IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 June 2024, accepted 17 July 2024, date of publication 5 August 2024, date of current version 16 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3438162

RESEARCH ARTICLE

Optimization-Based Monocular 3D Object Tracking via Combined Ellipsoid-Cuboid Representation

GYEONG CHAN KIM[1,2], (Graduate Student Member, IEEE), YOUNGSEOK JANG[2,3], (Graduate Student Member, IEEE), AND H. JIN KIM[1,2], (Member, IEEE)

[1]Department of Aerospace Engineering, Seoul National University, Gwanak-gu, Seoul 08826, South Korea
[2]Automation and Systems Research Institute (ASRI), Seoul National University, Gwanak-gu, Seoul 08826, South Korea
[3]Department of Mechanical and Aerospace Engineering, Seoul National University, Gwanak-gu, Seoul 08826, South Korea

Corresponding author: H. Jin Kim (hjinkim@snu.ac.kr)

This work was supported by Unmanned Vehicles Core Technology Research and Development Program through the National Research Foundation of Korea (NRF) and Unmanned Vehicle Advanced Research Center (UVARC) funded by the Ministry of Science and ICT, the Republic of Korea under Grant NRF-2020M3C1C1A010864.

ABSTRACT Monocular 3D object tracking is a challenging task because monocular image lacks depth information necessary for 3D scene understanding. Modern methods typically rely on deep learning to reconstruct 3D information from learned prior, which demands strenuous effort on acquiring ground-truth annotated data and does not generalize for various camera settings. We present a method to continuously track 3D location and orientation of the target object from a monocular image sequence from 2D instance segmentation methods. We reconstruct the structure and trajectory of the objects using factor graph optimization incorporating reprojection error of keypoint tracks, kinematic motion model and bounding box constraints. We propose a combined ellipsoid-cuboid object representation and bounding box constraint to model the object dimension. We evaluate our algorithm in simulation dataset generated using CARLA, and the result indicates that the method is robust to 2D bounding box error and the proposed object representation yields more accurate pose and size estimation compared to solely using either representation.

INDEX TERMS Graph optimization, monocular vision, 3D object tracking.

I. INTRODUCTION

Capability to understand surrounding environments is an essential requirement for autonomous vehicles and mobile robots. Specifically, detecting and tracking dynamic objects in 3D world is crucial for safe autonomous navigation, since it has a direct impact on downstream tasks such as collision avoidance and simultaneous localization and mapping (SLAM).

LiDAR is a popular sensor choice when it comes to 3D object tracking [1], [2], [3], since it offers highly accurate 3D point cloud data of the measurements of the surrounding environment. However, the cost, size, and weight of LiDAR sensors are recognized as expensive, and processing 3D point clouds data demands significant computing power. Camera is an alternative choice, which offers low cost and light weight solution that is widely applicable to variety of platforms. Stereo camera enables obtaining 3D information by disparity estimation, but the accuracy of depth information degrades as the target is far away compared to its baseline. RGBD camera is another sensor which provides depth measurement. However, reliability of depth measurement degrades under strong light source such as sunlight, hence it is inadequate for outdoor applications. Given this context, utilizing appearance cue to reconstruct 3D information is crucial despite the availability of depth measurements, which renders studying monocular 3D object tracking valuable for the progress of image-based methods.

To handle the absence of depth measurement which is crucial for reconstructing 3D geometry of objects, majority of the monocular image-based tracking approaches utilize deep learning methods. However, deep learning-based 3D object

The associate editor coordinating the review of this manuscript and approving it for publication was Andrea Bottino.

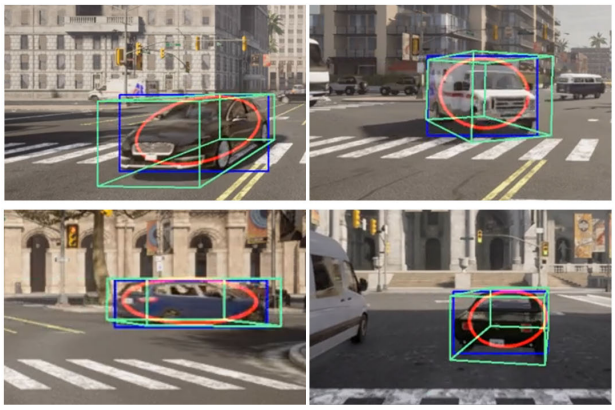


FIGURE 1. Illustrative images of objects' ground-truth 3D bounding box(colored green) and corresponding 3D ellipsoid(colored red) overlapped with 2D detection bounding box(colored blue). Image plane projection of these object representations does not fit the 2D bounding box most of the time. We model the 2D bounding box as weighted sum of the two 3D representations and estimate the weight parameters for each frame along with pose parameters.

tracking has few limitations that prevent it from widespread application. First, meticulous data processing procedure is required to generate accurate labels for training neural network. For example, ground-truth labels for Kitti3D [4] is obtained by processing 3D LiDAR pointclouds using special tools and projecting back to image plane with accurate extrinsic calibration between LiDAR and camera. Second, the performance of the 3D detector is known to be susceptible to domain gap between training and test data, such as different camera intrinsic/extrinsincs and placement on the platform [5], [6].

Considering aforementioned concerns, we present a monocular 3D object tracking framework solely relying on 2D instance segmentation. Not relying on neural network requires alternative method to recover 3D information. To this end, we adopt multiple view geometry technique [7] which is widely practiced in monocular visual odometry or SLAM methods. In particular, we track keypoints in region of interest(RoI) across multiple images, and jointly estimate their 3D coordinates and object poses by solving nonlinear least squares problem.

Still, sole reliance on keypoint tracks is insufficient for accurate estimation of scale and shape of the object, and requires us to handle the following ambiguities.

1) shape ambiguity : only part of objects' 3D geometry which is visible from the camera can be reconstructed.
2) pose ambiguity : The reconstructed points' 3D coordinates are invariant to SE(3) transformation, which makes the object pose ambiguous.
3) scale ambiguity : Absolute scale of the reconstructed shape and trajectory cannot be estimated without appropriate prior.

To handle shape and pose ambiguity, it is necessary to have an appropriate object representation and constraint to restrict the search space for size and pose of the object. Multiple types of object representation have been applied, from 3D cuboid [8], [9], semantic keypoints [10], [11], shape prior [12], [13] and 3D ellipsoid [14], [15], [16]. From these schemes, 3D cuboid and ellipsoid both fully encodes the size(width, height and longitude) and pose of the object with minimal number of parameters. Also, previous works have demonstrated that their 3D parameters can be recovered from 2D bounding boxes, both for 3D cuboid [9], [17], and Ellipsoid [14], [15]. This characteristic is especially desirable since we choose to not rely on deep neural network for 3D pose and shape estimation.

Most of these approaches apply the tight 2D bounding box model to constrain the object's parameters: the 2D detection bounding box tightly bounds the image plane projection of either 3D bounding box or ellipsoid. However, we observe that both of the assumption leads to erroneous estimation of shape parameters. An illustration of tight 2D bounding box model failing to properly model the shape of the object is illustrated in Figure 1. When 2D bounding box model is applied to the 3D ellipsoid the object size is overestimated, while when applied to 3D bounding box the size is underestimated.

Based on these observations, we choose to combine 3D cuboid and ellipsoid, and devise a method to constrain the shape of the object with 2D bounding boxes. We assume that a 2D detection bounding boxes' edge can be expressed as weighted average of corresponding edges of 2D bounding boxes each tightly enclosing the image plane projection of the ellipsoid and the 3D bounding box. We jointly estimate the weight parameter for each edge alongside point tracks, size and trajectory of the object.

To tackle the scale ambiguity, We assume that the motion of dynamic objects is restricted to a known supporting plane, e.g. a ground plane. This assumption is commonly accepted in both deep learning-based approaches [18], [19], and approaches that do not use deep learning for 3D reconstruction [17], [20]. We demonstrate that incorporating supporting plane assumption with our object representation leads to an accurate scale recovery of object trajectory.

Properly initializing the object pose solely based on geometric information is another challenging task. Nevertheless, it can have a significant impact on the performance of pose estimation after initialization. Previous methods has tackled this problem by utilizing multiple tangential planes from surface reconstruction [21] or appearance-based heuristics [17], [22]. In this paper, we propose a two-view initialization scheme for object pose and shape parameters by assuming a simple motion model which is widely applicable to many dynamic objects.

