Wu, Zhaoyong Zhuang, Canqun Xiang, Wenbin Zou, and Xia Li. 6d-vnet: End-to-end 6-dof vehicle pose estimation from monocular rgb images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[28] Fuzhen Zhang. Quaternions and matrices of quaternions. *Linear algebra and its applications*, 251:21–57, 1997.

[29] Yanting Zhang, Haotian Zhang, Gaoang Wang, Jie Yang, and Jenq-Neng Hwang. Bundle adjustment for monocular visual odometry based on detections of traffic signs. *IEEE transactions on vehicular technology*, 69(1):151–162, 2019.

[30] Zhaoxiang Zhang, Kaiqi Huang, Tieniu Tan, and Yunhong Wang. 3d model based vehicle tracking using gradient based fitness evaluation under particle filter framework. In *2010 20th International Conference on Pattern Recognition*, pages 1771–1774. IEEE, 2010.

[31] Zhaoxiang Zhang, Tieniu Tan, Kaiqi Huang, and Yunhong Wang. Three-dimensional deformable-model-based localization and recognition of road vehicles. *IEEE transactions on image processing*, 21(1):1–13, 2011.