Navigation of a Self-Driving Vehicle Using One Fiducial Marker

Yibo Liu, Hunter Schofield, Jinjun Shan

Abstract— Navigation using only one marker, which contains four artificial features, is a challenging task since camera pose estimation using only four coplanar points suffers from the rotational ambiguity problem in a real-world application. This paper presents a framework of vision-based navigation for a self-driving vehicle equipped with multiple cameras and a wheel odometer. A multiple camera setup is presented for the camera cluster which has $360°$ vision such that our framework solely requires one planar marker. A Kalman-Filter-based fusion method is introduced for the multiple-camera and wheel odometry. Furthermore, an algorithm is proposed to resolve the rotational ambiguity problem using the prediction of the Kalman Filter as additional information. Finally, the lateral and longitudinal controllers are provided. Experiments are conducted to illustrate the effectiveness of the theory.

I. INTRODUCTION

Fiducial marker systems, such as AprilTag [1][2], are designed to enhance augmented reality, evaluate performance of robot systems, and improve human-robot interactions, etc. Markers provide the environment with controllable and stable artificial features. This simplifies feature extraction [3], which is why markers are widely used for tasks that require high detection speed, such as visual servoing [4], [5]. As markers are being designed to be easily detected from a wider range of locations, planar marker-based SLAM [6], [7] is becoming a hot trend. It is fundamental to estimate the 6-DOF pose of the camera with respect to the world or marker coordinate system in research that employs marker-based localization. In practice, the pose is computed using the four corners of the marker. Since the corners are co-planar, this is known as a specific case of the Perspective-N-Point problem called planar pose estimation [3].

Theoretically, there exists a unique estimation of the 6-DOF pose given four co-planar but non-collinear points like the corners of a marker [8]. Yet in real-world, when the projected marker on the image plane is small and the relative motion between the camera and the marker is fast, the perspective effects become weak due to sensory noise. Therefore, in weak-perspective conditions, the problem of two-fold rotation ambiguity that corresponds to an unknown reflection of the plane about the camera's z-axis arises [9]. Fig. 1 shows a sketch of the rotational ambiguity problem which occurs very often in real applications [6].

Solving the rotational ambiguity is crucial for visual servoing works where the control commands are generated based on the camera's measurements. Take our proposed framework as the example, the steering angle command

The authors are with Department of Earth and Space Science and Engineering, York University, Toronto, Ontario M3J1P3, Canada {yorklyb,hunterls,jjshan}@yorku.ca
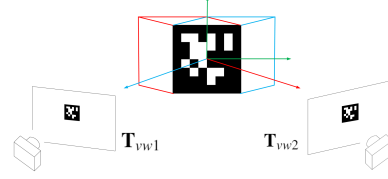

Fig. 1. An illustration of the rotational ambiguity problem. When the camera is moving with the robot, the image noise shall weaken the perspective such that a marker could project at the same pixels from two different poses, $\mathbf{T}_{vw1}$ and $\mathbf{T}_{vw2}$.
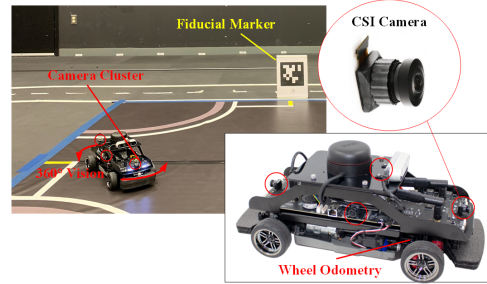

Fig. 2. A diagram of our self-driving vehicle. There are four 2D CSI cameras placed on the vehicle on the front, rear, and both sides. The $360°$ view of the camera cluster guarantees that the marker can always be observed. The on-board wheel odometer is also employed in this work.

mainly relies on the relative orientation between the desired waypoint and the current location of the vehicle (Fig. 2). Thus, the rotational ambiguity can affect the decision of whether to turn right or left. Once the vehicle executes the wrong turning decision; the navigation task then fails. Even worse, the ambiguity of rotation leads to the flipping of translation, which ruins the positioning directly. Note that such a problem cannot be solved by methods like applying a Kalman Filter (KF) since the error is not caused by noise but by possible dual solutions.

In this work, we employ the AprilTag system since it has better performance over other systems [2]. Even though the rotational ambiguity is not considered in the implementation of the AprilTag system [1], [2], the state-of-art planar pose estimation method, Infinitesimal Plane-Based Pose Estimation (IPPE) [10], returns the two possible solutions. A straightforward method to disambiguate the two poses is comparing the reprojection errors. The pose with a lower reprojection error is then selected as the true pose. However, as shown in [7] and [3], the reprojection errors of the two ambiguous poses can be extremely close to each other and the one closer to the ground truth could have a higher reprojection error. Although the rotational ambiguity

problem has been clarified in the novel theory [10] for marker pose estimation, it is still challenging to find the true pose from the two returned physically possible solutions. In the next section, a further survey is presented for the works that have been done for resolving rotational ambiguity in marker-based pose estimation.