In summary, our contributions are as follows. We propose a monocular 3D object tracking method based on solving nonlinear optimization over sliding window of frames, incorporating multiple-view geometry techniques, planar motion assumption and bounding box constraints. To initially estimate the object pose and shape parameters, we devise a two-view initialization method which utilizes 2D bounding

box constraints on 3D geometric representations and simple motion model. Through experiments on simulation dataset, we show that the suggested tight 2D bounding box constraint on combined ellipsoid-cuboid representation leads to more accurate recovery of object size and pose parameters compared to either of the two representation used separately.

II. RELATED WORKS

A. MONOCULAR 3D OBJECT TRACKING

3D object tracking has been studied for various sensor settings, including LiDAR [2], [3], stereo camera [23] and monocular camera [24], [25]. In this section, we primarily focus on monocular 3D object tracking literature. The most prominent approach in 3D object tracking is the tracking-by-detection paradigm, where first a detector is employed to each frame, and association between object track and detection is established by utilizing multiple cues. Joint 3D detection and tracking [24] train a neural network to estimate 3D bounding box from 2D bounding boxes and association is performed utilizing depth ordering strategy. Quasi-dense tracking [25] leverages 3D trajectory prediction based on motion model and quasi-dense similarity learning for association. 3DOT [1] employs a 3D bounding box detector and associates the tracks based on 3D Kalman filter with simple motion model.

These methods rely on deep learning to estimate the object pose and shape parameters required for 3D tracking. Unlike their 2D counterparts, deep-learning-based 3D object detection methods have the following drawbacks which hinder its straightforward employment in practice. One problem is the difficulty in acquiring the ground-truth annotation. Since RGB image does not contain depth information, acquiring 3D object detection ground-truth involves careful processing of additional sensor data which straightforwardly yields depth measurements such as LiDAR [4], [26]. To relax this requirement, self-supervised learning methods have been invested [27], [28], although these methods still require LiDAR or depth measurement for training. In addition, it is reported that performance of the deep learning-based 3D object detectors is affected by discrepancy between sensor settings in the training dataset and during deployment [5], [6]. In this work, we step aside from relying on deep learning methods and instead utilize multiple view geometry to estimate 3D attributes of the target objects.

B. 3D OBJECT REPRESENTATIONS

Various object representations have been employed to estimate and track the position and orientation of objects in the scene. Methods utilizing object shape priors [29], [30] train a neural network to infer category-specific object shape models from images which are then utilized to provide depth supervision required for 3D object detection. Training an object shape prior often involves additional training with large amount of CAD data, which requires a significant amount of computing resources in both the training and inference phases. Another popular choice is the 3D cuboid [8], [9], [24], [25], in which the system attempts to estimate the 3D bounding box of the detected objects. One typical approach is to formulate the problem as a direct regression of 3D bounding box parameters from 2D image features within the object region of interest(ROI) [8], [24]. This approach when applied for monocular 3D object detection suffers from large search space and ambiguity, since appearance is the only cue available for 3D object detection.

It is generally accepted that leveraging additional constraints to regulate 3D object parameters is the key to overcome this shortcoming. Assuming 2D detection bounding box tightly encloses the 3D bounding box [8], [9], [17] is an assumption that is frequently made. Other approaches assume objects are lies on a ground plane to resolve pose ambiguity [18], [19]. In our approach, we also assume that the objects lie on a known ground plane, and exploits the relationship between 2D bounding box and projection of 3D bounding box. However, we observe that the tight 2D bounding box model often leads to erroneous estimation of object size due to discrepancy between the actual object boundary and its 3D bounding box.

3D ellipsoid representation is another object representation which enables parametrization of object size and pose with minimal number of variables. It is mainly invested in object SLAM literature, whose objective is to construct a lightweight object-centric map from sequence of sensor data. Early works [14], [31] demonstrated that it is feasible to fully specify a 3D ellipsoid given 2D bounding boxes from two views under the tight 2D bounding box assumption. In practice, lack of observations from distinct enough viewpoints result in degeneracy in the estimated quadric parameters. Several following researches have suggested methods to mitigate this issue and robustly estimate the ellipsoid parameters. Cao et al. [21] utilized a supporting plane and surface reconstruction to enforce multiple tangential plane constraints. EAO-SLAM [22] employed line segment detection and alignment method for robust estimation of object orientation.

In this work, we make use of both geometric representations to model the size and pose of an object. We figure out that tight 2D bounding box model for 3D bounding box yields an underestimation of the object size, while for 3D ellipsoid it leads to an overestimation of the object dimension. We combine both representation for more accurate size estimation.

C. OPTIMIZATION FOR 3D OBJECT TRACKING

Accuracy of initial estimate of object pose and size is often not satisfactory. Hence existing 3D object tracking methods take further refinement steps to improve their accuracy. References [1], [32], and [33] employ variants of Kalman filter to accurately track 3D objects' state. While filtering achieves low computational cost by only retaining estimate of the most recent frame, it lacks the capability to utilize long-term measurement history.

On the other hand, nonlinear optimization methods incorporate measurements from previous frames in the expense of higher computational cost. Studies in SLAM literature [34], [35] have consistently shown that nonlinear optimization methods outperform filtering methods in terms of estimation accuracy. Multiple researches have explored the application of nonlinear optimization for solving 3D object pose tracking problem. Reference [23] solves a dynamic object bundle adjustment problem to optimize 3D bounding box and point cloud. Subsequent work [36] employs a per-frame marginalization strategy and solves the optimization over a fixed-size sliding window, reducing repeated computation. ClusterVO [37] proposes a dual sliding window structure which consists of multiple past keyframes and recent frames. This structure is also utilized in the formulation of optimization problem in our approach.

A common practice in optimization-based object pose tracking is to restrict object poses to follow a certain motion model. SAMP [13] utilizes a motion model and ground plane prior to optimize the shape distance function representation of the target vehicle over the history of depth measurements. CubeSLAM [17] also assumes that the objects are constrained to move on a common ground plane. Our method also incorporates both the ground plane prior and motion model into the optimization problem to estimate target object's pose and size.

III. PROBLEM DESCRIPTION

In this article, we use the nomenclature presented in Table 1. We also note that if a vector or a transformation is stated without a specific coordinate frame, by default its reference coordinate frame is the ground-centered coordinate frame $\{\mathrm{g}\}$, whose $\mathbf{z}$ axis is parallel to the normal of the ground plane and zero is located on an arbitrary point on the ground plane. We assume that the target objects lie on a ground plane which parameters are known. We also presume that the camera transform ${}_{\mathrm{c}}\mathbf{T}_{\mathrm{g}}$ are known for each frame.

We formulate the problem of monocular 3D object tracking as follows. We assume that 2D instance segmentation of a single target of interest is given in the first frame. For input image at time k, 2D object detection result $\{D^{(1)}(k), D^{(2)}(k), \cdots, D^{(N_k)}(k)\}$ is provided by an off-the shelf instance-segmentation method, where each detection $D^{(i)}(k)$ is associated with a 2D bounding box $\boldsymbol{\beta}^{(i)}(k)$ and a segmentation mask $M^{(i)}(k)$. Out of these, we determine the appropriate object detection for the tracked object. Afterwards, utilizing the associated detection result, we estimate the object center position $\mathbf{t}_o(k) = [x_o(k), y_o(k), z_o(k)]^{\mathrm{T}}$, orientation ($\mathbf{R}_o(k)$) and size ($\mathbf{r} = [r_x, r_y, r_z]^{\mathrm{T}}$) of the object. Since we consider the object pose to be restricted by the ground plane, the number of variables that need to be estimated per frame is reduced from 6 to 3: We can set the $\mathbf{z}$ axis translation to be equivalent to the $\mathbf{z}$ axis radius of the object r_z, thus only regard the yaw angle $\psi_o(k)$ for orientation. We also mention that the object's speed $\mathbf{s}_o(k)$ and angular velocity $\omega_o(k)$ are estimated in the process as a byproduct, but it is not the primary focus. We define 2D state variables $\mathbf{X}_o(k) := \{{}_{\mathrm{g}}\mathbf{x}_o(k), \psi_o(k), \mathbf{s}_o(k), \omega_o(k)\}$ to denote the variables which are estimated per frame.

TABLE 1. Notation used in this article.

