To appear in IROS 2009

Vision-based estimation of three-dimensional position and pose of multiple underwater vehicles

Sachit Butail and Derek A. Paley

***Abstract*— This paper describes a model-based probabilistic framework for tracking a fleet of laboratory-scale underwater vehicles using multiple fixed cameras. We model the target motion as a steered particle whose dynamics evolve on the special Euclidean group. We provide a likelihood function that extracts three-dimensional position and pose measurements from monocular images using projective geometry. The tracking algorithm uses particle filtering with selective resampling based on a threshold and nearest neighbor data association for multiple targets. We describe results obtained from two tracking experiments: first with one vehicle and a second experiment with two targets. The tracking algorithm for single target experiment is validated using data denial.**

I. INTRODUCTION

Autonomous underwater vehicles have a wide number of applications [11]. Teams of such vehicles can be controlled to cover a large space or perform complicated tasks [22], [18]. In order to validate control strategies we have designed a multi-submarine testbed for use in the University of Maryland Neutral Buoyancy Research Facility (NBRF) [1]. The subs are 1 m long and have a single rear propeller and rear-situated rudder and dive-planes. Because the subs lack the ability to measure or record absolute position, we would like to externally track the position and orientation of each vehicle in three dimensions. In this paper we present a probabilistic tracking framework that uses a body-frame-based motion model and a simple yet robust approximation of target shape to track multiple laboratory-scale underwater vehicles.

An underwater test environment presents several challenges to target tracking not commonly seen on land-based systems, such as changing light conditions, clutter, and internal reflections from the surface. With the availability of small underwater cameras, faster computer processors, and advancement in the field of computer vision, it has become increasingly popular to perform vision-based tracking of underwater vehicles [13], [3], [23]. Alternatives such as GPS and acoustics are less attractive, because GPS signals do not propagate well under water [15] and acoustic sensing is noisy due to scattering in a steel-reinforced test facility [26].

The challenges we seek to address with our tracking system are nonlinear measurement and motion models. A good motion model can improve performance in tracking a maneuvering target [20]. An example of a motion model for a maneuvering target is to model control input as a random process with variance adjustment based on estimation error in measurement space [4]. Another method is to choose between several motion models at each step [4].

Related work on vision-based underwater tracking systems includes using optical flow and disparity measurements in a stereo system [13] and an extended kalman filter (EKF) based vision positioning system [23] that tracks a slow moving target with maximum speed 0.2 m/s.

The contributions of this paper are:

- Selection of a motion model that approximates the 3D dynamics of a self-propelled underwater vehicle
- Apply a method to extract target position and pose from a monocular image
- Implement a probabilistic framework for assimilating visual information from multiple cameras

To model target motion we use a dynamic model for a vehicle subject to steering control [9]. Control inputs (expressed in body frame) are yaw, pitch and roll moments. This model has two advantages: Firstly, by packaging the dynamics into a class of rigid-body transformations we preserve target-state validity despite noisy inputs. Secondly, by making an assumption of low angle of attack we can predict the pose of our target using state estimates.

Target geometry is modeled as a single *quadric* — a quadratic surface in 3D — and we use results from projective geometry to define measurement models for location and pose. For estimation we use particle filtering on the special Euclidean group which preserves state validity during prediction.

We experimentally validate the estimation algorithm using an asynchronous multi-view camera system. We establish a ground-truth dataset for a single target by performing a least-squares fit on data from all cameras. We characterize the performance of the tracking algorithm by running it on a sequence of measurements from a subset of cameras (data denial). We describe the results of tracking two subs using a nearest-neighbor standard filter [4] to associate measurements to targets.

The paper is outlined as follows: Section II provides a theoretical background for motion model of a steered particle, particle filtering on $SE(3)$, and model-based tracking. Section III presents the measurement model in the form of likelihood functions and the tracking algorithm. Section IV describes experimental results. Section V provides conclusions and summarizes ongoing work.