A. Related Work

To the best of our knowledge, the methods for resolving the rotation ambiguity can be categorized as ambiguity detection [6], [7], marker improvement [11], [12], and additional information assistance [13], [3], [14], [15].

Munoz et al. proposed an ambiguity detection algorithm in [6]. If the difference of the reprojection errors of the two possible poses is greater than a predetermined threshold, the pose with lower reprojection error is selected as the true pose. Otherwise, that marker detection is discarded. This algorithm is proved to be effective for marker-based SLAM [6], [7]. However, for difficult environments where only one marker is available, and also for the applications where the robot needs to respond to the camera's measurements in real-time, simply detecting the ambiguity poses and discarding the ambiguity detections will lead to loss of localization and unstable performance.

Tanaka et al. [11], [12] presented a new type of marker called LentiMark by adding two new Moiré patterns to the conventional marker design. Unique angle information can be obtained from LentiMark, which eliminates the pose ambiguity. The main limitation of LentiMark is the complicated fabrication compared to that of the low-cost conventional marker.

Besides the methods mentioned above, additional information assistance is also very effective for mitigating the rotational ambiguity. The source of the alleged additional information can from another sensor, predefined motion model, etc. Jin et al. [13] introduced an algorithm based on the measurements from an RGBD camera. The depth pose estimation provides optimization constraints such that the optimal pose is the one aligned with the plane in the depth space. While this algorithm requires an RGBD camera. Ch'ng et al. [3] developed a method that solves ambiguity by examining the consistencies of a set of markers across multiple views. This method shows good potential in marker-based SLAM, but it is not applicable for cases where only one marker is available and the robot needs to respond based on a single view. Wu et al. [14] showed a filtering based approach to disambiguate the poses. However, their approach only considers the 3D object-space error. Visual-IMU-wheel odometry [16] methods have the capability to deal with ambiguity poses using factor-graph based optimization. A factor-graph based approach using IMU preintegration is presented in [15] to solve the rotational ambiguity. Although factor-graph is very effective as it uses the measurements from inertia sensors to construct constraint edges, the graph optimization is usually a back-end process that cannot provide the optimized results for the current frame in real-time.

B. Contributions

We first introduce a multiple-camera model for fusing the features detected on different image planes and a KF to fuse the sensor data from the camera cluster and the wheel odometry. Then, we propose a novel algorithm to solve the rotational ambiguity by constructing a cost function, e, at the feature level. In particular, the cost function considers both the 2D reprojection error (image-space error), and the 3D object-space error. This algorithm is capable of providing robust pose estimation under weak-perspective conditions caused by robot motion and limited marker size. Moreover, our algorithm only requires one marker and the current frame to work.

II. PROPOSED FRAMEWORK

A. Multiple-Camera Model

Before introducing the coupled multiple-camera model, we first present the pinhole camera model (see Fig. 3) for a single camera:

$$s_i \begin{bmatrix} \mathbf{u}_i \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{T} \begin{bmatrix} \mathbf{p}_i^w \\ 1 \end{bmatrix} \tag{1}$$

where s_i are scalar projective parameters for $i = \{1, 2, 3, 4\}$ in this work. $\mathbf{K} \in \mathbb{R}^3$ is the camera intrinsic matrix determined by calibration [17]. $\mathbf{T} \in \mathrm{SE}(3)$ is defined as the camera extrinsic matrix that represents the transmission from the world coordinate system $\{W\}$ to the camera coordinate system $\{C\}$. Detailed math preliminaries of $\mathrm{SE}(3)$ are introduced in [18].
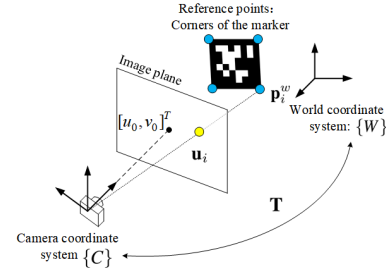