Symbol	Meaning
${}_{\mathrm{a}}\mathbf{T}_{\mathrm{b}}$	SE(3) transform from coordinate frame $\{\mathrm{b}\}$ to $\{\mathrm{a}\}$.
${}_{\mathrm{a}}\mathbf{R}_{\mathrm{b}}, {}_{\mathrm{a}}\mathbf{t}_{\mathrm{b}}$	Rotation and translation of the SE(3) transform ${}_{\mathrm{a}}\mathbf{T}_{\mathrm{b}}$.
$\{\mathrm{g}\}, \{\mathrm{c}\}, \{\mathrm{o}\},$	Ground-centered coordinate frame, Camera Coordinate frame and object coordinate frame.
$\mathbf{r} := [r_x, r_y, r_z]$	The radius of 3D ellipsoid representation along each axis
$\mathbf{X}_o$	2D object state, which comprise of 2D position, heading angle, speed and angular velocity
${}_{\mathrm{a}}\mathbf{Q}^*_{\mathrm{o}}$	Dual quadric matrix of the 3D ellipsoid, expressed in coordinate frame $\{\mathrm{a}\}$.
$\mathbf{P}_{\mathrm{a}}$	The camera projection matrix from the coordinate frame a.
$proj(\mathbf{P}, \mathbf{x})$	Projection of a 3D point $\mathbf{x}$ defined by the projection matrix $\mathbf{P}$.
$\boldsymbol{\beta} := [\beta_l, \beta_t, \beta_r, \beta_b]$	2D bounding box in image plane, defined by its $\mathbf{x}$ axis extrema(β_l, β_r) and $\mathbf{y}$ axis extrema(β_t, β_b)
u, v	Horizontal / vertical axis of the image plane
$\mathbf{u}_i$	2D observation of a keypoint specified by index i
$\mathrm{P} = \{\mathbf{p}_1, \cdots \mathbf{p}_{N_p}\}$	3D position of the keypoint tracks in object-centric coordinate frame
$S = \mathrm{P} \cup \{\mathbf{r}\}$	The structure variables (size of the object, and the 3d coordinates of point tracks)
W_r, W_k	recent frame window / keyframe window
$[\cdot](k)$	The value of symbol $[\cdot]$ at time(frame) k.

IV. METHOD

An overview of our method is presented in Figure 2. We maintain a sliding window of frames which consists of N_r most recent frames (termed recent frame window W_r) and N_k latest keyframes (termed keyframe window W_k). Given an input image and instance segmentation result, we first perform tracking in 2D domain using grid-based color histogram matching (Sec. IV-B), followed by extraction and tracking of keypoints [38] from the associated 2D segmentation mask. If the tracked object's 3D pose and shape are not initialized, the object initialization algorithm (Sec. IV-C) attempt to estimate the 3D size and pose parameters from two view reconstruction result. Otherwise, object pose is first optimized with motion only optimization (Sec. IV-D).

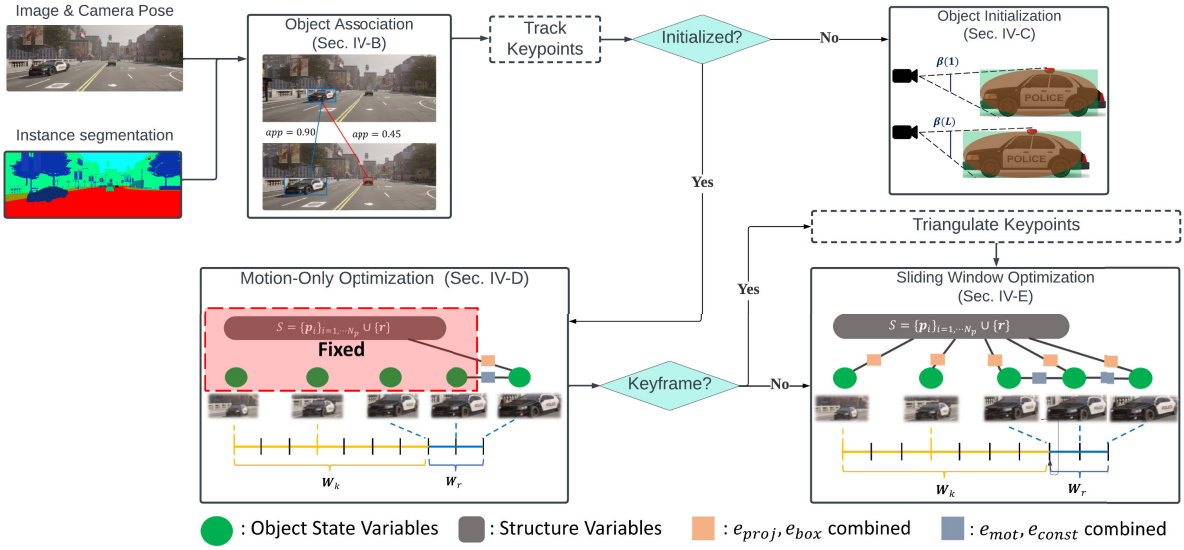


FIGURE 2. Flowchart of the proposed system. We represent the target object's shape as combined ellipsoid-cuboid form (Sec IV-A). Given a sequence of monocular images with known camera pose and instance segmentation, the object association module tracks the object in image space using appearance cues (Sec IV-B). If a previous estimate of the object's state is available, we perform motion-only optimization (Sec IV-D). Otherwise, we initialize the variables utilizing ground plane and motion constraints (Sec IV-C). Finally, all variables are optimized within the sliding window optimization framework (Sec IV-E).

Afterwards, a sliding window optimization is performed to jointly refine the structure variables (3D position of keypoint tracks and size of the object) and the state variables (pose and velocity of the object) at each frame in the sliding window. In the following subsections, we explain each module of the proposed method in detail.

A. COMBINED ELLIPSOID-CUBOID OBJECT REPRESENTATION

We begin with an explanation of the combined ellipsoid-cuboid object representation and the tight 2D bounding box model utilized to estimate the target object's size and pose. The 3D cuboid representation refers to the object's 3D bounding box. Since we only consider the objects lying on a known ground plane, the 3D bounding box can be fully defined using six free variables as mentioned in III. We can also consider the 3D ellipsoid with the same position and orientation, and radius equal to the half of each axis length.

The tight 2D bounding box model for each 3D representation assumes that the 2D bounding box tightly encloses image plane projection of the 3D representation. For 3D cuboid, four edges of the 2D bounding box $\boldsymbol{\beta}^{\text{cub}}(k) = [\beta_l^{\text{cub}}(k), \beta_t^{\text{cub}}(k), \beta_r^{\text{cub}}(k), \beta_b^{\text{cub}}(k)]$ can be acquired by reasoning over the image plane projection of the 8 corners of the cuboid $\{\mathbf{c}_1, \mathbf{c}_2, \cdots \mathbf{c}_8\}$.

$$\beta_l^{\text{cub}}(k) = \arg\min_u proj(\mathbf{P}_o(k), \mathbf{c}_i)$$
$$\beta_t^{\text{cub}}(k) = \arg\min_v proj(\mathbf{P}_o(k), \mathbf{c}_i)$$
$$\beta_r^{\text{cub}}(k) = \arg\max_u proj(\mathbf{P}_o(k), \mathbf{c}_i)$$
$$\beta_b^{\text{cub}}(k) = \arg\max_v proj(\mathbf{P}_o(k), \mathbf{c}_i), \quad i = 1, \cdots, 8 \tag{1}$$

Meanwhile, the image plane projection of the ellipsoid forms a 2D ellipse. The dual conic matrix of 2D ellipse $\mathbf{C}_{\text{o}}^*$ and the dual quadric matrix of 3D ellipsoid $\mathbf{Q}_{\text{o}}^*$ are associated by the projection matrix $\mathbf{P}$:

$$\mathbf{C}_{\text{o}}^*(k) = \mathbf{P}_{\text{a}}(k)_{\text{a}}\mathbf{Q}_{\text{o}}^*(k)\mathbf{P}_{\text{a}}(k)^{\text{T}} \tag{2}$$

where the subscript {a} can be an arbitrary coordinate frame. Mathematical details about the dual conic and dual quadric representation can be referred to [7]. For an ellipse, the maximum/minimum of u, v coordinate values which define the 2D bounding box edges $\boldsymbol{\beta}^{\text{ell}}(k) = [\beta_l^{\text{ell}}(k), \beta_t^{\text{ell}}(k), \beta_r^{\text{ell}}(k), \beta_b^{\text{ell}}(k)]$ can be evaluated in an analytic form [14].

Instead of regarding a single representation for the 2D bounding box constraint, we model each edge of the 2D detection bounding box to be weighted average of the corresponding edge of $\boldsymbol{\beta}^{ell}$ and $\boldsymbol{\beta}^{cub}$.

$$\beta_i(k) = \lambda_i \beta_i^{ell}(k) + (1 - \lambda_i)\beta_i^{cub}(k), \quad i = l, t, r, b \tag{3}$$

The proposed 2D bounding box constraint is visualized in Figure 3. Alongside the object size and poses in the sliding window, we jointly estimate the values of optimal weight parameters that minimize the error between the model and the observation.