S. Butail is a graduate student in the Department of Aerospace Engineering, University of Maryland, College Park, MD 20742, USA sbutail@umd.edu

D. A. Paley is an Assistant Professor in the Department of Aerospace Engineering, University of Maryland, College Park, MD 20742, USA dpaley@umd.edu

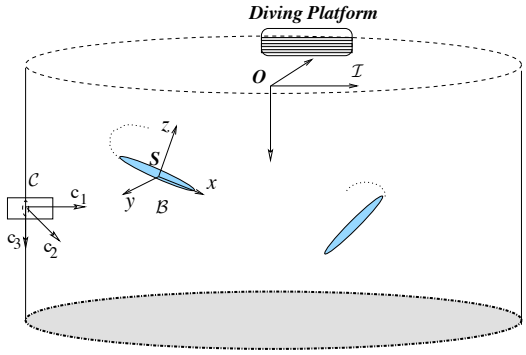

Fig. 1. The University of Maryland Neutral Buoyancy Research Facility is 7.6 m deep and 15 m wide. The inertial frame $\mathcal{I}$, body frame $\mathcal{B}$, and camera frame $\mathcal{C}$ for a single target and a single camera respectively are shown.

II. Probabilistic Data Assimilation and Tracking

In this section we discuss the motion model of a steered target, particle filtering on the special Euclidean group, and model-based tracking.

A. Modeling the motion of a steered target

A self-propelled particle moving with constant speed s under steering control $\boldsymbol{v}$ can be modelled as [22]

$$\dot{\boldsymbol{r}} = s\boldsymbol{x}, \quad \dot{\boldsymbol{x}} = \boldsymbol{v} \times \boldsymbol{x}, \tag{1}$$

where $\boldsymbol{r}$ is the position of the target in inertial frame $\mathcal{I}$ (see Fig. 1) and $\boldsymbol{x}$ is the unit velocity vector of the target with respect to $\mathcal{I}$. The components of the control vector $\boldsymbol{v}$ are $[w, -h, q]^T$, where q and h are curvature controls on yaw and pitch and w is the control on roll motion. In a curve-framing setting [9], under the assumption that $\boldsymbol{r}(t)$ is twice differentiable, an alternate way to represent (1) is to express it in components of a body frame $\mathcal{B} = (\boldsymbol{S}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ fixed to the target. The dynamics are [9], [22]

$$\begin{aligned} \dot{\boldsymbol{r}} &= s\boldsymbol{x} \\ \dot{\boldsymbol{x}} &= \boldsymbol{y}q + \boldsymbol{z}h \\ \dot{\boldsymbol{y}} &= -\boldsymbol{x}q + \boldsymbol{z}w \\ \dot{\boldsymbol{z}} &= -\boldsymbol{x}h - \boldsymbol{y}w. \end{aligned} \tag{2}$$

By attaching a body frame to each target we can relate the dynamics (2) to rigid-body kinematics. The system (2) describes rigid-body motion in four degrees of freedom (translation along $\boldsymbol{x}$, and rotation about $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$). It represents a subset of a group of rigid-body transformations called the special Euclidean group, $SE(3)$. The special Euclidean group includes all real rotations and translations of a rigid body [17]. One way to represent an element g of $SE(3)$ is by a 4×4 matrix $g = \begin{bmatrix} \boldsymbol{x} & \boldsymbol{y} & \boldsymbol{z} & \boldsymbol{r} \\ 0 & 0 & 0 & 1 \end{bmatrix}$. The system dynamics in (2) can be represented as [22]

$$\dot{g} = g\xi, \tag{3}$$

where $\xi = \begin{bmatrix} \hat{\boldsymbol{v}} & s \\ \boldsymbol{0}^T & \boldsymbol{0} \end{bmatrix} \in se(3)$, the Lie algebra of $SE(3)$. $\hat{\boldsymbol{v}}$, a 3×3 skew-symmetric matrix, is the linear operator representing cross product by $\boldsymbol{v}$. $\boldsymbol{0}$ denotes $[0, 0, 0]^T$.