Fig. 3. The pinhole camera model. $\mathbf{p}_i^w = [x_i^w, y_i^w, z_i^w]^T$ are the 3D world coordinates of the 4 reference points. $\mathbf{u}_i = [u_i, v_i]^T$ are the 2D image coordinates of the projection in the image plane. $[u_0, v_0]^T$ is the optic center of the image plane.
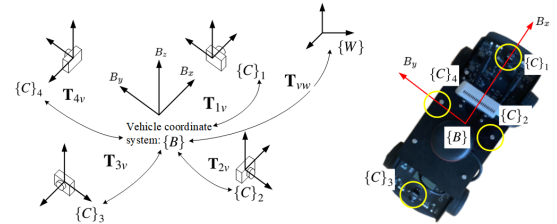

Fig. 4. The multiple-camera model. $\mathbf{T}_{1v}$, $\mathbf{T}_{2v}$, $\mathbf{T}_{3v}$, and $\mathbf{T}_{4v}$ represent the extrinsic matrices that describe the transmission from the camera coordinate systems $\{C\}_1$, $\{C\}_2$, $\{C\}_3$, and $\{C\}_4$ to the vehicle coordinate system $\{B\}$, respectively. These extrinsic matrices are determined by calibration [17].

Due to the distribution of the cameras (see Fig. 4), only the adjacent cameras have overlapping views. Hence, no more than two cameras can observe the marker at a moment. Looking down from above, the cameras are indexed with the first at the front of the vehicle, incrementing clockwise around the vehicle. Then, we denote the index number(s) of the camera(s) that can observe the marker as $j \subseteq \{\{1\}, \{2\}, \{3\}, \{4\}, \{1,2\}, \{2,3\}, \{3,4\}, \{4,1\}\}$.

Inspired by the camera model presented in [19], the proposed multiple-camera model is:

$$s_{ij} \begin{bmatrix} \mathbf{u}_{ij} \\ 1 \end{bmatrix} = \mathbf{K}_j \mathbf{T}_{jv} \mathbf{T}_{vw} \begin{bmatrix} \mathbf{p}_i^w \\ 1 \end{bmatrix} \tag{2}$$

where s_{ij} are scalar projective parameters and $\mathbf{u}_{ij}$ is the 2D projection of the ith reference point on the image plane of the jth camera. $\mathbf{K}_j$ denotes the camera intrinsic matrix of the jth camera. $\mathbf{T}_{jv}$ is introduced in the caption of Fig. 4. $\mathbf{T}_{vw}$ is the extrinsic matrix that describes the transmission from the vehicle coordinate system, $\{B\}$, to the world coordinate system, $\{W\}$. Note that no central camera is defined here and we use the pose of the vehicle to represent the pose of the camera cluster. In the cases where two cameras see the marker at the same time, we define the camera index numbers as $j = \{j_1, j_2\}$. The reprojection error of a given pose $\mathbf{T}_{vw}$ becomes:

$e_1 : \mathrm{SE}(3) \rightarrow \mathbb{R}^+$

$$e_1(\mathbf{T}_{vw}) = \sum_{j=j_1}^{j_2} \sum_{i=1}^{4} \| \frac{1}{s_{ij}} \mathbf{K}_j \mathbf{T}_{jv} \mathbf{T}_{vw} \begin{bmatrix} \mathbf{p}_i^w \\ 1 \end{bmatrix} - \begin{bmatrix} \mathbf{u}_{mij} \\ 1 \end{bmatrix} \|^2 \tag{3}$$

where $\mathbf{u}_{mij} = [u_{mij}, v_{mij}]^T$ is the observed 2D projection of the ith reference point on the image plane of the jth camera. As for the scenes with only one camera observing the marker, the summation $\sum_{j=j_1}^{j_2}$ in Eq. (3) shall be removed.

The vehicle pose $\mathbf{T}_{vw}$ is determined through minimizing the reprojection error shown in Eq. (3), which is inherently a least square problem that can be solved using the Levenberg Marquardt (LM) algorithm [20]. The initial values for LM optimization are obtained from solving IPPE [10].

B. Kinematics of the Vehicle

The wheel odometer provides the measurement of linear velocity, $\mathbf{V}_w$, by transforming the encoder counts. Moreover, the steering angle δ_r can be measured by a angle recorder in this work. To propagate the pose of the vehicle, a 3-DOF bicycle model (see Fig. 5) is employed.