B. OBJECT ASSOCIATION

Given an input image at time k and 2D detection results $\{D^{(1)}(k), D^{(2)}(k), \cdots, D^{(N_k)}(k)\}$, the object association module aims to find the detection result $D^*(k)$ with the highest

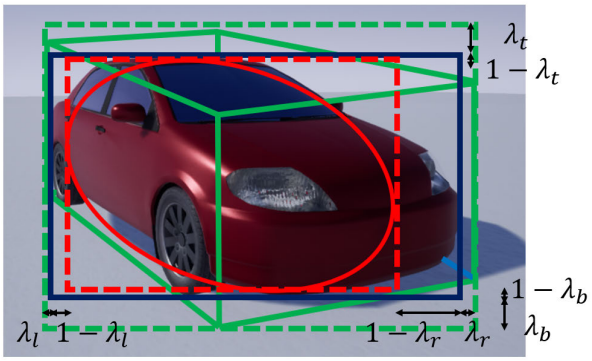


FIGURE 3. Visualization of the proposed 2D bounding box constraint for combined ellipsoid-cuboid representation. We jointly estimate the size of the object alongside the edge-wise weight $\lambda = [\lambda_l, \lambda_t, \lambda_r, \lambda_b]$ for recent observations in the sliding window.

association score for the tracked object. Here, we only take account for the detection results with class label equal to the tracked object. The association score comprises of two measures: Appearance similarity and Spatial affinity.

To evaluate appearance similarity score, we employ a method based on color histogram matching similar to the one proposed in Karunasekera et al. [39]. Figure 4 represents the evaluation process appearance similarity. In detail, the image is transformed into HSV color space, and for each 2D detection result the corresponding region of interest is uniformly divided into $N_g \times N_g$ grids. Histograms with N_b bins for each pair of hue (H) and saturation (S) values are calculated on each grid. Then the appearance similarity between two detection results D^1 and D^2 is defined as:

$$app(D^1, D^2) = \frac{\sum_n vis(D_n^1) \cdot vis(D_n^2) \cdot corr(H_n^1, H_n^2)}{\sum_n vis(D_n^1) \cdot vis(D_n^2)} \quad (4)$$

where H_n^i and $vis(D_n^i)$ denote the histogram value of nth grid for detection D^i and visibility score for the grid respectively. Here, the visibility score is simply computed as the ratio of pixels which belong to the instance segmentation mask M^i. We evaluate the similarity of each 2D detection $D^{(i)}(k)$ against all 2D detection results associated with the tracked object within the recent frame window. The final appearance similarity score is selected as the maximum among the computed similarity values.

Spatial affinity score evaluates how closely the predicted object's 2D bounding box matches the detection bounding box. We choose to evaluate the Intersection over Union (IoU) against the predicted 2D bounding box to measure the spatial affinity. For 2D bounding box prediction, two cases should be taken into account.

Case 1 is when the object size and pose estimation for the last frame are available. In this case, we predict the pose of the object at the next frame by assuming constant velocity and angular velocity from the last two frames. We use the weight parameter associated to the latest frame, and take the weighted average of 2D bounding boxes of the projected 3D ellipsoid and cuboid to output the predicted bounding box.

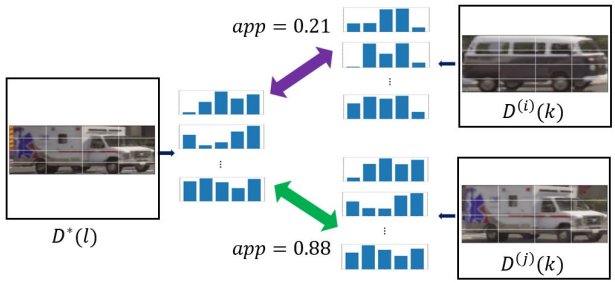


FIGURE 4. Simplified illustration of the appearance similarity measurement scheme.

Case 2 is when the object track does not contain 3D information yet. This occurs due to our framework requiring at least two frames for object's pose and size estimation. We roughly predict the 2D bounding box as follows: We unproject 2 bottom corners of the bounding box from the last frame so they lie on the ground plane. The two top corners are also unprojected to 3D assuming that they respectively lie on the line vertical to the plane and crossing bottom corner. We reproject the 3D points onto current frame, and take the extrema of u, v coordinates to obtain the 2D bounding box.

The final association score is expressed as a weighted sum of the appearance similarity score and the spatial affinity score. Regarding the unreliability of spatial affinity evaluation in case 2, we apply a slightly smaller weight for spatial affinity score in this case.

C. OBJECT INITIALIZATION USING ELLIPSOID REPRESENTATION

In this subsection, we explain the object initialization module in detail. Given first few frames and 2D detections associated with the object $\{D^*(1), D^*(2), \cdots, D^*(L)\}$, the initialization module attempts to estimate the dimension $\mathbf{r}$ and state history of the object $\{\mathbf{X}(1), \mathbf{X}(2), \cdots, \mathbf{X}(L)\}$. For the first (1st) and the last (Lth) frame in the 2D track history, the relative pose between the frames with respect to the camera frame can be recovered by epipolar geometry. specifically, the epipolar matrix acquired by two view geometry method [7] is related to the object-to-camera transform at each frame as follows.

$$\begin{aligned} E_{L1} &= [\bar{\mathbf{t}}_{L1}]_\times \mathbf{R}_{L1} \\ \mathbf{T}_{L1} &= \{\mathbf{R}_{L1}, s\bar{\mathbf{t}}_{L1}\} := {}_{\mathrm{c}}\mathbf{T}_{\mathrm{o}}(L)\,{}_{\mathrm{o}}\mathbf{T}_{\mathrm{c}}(1) \\ &= {}_{\mathrm{c}}\mathbf{T}_{\mathrm{g}}(L)\,{}_{\mathrm{g}}\mathbf{T}_{\mathrm{o}}(L)\,{}_{\mathrm{o}}\mathbf{T}_{\mathrm{g}}(1)\,{}_{\mathrm{g}}\mathbf{T}_{\mathrm{c}}(1) \end{aligned} \quad (5)$$

where s is a scale variable that arises from scale ambiguity. Since rotation is scale-invariant, we can compute the orientation difference between the two frames.

$$\begin{aligned} {}_{\mathrm{g}}\mathbf{R}_{\mathrm{c}}(L)\mathbf{R}_{L1}\,{}_{\mathrm{c}}\mathbf{R}_{\mathrm{g}}(1) &= {}_{\mathrm{g}}\mathbf{R}_{\mathrm{o}}(L)\,{}_{\mathrm{o}}\mathbf{R}_{\mathrm{g}}(1) \\ &= \mathrm{Rot}_z(\psi_o(L) - \psi_o(1)) \end{aligned} \quad (6)$$

Let us define $\Delta\psi := \psi_o(L) - \psi_o(1)$. We would like to identify the values of $\psi_o(1)$ and $\psi_o(L)$ for given translation ${}_{\mathrm{g}}\mathbf{x}(1)$ and ${}_{\mathrm{g}}\mathbf{x}(L)$ by utilizing a simple motion model which

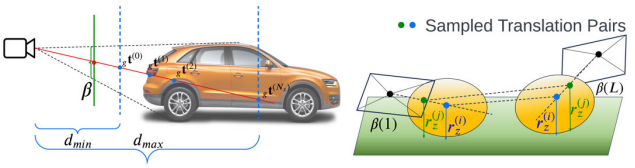


FIGURE 5. Visualization of the two view Translation sampling strategy. (Left) : We uniformly sample along the ray crossing the center of the first frame's 2D bounding box $\beta(1)$. (right) Corresponding translation at frame L (denoted as ${}_{\mathrm{g}}\mathbf{t}^{(i)}(L)$) is determined by $r_z^{(i)}$ and the center of the 2D bounding box $\beta(L)$.

couples the object's orientation with displacement between the frames. Specifically, we apply the kinematic bicycle model of the following form.

$$\begin{aligned} \psi_o(L) &= \psi_o(1) + \Delta\psi \\ {}_{\mathrm{g}}\mathbf{x}_o(L) &= {}_{\mathrm{g}}\mathbf{x}_o(1) + \begin{bmatrix} \cos\phi_{L,1} \\ \sin\phi_{L,1} \end{bmatrix} \Delta d \end{aligned} \tag{7}$$

where $\phi_{L,1} = (\psi_o(1) + \psi_o(L))/2$ and Δd denotes the size of displacement between two frames. If ${}_{\mathrm{g}}\mathbf{x}(1)$ and ${}_{\mathrm{g}}\mathbf{x}(L)$ are given, we can calculate $\phi_{L,1}$, and the values of $\psi_o(1)$, $\psi_o(L)$ are obtained consequently.