For probabilistic model-based tracking it is common to model unknown inputs as random processes [20]. The stochastic equivalent of (3) can be written as $dg = gdW$ where dW is a standard Wiener process on $se(3)$ [10]. In the context of rigid-body motion, dW is a disturbance input on each degree of freedom of a target. Let $E_i, i = \{1, 2...6\}$, be the basis elements of $se(3)$ [27] and ε_i be a zero-mean Gaussian random variable representing the corresponding variance. For our purposes $\varepsilon_1 = \mathbb{N}(0, \sigma_w^2), \varepsilon_2 = \mathbb{N}(0, \sigma_h^2)$, and $\varepsilon_3 = \mathbb{N}(0, \sigma_q^2)$, and $\varepsilon_4 = \varepsilon_5 = \varepsilon_6 \approx 0$. These values signify the disturbance input in each degree of freedom ($i = 1, 2, 3$ represent motion along roll, pitch, and yaw directions, while $i = 4, 5, 6$ represent translational motion in $\boldsymbol{x}, \boldsymbol{y}$, and $\boldsymbol{z}$ directions). A first-order Euler discretization of the stochastic differential equation $dg = gdW$ with time-step Δ is [10]

$$g[k] = g[k-1] \exp(\sum_{i=1}^{6} E_i \sqrt{\Delta} \varepsilon_i[k-1]) \tag{4}$$

Note that (4) assumes that the motion along each degree of freedom is independent. Writing (2) in this form forces the rigid body to stay on $SE(3)$ at all times despite varying inputs and first-order approximation. In other words, the orthonormality of the body-fixed frame is preserved at every time step.

B. Particle filtering on the special Euclidean group

Particle filtering is a sequential Monte Carlo method used extensively since the early nineties [7]. Its attractiveness over alternative approaches like the extended kalman filter is due to the ability of a particle filter to easily accommodate nonlinearities in measurement and state space. A particle filter can handle non-Gaussian, multi-modal distributions [2]. For example, particle filters have been shown to perform better than EKF for a large class of nonlinear problems [2].

Within a particle filter we use a *likelihood function* to encode our confidence in the information we receive. In its simplest form a likelihood function is a conditional probability $P(Z|X)$ of a measurement Z given state X. A particle filter can easily incorporate additional knowledge about target environment and behavior. For instance, the fact that an underwater vehicle cannot go above the surface of water can be encoded in the likelihood function.

In order to find the output of our particle filter we need to compute averages on the special Euclidean group. An algebraic mean of 4×4 matrices may not itself lie on $SE(3)$. A method of calculating estimates is to compute a mean of multiple rotations based on distance metrics on $SO(3)$ [16] and augment that value with an algebraic mean of location estimates [10]. For a multimodal distribution on $SO(3)$ we need to compute modes on a group of rotations. This can be accomplished by using mean shift algorithm [25] which involves computing matrix exponentials for each data point. However, our target rolls only a few degrees about its center line. We compute the mode by calculating a simple mode $\bar{\bar{\boldsymbol{x}}}$ of the velocity vectors. We then compute a cross product of the vertical axis in the inertial frame with $\bar{\bar{\boldsymbol{x}}}$ to get the $\bar{\bar{\boldsymbol{y}}}$

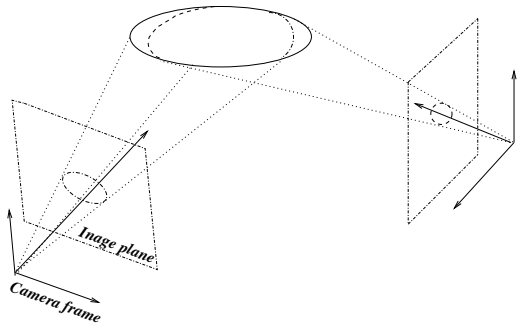

Fig. 2. Projection of a 3D ellipse onto two image planes. Notice that the projection on an image plane depends on the position of the camera and the position and orientation of the object.

direction in the body frame. The orthogonal body frame is completed by setting $\bar{\tilde{\boldsymbol{z}}} = \bar{\tilde{\boldsymbol{x}}} \times \bar{\tilde{\boldsymbol{y}}}$.