From the geometric relation behind Fig. 5, the sideslip angle β is obtained by:

$$\beta = \tan^{-1}(\frac{1}{2}\tan\delta_r) \tag{4}$$

The turning radius R_r becomes:

$$R_r = \frac{l}{\cos\beta \ \tan\delta_r} \tag{5}$$

where l is wheelbase of the vehicle. Given the linear velocity, v_w, and turning radius, R_r, the vehicle's angular rate is as follows:
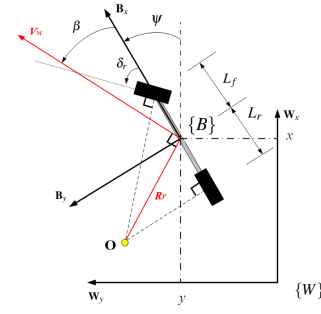
$$\dot{\psi} = \frac{v_w}{R_r} \tag{6}$$

Fig. 5. The 3-DOF bicycle model. $\{B\}$ and $\{W\}$ denote the vehicle and world coordinate systems, respectively. ψ is the yaw orientation of the vehicle w.r.t. $\{W\}$. δ_r is the steering angle of the front wheel. Depending on δ_r, the vehicle turns about the turning point, $\mathbf{O}$, with a turning radius, R_r. β is the sideslip angle that describes the orientation of the velocity vector, v_w, w.r.t. the vehicle coordinate system. $[x,y]^T$ is the 2D position of the vehicle w.r.t $\{W\}$. The centre of gravity of the vehicle is close to the centre of the vehicle chassis so that we assume that they are the same, i.e. $L_f = L_r$.

The velocity of the vehicle w.r.t. $\{W\}$ is then:

$$\begin{bmatrix} v_x \\ v_y \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} v_w\cos(\beta + \psi) \\ v_w\sin(\beta + \psi) \\ \dot{\psi} \end{bmatrix} \tag{7}$$

However, integrating the states shown in Eq. (7) will lead to errors due to integration drift [21]. Hence, we introduce a KF to correct the prediction of the kinematics using the measurements from the camera cluster in the next section.

C. Kalman Filter Based Sensor Fusion

A KF is employed to fuse the measurements from the camera cluster and the wheel odometer. The filter state vector is defined as:

$$\mathbf{x} = [x, y, \psi, v_x, v_y, \dot{\psi}]^T \tag{8}$$

The prediction is given by:

$$\begin{aligned} \hat{\mathbf{x}}_{k,k-1} &= \mathbf{A}\hat{\mathbf{x}}_{k-1,k-1} \\ \mathbf{P}_{k,k-1} &= \mathbf{A}\mathbf{P}_{k-1,k-1}\mathbf{A}^T + \mathbf{Q}_{k-1} \end{aligned} \tag{9}$$

where $\mathbf{A} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{dt} \cdot \mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix}$ is a 6×6 matrix. Here, $\mathbf{dt}$ is the sample period, and k is the sample step. $\mathbf{P}_{k-1,k-1}$ is a posterior covariance of the estimation error for sample step $k-1$. $\mathbf{P}_{k,k-1}$ is the *a priori* covariance of the estimation error for sample step k. $\mathbf{Q}_{k-1}$ is the covariance of the systematic noise. The observation equation is as follows:

$$z_k = \mathbf{H}\mathbf{x} \tag{10}$$

where $\mathbf{H} = \mathbf{I}_{6\times6}$ is an identity matrix and $z_k = [z_x, z_y, z_\psi, z_{vx}, z_{vy}, z_{\dot{\psi}}]^T$. Note that $[z_x, z_y, z_\psi]^T$ is measured by the camera cluster and $[z_{vx}, z_{vy}, z_{\dot{\psi}}]^T$ is obtained from the wheel odometer and steering angle recorder. The Kalman gain is obtained by:

$$\mathbf{K}_k = \mathbf{P}_{k,k-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{k,k-1}\mathbf{H}^T + \mathbf{R}_k)^{-1} \tag{11}$$

where $\mathbf{R}_k$ is the covariance of the observation noise. The update process will be then:

$$\begin{aligned} \hat{\mathbf{x}}_{k,k} &= \hat{\mathbf{x}}_{k,k-1} + \mathbf{K}_k(z_k - \mathbf{H}\hat{\mathbf{x}}_{k,k-1}) \\ \mathbf{P}_{k,k} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_{k,k-1} \end{aligned} \tag{12}$$

where $\boldsymbol{P}_{k,k}$ is the *a posterior* covariance of the estimation error of sample step k.