We generate N_s samples of translation pair $\{{}_{\mathrm{g}}\mathbf{t}(1), {}_{\mathrm{g}}\mathbf{t}(L)\}$. Here, we assume that the center of the 3D bounding box lies on the ray crossing the center of 2D detection bounding box. We take N_s translation samples, evenly spaced along the ray and ranging in distance from $d_{\min}$ to $d_{\max}$. The maximum distance along the ray is constrained the ground plane. We set the minimum height for the object center to lie above the ground plane to limit the search space. For each sample ${}_{\mathrm{g}}\mathbf{t}^{(i)}(1)$, corresponding height of the object center $r_z^{(i)}$ is determined, and accordingly the translation at frame L is derived from the 2D bounding box center and its $\mathbf{z}$ axis coordinate value. The sampling strategy is visualized in Figure 5.

Now since the values of ${}_{\mathrm{g}}\mathbf{x}^i(1)$, ${}_{\mathrm{g}}\mathbf{x}^{(i)}(L)$ and r_z are available, $\psi_o^i(1)$ and $\psi_o^i(L)$ are also derived from the motion model. Now we estimate the best values of r_x, r_y for each sample. Here, we utilize the 3D ellipsoid model to initially estimate r_x and r_y. We assume that the 2D bounding box is tangential to the image plane projection of the 3D ellipsoid. An equivalent statment of this assumption is that 3D ellipsoid is tangential to the plane defined by the camera center and unprojected line of each 2D bounding box edges. The dual-quadric representation allows us to formulate this constraint as a single equation:

$$\boldsymbol{\pi}_i^{\mathrm{T}} \boldsymbol{Q}_o^* \boldsymbol{\pi}_i^{\mathrm{T}} = 0, \quad i = l, t, b, r \tag{8}$$

where $\boldsymbol{\pi}_i$ is the plane parameter of the plane corresponding to 2D bounding box edge β_i. In the case of 3D ellipsoid, the dual-quadric matrix $\boldsymbol{Q}_o^*$ can be expressed as follows:

$$\begin{aligned} \boldsymbol{Q}_o^* &= \begin{bmatrix} \mathbf{R}_o \mathbf{D}_o \mathbf{R}_o^{\mathrm{T}} - \mathbf{t}_o \mathbf{t}_o^{\mathrm{T}} & -\mathbf{t}_o \\ -\mathbf{t}_o^{\mathrm{T}} & -1 \end{bmatrix}. \\ \mathbf{D} &= \mathrm{Diag}([r_x^2, r_y^2, r_z^2]) \end{aligned} \tag{9}$$

When the ground plane assumption is applied, the expression in equation (9) can be further simplified. Substituting this simplified equation to equation (8) yields equation (10).

$$\begin{aligned} \boldsymbol{\pi}_i^{\mathrm{T}} \begin{bmatrix} \mathbf{C} & \mathbf{00} & \\ \mathbf{0}^{\mathrm{T}} & r_z^2 & 0 \\ \mathbf{0}^{\mathrm{T}} & 0 & -1 \end{bmatrix} \boldsymbol{\pi}_i &= \boldsymbol{\pi}_i^{\mathrm{T}} \mathbf{S} \boldsymbol{\pi}_i, \\ \mathbf{S} &:= \begin{bmatrix} \mathbf{t}_o \mathbf{t}_o^{\mathrm{T}} & \mathbf{t}_o \\ \mathbf{t}_o^{\mathrm{T}} & 0 \end{bmatrix} \end{aligned} \tag{10}$$

Let $\boldsymbol{\pi}_i := [\mathbf{n}_i, a_i] = [\alpha_i, \beta_i, \gamma_i, a_i]$. Equation (10) can be reduced to the following equation:

$$\begin{aligned} \begin{bmatrix} \alpha_i^2 & 2\alpha_i\beta_i & \beta_i^2 \end{bmatrix} \begin{bmatrix} \mathbf{C}_{11} \\ \mathbf{C}_{12} \\ \mathbf{C}_{22} \end{bmatrix} &= -\gamma_i^2 r_z^2 + (a_i + \mathbf{n}_i^{\mathrm{T}} \mathbf{t})^2 \\ i &= l, t, b, r \end{aligned} \tag{11}$$

where $\mathbf{C}_{ij}$ denotes the entry in (i, j) of the matrix $\mathbf{C}$. Since each entry in $\mathbf{C}$ is a linear combination of r_x^2, r_y^2 for a fixed ψ. Having 2 bounding boxes at time 1 and L offers 8 linear equations. We additionally apply a simple prior η_x, η_y on the ratio of r_x, r_y and r_z.

$$\begin{bmatrix} 1/r_z^2 & 0 \\ 0 & 1/r_z^2 \end{bmatrix} \begin{bmatrix} r_x^2 \\ r_y^2 \end{bmatrix} = \begin{bmatrix} \eta_x^2 \\ \eta_y^2 \end{bmatrix} \tag{12}$$

We obtain r_x and r_y for each sample $\{{}_{\mathrm{g}}\mathbf{x}^i(1), {}_{\mathrm{g}}\mathbf{x}^{(i)}(L)\}$ by solving the linear system. We measure each sample's suitability with sum of two measures. The first is L1-norm of the residual of the linear system given by equation (11) and (12). The second is the compliance of the sampled translation with the epipolar constraint written at equation (5). Rearranging equation (5) yields the following constraint,

$$s_{\mathrm{o}} \mathbf{R}_{\mathrm{c}}(L) \bar{\mathbf{t}}_{L1} = {}_{\mathrm{o}}\mathbf{t}_{\mathrm{g}}(1) - {}_{\mathrm{o}}\mathbf{t}_{\mathrm{g}}(L) - \mathbf{V} \tag{13}$$

where $\mathbf{V}$ is a variable that is independent of the object's position. We evaluate the orientation difference between the left and right hand sides of the equation. The sample with the lowest value for the weighted sum of these two metrics is selected.

We further refine the two view estimation using nonlinear optimization to obtain the final initialization result. We incorporate reprojection error of the point tracks, motion model and bounding box model to formulate the nonlinear optimization problem.

$$\begin{aligned} \{\mathbf{X}_o(1), \mathbf{X}_o(L), \mathrm{S}\} = \sum_{k=1,L} [&\rho_{box}(||e_{box}(k)||^2_{\Sigma_{box}}) \\ &+ \sum_i \rho_{proj}(||e_{proj}(k,i)||^2_{\Sigma_{proj}}] \\ &+ \rho_{mot}(||e_{mot}(1,L)||^2_{}) \end{aligned} \tag{14}$$

where $\rho_{(\cdot)}$ represents the Huber norm with distinct thresholds for individual loss terms. The bounding box error e_{box} and the motion model error e_{mot} are defined in equation (15). Since

the weight parameters are yet to be estimated, we presume equal weights of 0.5 for both cuboid and ellipsoid.

$$e_{box}(\mathbf{X}_o(k), \mathbf{r}; \boldsymbol{\beta}(k)) = \frac{1}{2}(\beta_i^{ell}(\mathbf{X}, \mathbf{r}) + \beta_i^{cub}(\mathbf{X}, \mathbf{r})) - \beta_i(k)$$
$$e_{mot}(\mathbf{X}_o(j), \mathbf{X}_o k) = \frac{(\psi_o(k) + \psi_o(j))}{2} - \arctan(\frac{{}_{\mathrm{g}}y_o(k) - {}_{\mathrm{g}}y_o(j)}{{}_{\mathrm{g}}x_o(k) - {}_{\mathrm{g}}x_o(j)}) \quad (15)$$

D. MOTION-ONLY OPTIMIZATION AND KEYFRAME SELECTION

Now suppose that the 3D size and pose of the object for the latest frames are available. Given the input image at time k and the selected 2D detection $D^*(k)$, motion-only optimization is performed to provide an initial estimation of the current object state $\overline{\mathbf{X}}_o(k) = \{{}_{\mathrm{g}}\overline{\mathbf{x}}_o(k), \overline{\psi}_o(k), \bar{\mathbf{s}}_o(k), \overline{\omega}(k)\}$. First, we perform state prediction by applying the kinematic bicycle model (7) on the last state estimate $\hat{\mathbf{X}}_o(k-1)$. Afterwards, 2D position ${}_{\mathrm{g}}\overline{\mathbf{x}}_o(k)$ and the orientation $\overline{\psi}_o(k)$ is further optimized by solving a nonlinear optimization problem. Specifically, we compute the 2D state which minimizes the reprojection error of point tracks and the bounding box error.

$$\{{}_{\mathrm{g}}\overline{\mathbf{x}}_o^*(k), \overline{\psi}_o^*(k)\} = \arg\min_{\mathbf{x},\psi} \rho_{box}(||e_{box}(\mathbf{X}, \mathbf{r}; \boldsymbol{\beta}(k))||^2_{\Sigma_{box}}) + \sum_i \rho_{proj}(||e_{proj}(\mathbf{X}, \mathbf{r}, {}_{\mathrm{o}}\mathbf{p}_i; \mathbf{u}_i)||^2_{\Sigma_{proj}}) \quad (16)$$

Here, we use the weight parameter associated to the latest frame to evaluate e_{box}.

After tracking is performed, we extract new keypoints, and determine whether the current frame should be selected as a new keyframe. Our keyframe selection strategy is similar to that of keyframe-based SLAM systems [34], [40]. The current frame is tested for the following conditions, and it is selected as a new keyframe if it satisfies at least one of them.

1) Under 70% of the keypoints in the current frame are observed in the last keyframe.
2) More than 0.4 seconds have elapsed since latest keyframe insertion.
3) Displacement or rotation exceeding a predefined threshold (2.0 m/10°) on the estimated object-to-camera transformation ${}_{\mathrm{c}}\mathbf{T}_{\mathrm{o}}(k) = \{{}_{\mathrm{c}}\mathbf{R}_{\mathrm{o}}(k), {}_{\mathrm{c}}\mathbf{t}_{\mathrm{o}}(k)\}$ since the last keyframe.

If the frame is selected as a keyframe, we attempt 3D reconstruction on the keypoint tracks which are not yet triangulated. A 3D reconstruction ${}_{\mathrm{o}}\mathbf{p}_i$ is considered failure if the reprojection error is larger than a certain threshold or ${}_{\mathrm{o}}\mathbf{p}_i$ is too large compared to the object size. If more than 50% of the 3D reconstruction trials result in failure, keyframe insertion is cancelled.

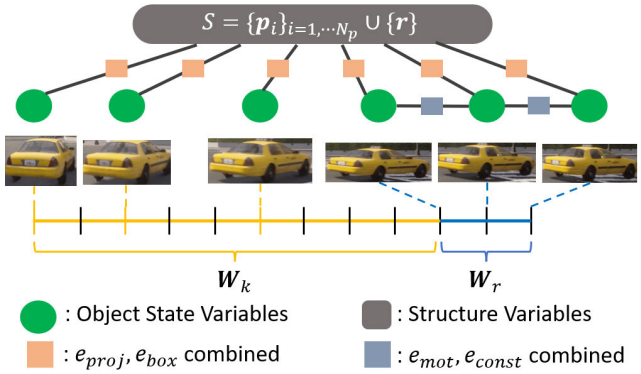


FIGURE 6. Factor Graph representation of the sliding window optimization problem. The sliding window comprises of the recent frame window W_r and the keyframe window W_k. We optimize the state variables and the structure variables regarding multiple error functions presented in the figure.

E. SLIDING WINDOW OPTIMIZATION

After the motion-only optimization, we optimize pose of the object for the frames within the sliding window alongside the structure variables $\mathrm{S} = \{\mathbf{p}_1 \cdots, \mathbf{p}_{N_p}\} \cup \{\mathbf{r}\}$. We apply sliding window formulation similar to that of ClusterVO [37], where the frames are managed in two tracks. The recent frame window W_r consists of the most recent N_r frames, which enables the tracker to maintain enough observations to capture the recent motion pattern and track the 2D appearance status. The keyframe window W_k maintains N_k latest keyframes ahead of the recent frame window, which provides multiple observations across diverse viewpoints. This allows for more stable optimization of the structure variables. We heuristically select $N_r = 3$ and $N_k = 7$.

The sliding window optimization involves state variables over all frames within the window, size and the point tracks. In addition to all the constraints introduced beforehand, we include constant speed error $e_{const}(j, j+1) = \mathbf{s}_o(j+1) - \mathbf{s}_o(j)$ for the recent frame window. For more stable tracking performance, we fix the state of the object at keyframes which are old enough. The entire optimization is configured as a factor graph optimization problem. We implement the optimization algorithm based on GTSAM [41] library. Figure 6 illustrates the sliding window formulation used in the proposed method.

Since the observed 2D boundary of an object alters by viewpoint change, presuming uniform weight parameters for all observation is not a reasonable approach. Meanwhile, using separate weight parameters for each frame leads to unstable estimation of object poses and shape due to high degree of freedom in variable space, and makes the framework susceptible to noise in object detection results. We set the same weights for the recent frame window, since object viewpoint does not change significantly among consecutive observations. We designate this weight parameter as $\boldsymbol{\lambda}^r = [\lambda_l^r, \lambda_t^r, \lambda_r^r, \lambda_b^r]$. For keyframes, we maintain the value of $\boldsymbol{\lambda}^r$

TABLE 2. Object tracking performance with different 3D object representations.

Representation	Ellipsoid				Cuboid				Combined Ellipsoid-Cuboid			
seq / obj. id	S	P	$\Delta\mathbf{t}$(m)	$\Delta\psi(^\circ)$	S	P	$\Delta\mathbf{t}$(m)	$\Delta\psi(^\circ)$	S	P	$\Delta\mathbf{t}$(m)	$\Delta\psi(^\circ)$
1/1	**0.607**	**0.799**	**0.401**	3.104	0.475	0.771	0.459	**2.439**	0.541	0.719	0.563	3.127
1/3	0.503	0.403	1.195	8.625	**0.555**	**0.712**	**0.575**	**6.791**	0.549	0.498	1.004	7.186
1/4	0.596	0.592	0.815	1.845	**0.629**	**0.781**	**0.437**	2.204	0.592	0.584	0.831	**0.431**
1/5	0.613	0.626	0.749	1.327	0.617	**0.731**	**0.539**	1.718	**0.622**	0.603	0.794	**0.693**
1/6	**0.542**	**0.381**	**1.238**	3.228	0.448	0.239	1.522	**2.858**	0.510	0.340	1.321	4.225
1/9	0.448	0.622	0.757	13.046	0.380	0.557	0.887	**7.241**	**0.579**	**0.700**	**0.600**	12.312
1/10	**0.597**	**0.663**	**0.673**	**8.157**	0.556	0.521	1.259	17.917	0.504	0.352	1.296	8.510
2/3	0.675	0.710	0.581	2.806	0.507	0.451	1.097	1.816	**0.778**	**0.798**	**0.403**	**0.934**
2/6	0.524	0.220	1.570	4.046	0.374	0.000	3.001	3.443	**0.594**	**0.343**	**1.314**	**0.993**
2/7	**0.451**	0.488	1.025	37.399	0.379	**0.693**	**0.614**	53.910	0.430	0.450	1.099	**26.789**
2/8	**0.627**	**0.783**	**0.434**	3.415	0.439	0.670	0.661	3.188	0.590	0.768	0.464	**2.641**
2/9	0.677	0.788	0.423	2.707	0.535	0.711	0.577	**1.648**	**0.775**	**0.871**	**0.257**	1.718
2/10	0.529	0.611	0.778	4.200	0.507	0.396	1.208	**1.970**	**0.591**	**0.743**	**0.515**	1.997
2/12	**0.542**	**0.718**	**0.565**	9.658	0.454	0.667	0.665	8.298	0.487	0.677	0.647	**7.726**
2/13	0.344	0.419	1.163	3.043	0.521	0.693	0.615	**3.030**	**0.761**	**0.852**	**0.296**	3.764
2/15	**0.758**	**0.855**	**0.289**	4.600	0.273	0.462	1.076	**4.183**	0.551	0.712	0.577	5.273
Mean	0.564	0.605	0.791	6.950	0.478	0.566	0.949	7.666	**0.591**	**0.626**	**0.749**	**5.520**

from the last time when the keyframe was part of the recent frame window.

V. RESULTS AND DISCUSSION

In this section, we examine the performance of the proposed method. We use our dataset generated with CARLA [42] simulator for experiment. The dataset comprises of RGB images of size 1280 × 400 captured from a simulated camera equipped on a vehicle in a traffic scenario, and the ground-truth instance segmentation mask for corresponding RGB image. Ground-truth histories of vehicle poses over time and the dimension of the vehicle are also accessible, thus utilized for performance evaluation. Since our algorithm assumes that the target object is not occluded, we evaluate the algorithm over sequences of observations where the target vehicle is not occluded by other objects.

A. EVALUATION METRIC

We choose 4 evaluation metrics commonly used in single object tracking. The first two metrics are **Success** and **Precision** metrics defined in [43]. Success (S) is defined as average overlap between the predicted and ground-truth bounding boxes, and precision (P) is defined as the Area Under the Curve (AUC) with distance threshold ranging from 0m to 2m. For evaluating Intersection Over Union (IoU) for bounding boxes, we choose to use bird-eye-view IoU since our target objects lie on a common ground plane. The other two are mean translation error ($\Delta\mathbf{t}$) and mean rotation error ($\Delta\psi$).