D. Resolving Rotation Ambiguity

Suppose $\hat{\boldsymbol{x}}_{k,k-1}$ is the *a priori* of sample step k (see Eq. (8) and Eq. (9)), we defined the pose associated with $\hat{\boldsymbol{x}}_{k,k-1}$ as $\mathbf{T}_w \in \mathrm{SE}(3)$. The formula that transforms the Euler angles and translation vector into $\mathrm{SE}(3)$ is shown in [18]. We define the function of this transmission as:

$$f : \mathbb{R}^6 \to \mathrm{SE}(3)$$
$$\mathbf{T}_w = f([\hat{\boldsymbol{x}}_{k,k-1}^T, \mathbf{0}_{1\times 3}]^T) \tag{13}$$

where $[\hat{\boldsymbol{x}}_{k,k-1}^T, \mathbf{0}_{1\times 3}] \in \mathbb{R}^6$ represents $\hat{\boldsymbol{x}}_{k,k-1}^T$ augmented by $\mathbf{0}_{1\times 3}$. That is, we set the pitch, roll angles and Z position as zero. The core of the proposed algorithm for resolving the rotational ambiguity is to minimize the value of a cost function at feature-level:

$$e : \mathrm{SE}(3) \to \mathbb{R}^+$$
$$e(\mathbf{T}_{vw}) = e_1(\mathbf{T}_{vw}) + e_2(\mathbf{T}_{vw}) \tag{14}$$

where $\mathbf{T}_{vw}$ is the to be determined pose of the vehicle and $e_1(\mathbf{T}_{vw})$ is the reprojection error shown in Eq. (3). The definition of $e_2(\mathbf{T}_{vw})$ is given by:

$$e_2 : \mathrm{SE}(3) \to \mathbb{R}^+$$
$$e_2(\mathbf{T}_{vw}) = \sum_{i=1}^{4} || \mathbf{T}_{vw} \begin{bmatrix} \mathbf{p}_i^w \\ 1 \end{bmatrix} - \mathbf{T}_w \begin{bmatrix} \mathbf{p}_i^w \\ 1 \end{bmatrix} ||^2 \tag{15}$$

where $\mathbf{T}_w$ is the pose associated with the *a priori* from the prediction of the KF (see Eq. (9)). $e_2(\mathbf{T}_{vw})$ describes the error between the 3D coordinates of two points, $\mathbf{T}_{vw} \begin{bmatrix} \mathbf{p}_i^w \\ 1 \end{bmatrix}$ and $\mathbf{T}_w \begin{bmatrix} \mathbf{p}_i^w \\ 1 \end{bmatrix}$, in $\{B\}$. There is no subtraction defined on $\mathrm{SE}(3)$ [18], but we can evaluate the difference between $\mathbf{T}_w$ and $\mathbf{T}_{vw}$ through Eq. (15). Hence, the cost e is actually a sum of the 2D image-space-error $e_1(\mathbf{T}_{vw})$ and 3D object-space error $e_2(\mathbf{T}_{vw})$.

Suppose the two poses given by the IPPE method [10] are $\mathbf{T}_{vw1}$ and $\mathbf{T}_{vw2}$. The unambiguous pose $\mathbf{T}_{vw}^*$ is given by:

$$\mathbf{T}_{vw}^* \in \{\mathbf{T}_{vw1}, \mathbf{T}_{vw2}\} \text{ s.t.}$$
$$e(\mathbf{T}_{vw}^*) = \min\{e(\mathbf{T}_{vw1}), e(\mathbf{T}_{vw2})\} \tag{16}$$

That is, the pose which has a smaller cost value is selected to be $\mathbf{T}_{vw}^*$.

There are two issues we want to clarify for this algorithm. First, usually the direct measurements are used to construct the residual errors, such as the preintegration-based methods proposed in [16]. In this work, we use kinematics and integration to obtain $\mathbf{T}_w$, then we construct $e_2(\mathbf{T}_{vw})$. The direct measurements are the encoder counts and steering angle. Thus, errors due to kinematics and integration are brought into $e_2(\mathbf{T}_{vw})$. Nevertheless, as illustrated in [3], [14], the true pose is one of the two possible poses returned by the IPPE method [10]. Moreover, the true pose can be found using additional information from other sensors [13]. Hence, when solely concerning the rotation ambiguity problem, Eq. (14) is still an effective criteria. Secondly, we

augmented $\hat{\boldsymbol{x}}_{k,k-1} \in \mathbb{R}^3$ into $\mathbb{R}^6$ to compute $\mathbf{T}_w$, which seems superfluous for a ground vehicle. However, using $\mathrm{SE}(3)$ indicates that Eq. (14) can also be used for aerial robots, while the kinematics and state vector of the KF will be different.

E. Lateral and Longitudinal Control

The lateral controller is based on the classic Ackermann steering geometry. As illustrated in Fig. 5, the vehicle turns about the turning point $\mathbf{O}$, with a turning radius $\boldsymbol{R}_r$. Thus, the trajectory of the centre of the vehicle chassis will be a circular arc as shown in Fig. 6.
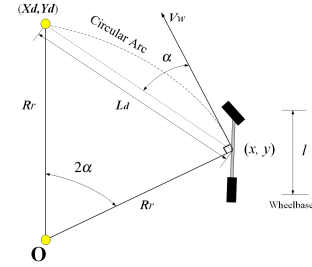

Fig. 6. The Ackermann steering geometry. (X_d, Y_d) is the desired waypoint. α is the angle between the desired waypoint and the vehicle velocity. $\boldsymbol{L}_d = \sqrt{(Y_d - y)^2 + (X_d - x)^2}$ is the distance between the desired waypoint and the centre of the vehicle chassis.