B. 3D REPRESENTATION COMPARISON

In this subsection, we demonstrate the performance of the proposed combined ellipsoid-cuboid object representation. To this end, we compare the proposed representation against two baseline representations: 1) ellipsoid-only representation, which is utilized in quadric-based object SLAM methods [14], [21], and 2) cuboid-only representation, which is used in Cube SLAM [17] or Stereo 3D object tracking [23]. In detail, we re-formulate the bounding box error function (ρ_{box}) for each 3D representation for the initialization refinement, motion-only optimization and the sliding window optimization phase.

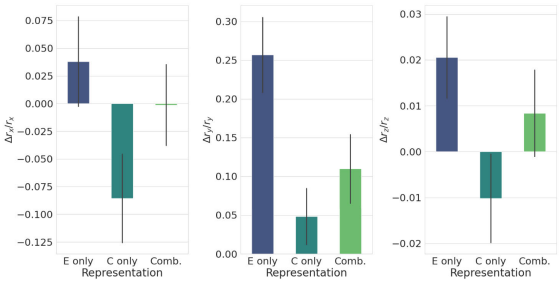


FIGURE 7. Average shape estimation error along 3 radial axis evaluated on each 3D representation. [E only] refers to "Ellipsoid-only" method and [C only] refers to "Cuboid-only" method.

We report results on 16 vehicle tracks for each representation in Table 2. While the best performing representation varies depending on the specific sequence, the proposed combined representation leads to the best overall performance.

Figure 8 shows snapshots of ground-truth and estimated 3D bounding box drawn on the image plane for each representation. We can recognize that the ellipsoid-only representation leads to overestimation of object size, while the cuboid-only representation leads to underestimation of object size. The bargraph in Figure 7 displays the average size estimation error for radius along each axis of object canonical frame. The size estimation error for each axis is defined as $e(r) = (\hat{r} - r)/r$ where $r, \hat{r}$ stand for the ground-truth and estimated axial radius respectively. The results support our observation about object size estimation bias for each representation, and validate that our combined

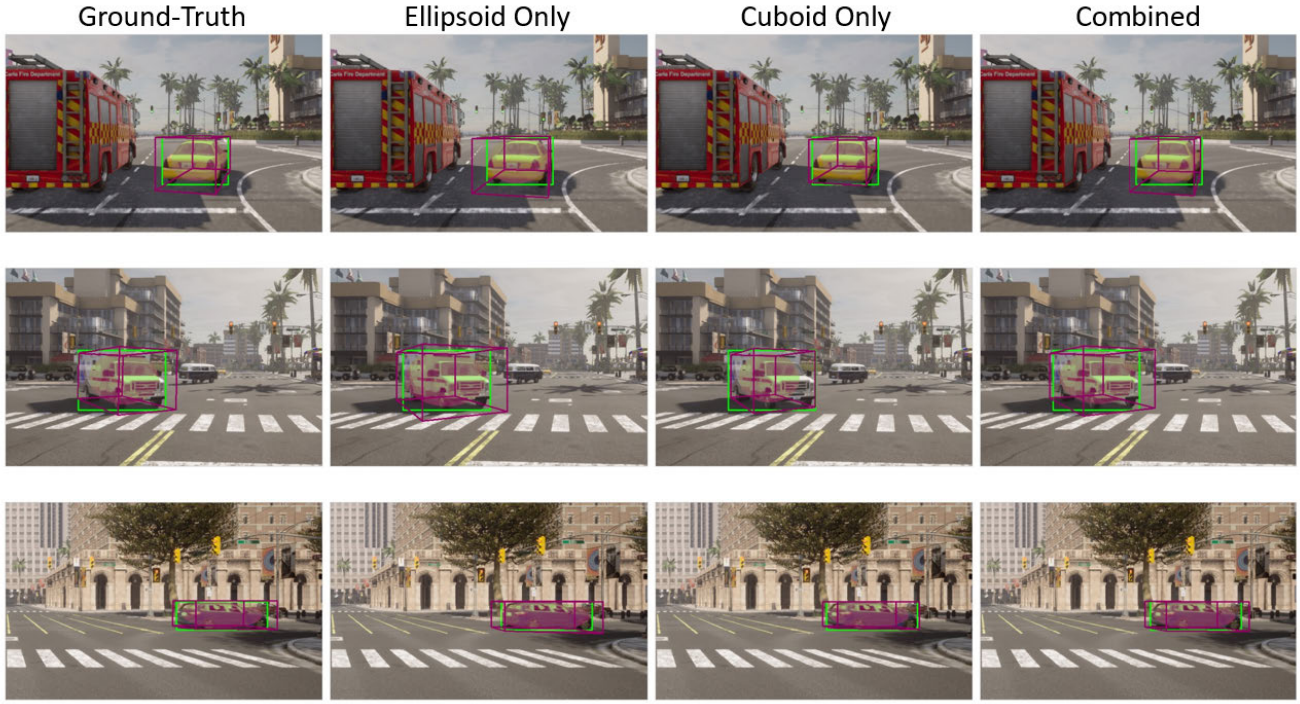


FIGURE 8. Snapshots of the ground-truth and estimated 3D bounding boxes (colored in magenta) for methods using each 3D representation. The reprojected bounding boxes of the combined representation most closely resemble the ground-truth, while ellipsoid-only and cuboid-only methods result in over/underestimated object size.

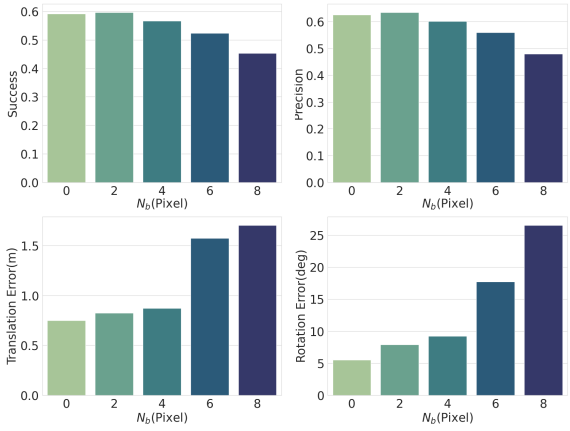


FIGURE 9. Performance evaluation against varying degree of noise level($N_b = 0, 2, 4, 6, 8$px)).

representation leads to the smallest average size estimation error. Since the size and pose of the objects are coupled by bounding box constraints in the proposed framework, size estimation error negatively affects the object trajectory estimation performance.

C. ROBUSTNESS TO 2D DETECTION ERROR

In this section, we test the robustness of the algorithm against varying degree of 2D detection noise. We assume that the class label is not misidentified, since state-of-the-art 2D object detection methods rarely misclassify detected object. Those rare incidents only happen if the appearance of the object is ambiguous or when the object is substantially occluded, which are not the cases we are interested in. Instead, we perturb the 2D bounding box by adding noise to 4 bounding box coordinates $(\beta_l, \beta_t, \beta_r, \beta_b)$. We sample noise from Gaussian distribution $n_{b,i} \sim \mathcal{N}(0, N_b), i = l, t, r, b$, and examine the method for different values of N_b (2, 4, 6, 8(px)).

For each 2D bounding box noise level, we run the simulation 5 times for every object observation sequence, and compute the average of each evaluation metric over the whole dataset. The graph shown in Figure 9 displays the result. The performance of the proposed method does not significantly degrade for $N_b = 0, 2, 4$px. As the noise grows larger, the performance is severely degraded. The reliance on 2D bounding box accuracy is a shortcoming which is commonly identified on two stage approaches for 3D object detection and tracking. However, we believe that modern 2D instance segmentation methods rarely exhibit such large error, and the result shows an acceptable level of robustness of the proposed method.

VI. CONCLUSION

In this paper, we propose an optimization-based 3D object tracking framework for monocular camera based on combining ellipsoid and cuboid object representation. The proposed method utilizes a 2D instance segmentation method to acquire

object region of interest from RGB image, and does not require additional neural network training.

To address scale ambiguity inherent in monocular vision, we limit the degree of freedom of the object pose assuming a known support plane for target objects. We additionally apply kinematic motion model to disambiguate the orientation of the object. The object size and trajectory are jointly optimized by minimizing the keypoint reprojection error, motion model error and the proposed bounding box error for the combined 3D representation.

One critical limitation of the proposed framework is that the method is not able to track object when the object is partially occluded, since the bounding box error formulation presumes that the object is fully observable except for inevitable self-occlusion. However, results show that when the object is fully visible, the proposed method is able to keep track of the object and accurately estimate its 3D trajectory. The simulation results reveal that the proposed 3D representation combining ellipsoid and cuboid leads to more accurate estimation of object size and trajectory compared to when using each representation separately. Also, the method is shown to be robust to moderate degree of error in 2D bounding box detection.

REFERENCES

[1] X. Weng, J. Wang, D. Held, and K. Kitani, "3D multi-object tracking: A baseline and new evaluation metrics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10359–10366.

[2] C. Zheng, X. Yan, H. Zhang, B. Wang, S. Cheng, S. Cui, and Z. Li, "Beyond 3D Siamese tracking: A motion-centric paradigm for 3D single object tracking in point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8101–8110.

[3] Z. Pang, Z. Li, and N. Wang, "SimpleTrack: Understanding and rethinking 3D multi-object tracking," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2022, pp. 680–696.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[5] Z. Li, Z. Chen, A. Li, L. Fang, Q. Jiang, X. Liu, and J. Jiang, "Unsupervised domain adaptation for monocular 3D object detection via self-training," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 245–262.

[6] Z. Li, Z. Chen, A. Li, L. Fang, Q. Jiang, X. Liu, and J. Jiang, "Towards model generalization for monocular 3D object detection," 2022, *arXiv:2205.11664*.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[8] P. Li, X. Chen, and S. Shen, "Stereo R-CNN based 3D object detection for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7636–7644.

[9] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecká, "3D bounding box estimation using deep learning and geometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5632–5640.

[10] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau, "Deep MANTA: A coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1827–1836.

[11] J. Shi, H. Yang, and L. Carlone, "Optimal and robust category-level perception: Object pose and shape estimation from 2-D and 3-D semantic keypoints," *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 4131–4151, Oct. 2023.

[12] J. K. Murthy, G. V. S. Krishna, F. Chhaya, and K. M. Krishna, "Reconstructing vehicles from a single image: Shape priors for road scene understanding," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 724–731.

[13] F. Engelmann, J. Stuckler, and B. Leibe, "SAMP: Shape and motion priors for 4D vehicle reconstruction," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 400–408.

[14] L. Nicholson, M. Milford, and N. Sunderhauf, "QuadricSLAM: Dual quadrics from object detections as landmarks in object-oriented SLAM," *IEEE Robot. Autom. Lett.*, vol. 4, no. 1, pp. 1–8, Jan. 2019.

[15] R. Tian, Y. Zhang, Y. Feng, L. Yang, Z. Cao, S. Coleman, and D. Kerr, "Accurate and robust object SLAM with 3D quadric landmark reconstruction in outdoors," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1534–1541, Apr. 2022.

[16] Z. Liao, Y. Hu, J. Zhang, X. Qi, X. Zhang, and W. Wang, "SO-SLAM: Semantic object SLAM with scale proportional and symmetrical texture constraints," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4008–4015, Apr. 2022.

[17] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D object SLAM," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 925–938, Aug. 2019.

[18] Y. Liu, Y. Yixuan, and M. Liu, "Ground-aware monocular 3D object detection for autonomous driving," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 919–926, Apr. 2021.

[19] S. Srivastava, F. Jurie, and G. Sharma, "Learning 2D to 3D lifting for object detection in 3D for autonomous vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4504–4511.

[20] A. Kundu, K. M. Krishna, and C. V. Jawahar, "Realtime multibody visual SLAM with a smoothly moving monocular camera," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2080–2087.

[21] Z. Cao, Y. Zhang, R. Tian, R. Ma, X. Hu, S. Coleman, and D. Kerr, "Object-aware SLAM based on efficient quadric initialization and joint data association," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9802–9809, Oct. 2022.

[22] Y. Wu, Y. Zhang, D. Zhu, Y. Feng, S. Coleman, and D. Kerr, "EAO-SLAM: Monocular semi-dense object SLAM based on ensemble data association," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 4966–4973.

[23] P. Li and T. Qin, "Stereo vision-based semantic 3D object and ego-motion tracking for autonomous driving," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 646–661.

[24] H.-N. Hu, Q.-Z. Cai, D. Wang, J. Lin, M. Sun, P. Kraehenbuehl, T. Darrell, and F. Yu, "Joint monocular 3D vehicle detection and tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5389–5398.

[25] H.-N. Hu, Y.-H. Yang, T. Fischer, T. Darrell, F. Yu, and M. Sun, "Monocular quasi-dense 3D object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 1992–2008, Feb. 2023.

[26] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "NuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11618–11628.

[27] D. Beker, H. Kato, M. A. Morariu, T. Ando, T. Matsuoka, W. Kehl, and A. Gaidon, "Monocular differentiable rendering for self-supervised 3D object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 514–529.

[28] I. Mouawad, N. Brasch, F. Manhardt, F. Tombari, and F. Odone, "Time-to-label: Temporal consistency for self-supervised monocular 3D object detection," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 8988–8995, Oct. 2022.

[29] L. Chen, J. Sun, Y. Xie, S. Zhang, Q. Shuai, Q. Jiang, G. Zhang, H. Bao, and X. Zhou, "Shape prior guided instance disparity estimation for 3D object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5529–5540, Sep. 2022.

[30] R. Wang, N. Yang, J. Stuckler, and D. Cremers, "DirectShape: Direct photometric alignment of shape priors for visual vehicle pose and shape estimation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 11067–11073.

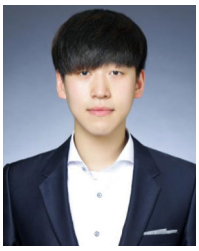
[31] N. Sunderhauf and M. Milford, "Dual quadrics from object detection bounding boxes as landmark representations in SLAM," 2017, *arXiv:1708.00965*.

[32] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granström, "Mono-camera 3D multi-object tracking using deep learning detections and PMBM filtering," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 433–440.

[33] G. Guo and S. Zhao, "3D multi-object tracking with adaptive cubature Kalman filter for autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 1, pp. 512–519, Jan. 2023.

[34] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[35] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Visual SLAM: Why filter?" *Image Vis. Comput.*, vol. 30, no. 2, pp. 65–77, Feb. 2012.
[36] P. Li, J. Shi, and S. Shen, "Joint spatial–temporal optimization for stereo 3D object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6876–6885.
[37] J. Huang, S. Yang, T.-J. Mu, and S.-M. Hu, "ClusterVO: Clustering moving instances and estimating visual odometry for self and surroundings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2165–2174.
[38] J. Shi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 1994, pp. 593–600.
[39] H. Karunasekera, H. Wang, and H. Zhang, "Multiple object tracking with attention to appearance, structure, motion and size," *IEEE Access*, vol. 7, pp. 104423–104434, 2019.
[40] C. Kim, Y. Jang, J. Kim, P. Kim, and H. Jin Kim, "Scale-aware monocular visual odometry and extrinsic calibration using vehicle kinematics," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 14757–14771, Dec. 2023.
[41] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," *Georgia Inst. Technol.*, vol. 2, p. 4, Jul. 2012.
[42] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn.*, Nov. 2017, pp. 1–16.
[43] M. Kristan, J. Matas, A. Leonardis, T. Vojír, R. Pflugfelder, G. Fernández, G. Nebehay, F. Porikli, and L. Cehovin, "A novel performance evaluation methodology for single-target trackers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2137–2155, Nov. 2016.

GYEONG CHAN KIM (Graduate Student Member, IEEE) received the B.S. degree in electrical and computer engineering from Seoul National University, South Korea, in 2019, where he is currently pursuing the integrated M.S. and Ph.D. degrees with the Department of Aerospace Engineering. His research interests include 3D reconstruction and vision-based navigation for robotics systems.



YOUNGSEOK JANG (Graduate Student Member, IEEE) received the B.S. degree in mechanical engineering from Sungkyunkwan University, Suwon, South Korea, in 2017. He is currently pursuing the integrated M.S. and Ph.D. degrees with the Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, South Korea. His research interests include visual navigation for multirobot systems, sensor fusion for navigation, and perception-aware path planning.

H. JIN KIM (Member, IEEE) received the B.S. degree from Korea Advanced Institute of Technology (KAIST), in 1995, and the M.S. and Ph.D. degrees in mechanical engineering from the University of California at Berkeley (UC Berkeley), in 1999 and 2001, respectively. From 2002 to 2004, she was a Postdoctoral Researcher in electrical engineering and in computer science with UC Berkeley. In 2004, she joined the Department of Mechanical and Aerospace Engineering, Seoul National University, as an Assistant Professor, where she is currently a Professor. Her research interests include intelligent control of robotic systems and motion planning.