At discrete control iteration k, from the geometry shown in Fig. 6, we have:

$$\boldsymbol{L}_{d,k}\cos\alpha_k = \boldsymbol{R}_{r,k}\sin(2\alpha_k) \tag{17}$$

where α_k is shown as follows:

$$\alpha_k = \arctan(\frac{y_k - \boldsymbol{Y}_{d,k}}{x_k - \boldsymbol{x}_{d,k}}) - \psi_k - \beta_k \tag{18}$$

Thus, given a desired waypoint $(\boldsymbol{X}_{d,k}, \boldsymbol{Y}_{d,k})$, the desired steering angle $\delta_{rd,k}$ is generated by substituting Eq. (5) into Eq. (17):

$$\delta_{rd,k} = \arctan(\frac{2l\sin\alpha_k}{\boldsymbol{L}_{d,k}\cos\beta_k}) \tag{19}$$

We employ a PI controller for the longitudinal control. The throttle input u_k of the vehicle is given by:

$$u_k = p_1\boldsymbol{L}_{d,k} + p_2\sum_{n=1}^{k}\boldsymbol{L}_{d,n}, \ n = \{1, \cdots, k\} \tag{20}$$

where p_1 and p_2 are the proportional and integral gains, respectively. We can summarize our proposed framework in the following algorithm.

Remark 1: The one marker case is not an artificially difficult scenario. Even if multiple markers are set in the environment, there might be a moment where only one marker is observable. Since our framework is developed for the one-marker case, it can be easily extended to the multi-marker scenario.

Algorithm 1: Proposed Framework

Input: a desired waypoint coordinate, (X_d, Y_d)
Output: steering angle, δ_r, and throttle input, $\mathbf{u}_k$.

Define a waypoint radius, r.
while $\sqrt{(X_d - x)^2 + (Y_d - y)^2} > r$ **do**

    Get images from camera.
    Get AprilTag features.[2]
    Get both possible poses T_{vw1}, and T_{vw2} using the IPPE method [10] and (3)
    Get the *a priori* from the Kalman Prediction (9)
    Compute T_w using (13).
    Compute $e(T_{vw1})$ and $e(T_{vw2})$ using (14).
    Compute T^*_{vw} using (16).
    Get the *a posteriori* from the Kalman update (12).
    Compute β, R_r, L_d, and α using (4), (5),(17), and (18).
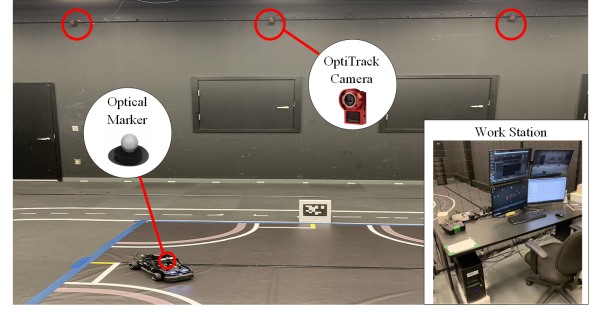   Solve for δ_r and u_k using (19) and (20).

Get next waypoint.

Fig. 7. The OptiTrack Motion Capture System, composed of 16 OptiTrack cameras and one work station. It can track the 6-DOF pose of predefined rigid bodies via optical markers at 100 Hz.

III. EXPERIMENTAL RESULTS

The proposed framework is validated using the Quanser's Qcar (see Fig. 2) which is a sensor-rich autonomous vehicle for testing novel self-driving algorithms. The QCar is equipped with an NVIDIA Jetson TX2 for a computer. Our algorithm is developed using Python 3, and is implemented on an Ubuntu 18.04 OS. Quanser provides the Python libraries required to drive all the sensors and actuators. The camera cluster's image timestamps are aligned, and the cluster can capture images while detecting the AprilTag at approximately 15 Hz. We sample from the encoder when new images are captured, and only generate the control commands after updating the KF. Overall, the proposed framework can run at approximately 11 Hz on the QCar platform. Additionally, the wheelbase of the vehicle is 25.6 cm and the steering servo can rotate from -0.5 to 0.5 rad. Our marker setup is a low-cost implementation. The marker, with a side length of 17.2 cm, is printed on A4 paper. To further verify the proposed framework, we employ the OptiTrack Motion Capture System (MoCap) at the Spacecraft Dynamics Control and Navigation Laboratory at York University to obtain the ground truth. Fig. 7 shows the MoCap system.

In the experiment, we compare the localization output of our algorithm (denoted **Our Method**), the MoCap, and the method (denoted **Method A**) which chooses the pose with lower reprojection error as the correct one. The original implementation of AprilTag [2] does not employ the IPPE method [10], so it's pose output is the one with the lower reprojection error. Thus, **Method A** is the method used in the AprilTag system, which is the reason for the large measurement errors found in [13]. The vehicle is placed approximately at (200, -100) cm in $\{W\}$, and we do not know the accurate initial poses in advance. From our experiments, we found that for this initial location, the true pose has a lower reprojection error while the vehicle is static, and the ambiguous pose has a lower reprojection error while the vehicle is moving. The objective is to navigate the vehicle through waypoints at (139, 0) cm and (50, 65) cm by turning. It should be noted that the navigation of the vehicle is based only on the output of **Our Method**, while **Method A** runs simultaneously with **Our Method**. Moreover, we applied the same KF (Section. II-C) for **Method A**. The experiment video can be found at the following URL: http://3vf8.2.vu/1.

As shown in Fig. 8 and Fig. 9, the outputs of **Method A + KF** and **Our Method** are almost the same at the beginning. This is because the perspective-effect is relatively strong when the vehicle is static and the true pose has the lower reprojection error. However, when the vehicle starts to move, the perspective-effect is weakened and **Method A + KF** starts to select the ambiguous pose while **Our Method** remains stable. Therefore, without using **Our Method** to resolve ambiguity, the vehicle can execute a wrong steering command, leading to a crash. As the vehicle approaches the marker, both **Method A+KF** and **Our Method** converge to the measurement from MoCap. This is because the size of the marker in the image plane increases as the vehicle gets closer the marker thus perspective-effect is enhanced and **Method A + KF** can also work. However, an initial error exists between the output of **Our Method** and the MoCap. If the perspective-effect is insufficient due to a limited marker size, and long distance between the camera and marker, then **Our Method** cannot handle the initial error since the cameras are the only source for the initial localization of $\{W\}$.

Using the proposed framework, the vehicle successfully passes through the waypoints. Fig. 8 and Fig. 9 show that even if the KF is applied, such a navigation task shall fail using **Method A**. This is the default logic used in the AprilTag system for choosing the true pose from possible poses. As for the unsolved initial error problem, a deep learning-based image deblurring approach, such as DeblurGAN [22] is a promising solution for cases where the perspective-effect is not ideal.

IV. CONCLUSIONS

In this work, a framework navigation of a self-driving vehicle using only one fiducial marker is proposed. The rotational ambiguity problem is solved through the computation
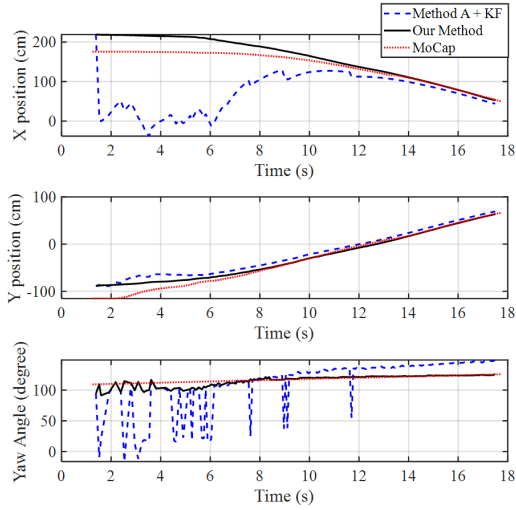
Fig. 8. Comparison of the trajectories in X position, Y position, and Yaw angle along time.
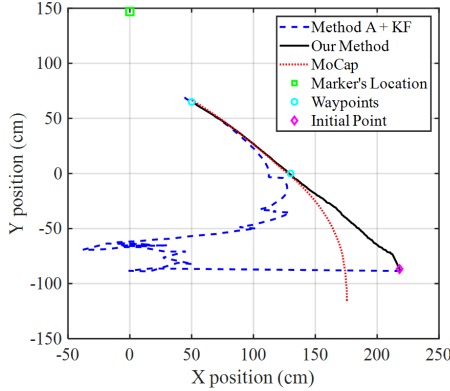

Fig. 9. Comparison of trajectories in the X-Y plane of {W}. The four corners of the marker are set at (-8.6, 147, 31.2) cm, (8.6, 147, 31.2) cm, (8.6, 147, 14) cm, and (-8.6, 147, 14) cm in {W}, respectively.

of a novel feature level cost. The cost is computed for each pose returned by the IPPE method, using the *a priori* given by the Kalman Filter, which fuses the measurements from the cameras with the wheel odometer. The pose with lower cost is selected for the Kalman Filter update and control command generation. The framework is validated experimentally and proven to be effective. Future work includes using Deblur-GAN to mitigate motion blur and enhance the perspective-effect, which has the potential for further improving the pose estimation accuracy.

ACKNOWLEDGMENT

REFERENCES

[1] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3400–3407.

[2] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4193–4198.

[3] S.-F. Ch'ng, N. Sogi, P. Purkait, T.-J. Chin, and K. Fukui, "Resolving marker pose ambiguity by robust rotation averaging with clique constraints," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9680–9686.

[4] A. Borowczyk, D.-T. Nguyen, A. P.-V. Nguyen, D. Q. Nguyen, D. Saussié, and J. Le Ny, "Autonomous landing of a quadcopter on a high-speed ground vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 9, pp. 2378–2385, 2017.

[5] S. M. Chaves, R. W. Wolcott, and R. M. Eustice, "Neec research: Toward gps-denied landing of unmanned aerial vehicles on ships at sea," *Naval Engineers Journal*, vol. 127, no. 1, pp. 23–35, 2015.

[6] R. Muñoz-Salinas, M. J. Marín-Jimenez, E. Yeguas-Bolivar, and R. Medina-Carnicer, "Mapping and localization from planar markers," *Pattern Recognition*, vol. 73, pp. 158–171, 2018.

[7] R. Munoz-Salinas, M. J. Marin-Jimenez, and R. Medina-Carnicer, "Spm-slam: Simultaneous localization and mapping with squared planar markers," *Pattern Recognition*, vol. 86, pp. 156–171, 2019.

[8] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle, "An analytic solution for the perspective 4-point problem," *Computer Vision, Graphics, and Image Processing*, vol. 47, no. 1, pp. 33–44, 1989.

[9] D. Oberkampf, D. F. DeMenthon, and L. S. Davis, "Iterative pose estimation using coplanar feature points," *Computer Vision and Image Understanding*, vol. 63, no. 3, pp. 495–511, 1996.

[10] T. Collins and A. Bartoli, "Infinitesimal plane-based pose estimation," *International journal of computer vision*, vol. 109, no. 3, pp. 252–286, 2014.

[11] H. Tanaka, Y. Sumi, and Y. Matsumoto, "A solution to pose ambiguity of visual markers using moire patterns," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3129–3134.

[12] H. Tanaka, K. Ogata, and Y. Matsumoto, "Solving pose ambiguity of planar visual marker by wavelike two-tone patterns," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 568–573.

[13] P. Jin, P. Matikainen, and S. S. Srinivasa, "Sensor fusion for fiducial tags: Highly robust pose estimation from single frame rgbd," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5770–5776.

[14] P.-C. Wu, J.-H. Lai, J.-L. Wu, and S.-Y. Chien, "Stable pose estimation with a motion model in real-time application," in *2012 IEEE International Conference on Multimedia and Expo*. IEEE, 2012, pp. 314–319.

[15] M. Fourmy, D. Atchuthan, N. Mansard, J. Sola, and T. Flayols, "Absolute humanoid localization and mapping based on imu lie group and fiducial markers," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2019, pp. 237–243.

[16] W. Lee, K. Eckenhoff, Y. Yang, P. Geneva, and G. Huang, "Visual-inertial-wheel odometry with online calibration," in *Proceedings of the 2020 International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA*, 2020, pp. 24–30.

[17] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1280–1286.

[18] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2017.

[19] M. J. Tribou, A. Harmat, D. W. Wang, I. Sharf, and S. L. Waslander, "Multi-camera parallel tracking and mapping with non-overlapping fields of view," *The International Journal of Robotics Research*, vol. 34, no. 12, pp. 1480–1500, 2015.

[20] S. Roweis, "Levenberg-marquardt optimization," *Notes, University Of Toronto*, 1996.

[21] S. A. Mohamed, M.-H. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen, and J. Plosila, "A survey on odometry for autonomous navigation systems," *IEEE Access*, vol. 7, pp. 97 466–97 486, 2019.

[22] N. Q. Truong, Y. W. Lee, M. Owais, D. T. Nguyen, G. Batchuluun, T. D. Pham, and K. R. Park, "Slimdeblurgan-based motion deblurring and marker detection for autonomous drone landing," *Sensors*, vol. 20, no. 14, p. 3918, 2020.