C. Model-based tracking using target geometry

Using projective geometry, the 3D position of a target can be estimated by ray-tracing the target centroid without knowing details about the target geometry. Prior knowledge about a target can aid in pose estimation. A common and relatively accurate approach involves tracking feature points in successive frames [14], but it is difficult to track features in noisy images of fast-moving targets. A simple yet robust method is to model the target as a series of connected quadratic surfaces or *quadrics* [24]. A quadric is a 2D surface defined by the equation $\tilde{\boldsymbol{r}}^T Q \tilde{\boldsymbol{r}} = 0$, where Q is a 4×4 symmetric matrix and $\tilde{\boldsymbol{r}} \triangleq [r_1, r_2, r_3, 1]^T$ is the homogeneous coordinate of a point on the surface of the quadric. The following properties of a quadric surface are relevant [8]:

1) A quadric has 10 degrees of freedom, three each for position, orientation, and shape and one for scale.
2) The intersection of a plane with a quadric is a set of points satisfying $\tilde{\boldsymbol{u}}^T C \tilde{\boldsymbol{u}} = 0$ where C is a 3×3 symmetric matrix, and $\tilde{\boldsymbol{u}} \triangleq [u, v, 1]^T$ is the homogeneous coordinate of a point on the plane.

We represent the matrix Q as $\begin{bmatrix} Q_{3\times3} & Q_4 \\ Q_4^T & Q_{44} \end{bmatrix}$ where $Q_{3\times3} \in \mathbb{R}^{3\times3}$, $Q_4 \in \mathbb{R}^3$ and $Q_{44} \in \mathbb{R}$. Let there be a vector $\boldsymbol{l} = [l_1, l_2, l_3]^T$ in the camera centered frame such that a line $L(t) = \boldsymbol{l}t$ that passes through the quadric surface will intersect the surface at all points satisfying [5]

$$\begin{bmatrix} L(t)^T & 1 \end{bmatrix} \begin{bmatrix} Q_{3\times3} & Q_4 \\ Q_4^T & Q_{44} \end{bmatrix} \begin{bmatrix} L(t) \\ 1 \end{bmatrix} = 0. \tag{5}$$

If $L(t)$ is tangent to the quadric surface, then (5) will have a single solution for t. In this case, the discriminant of (5) satisfies

$$\boldsymbol{l}^T [Q_4 Q_4^T - Q_{44} Q_{3\times3}] \boldsymbol{l} = 0. \tag{6}$$

Equation (6) defines a conic on a plane. Using it to solve (5) gives a single value of t as $-\boldsymbol{l}^T Q_4 (\boldsymbol{l}^T Q_{3\times3} \boldsymbol{l})^{-1}$ at which $L(t)$ touches the quadric. Normalizing $\boldsymbol{l}$ with respect to l_3 in equation (6), we get a conic on an image plane at unit focal length [6], [5]

$$\tilde{\boldsymbol{u}}^T [Q_4 Q_4^T - Q_{44} Q_{3\times3}] \tilde{\boldsymbol{u}} = 0. \tag{7}$$

We use the measured conic to estimate pose in a particle-filtering framework, described next.

III. Vision-based estimation of position and pose

In this section we describe the likelihood functions and particle-filtering algorithm for our tracking system.

A. Likelihood function for a monocular image

Consider a monocular image of a target where $\boldsymbol{u} = [u, v]^T$ are coordinates of centroid of the target in the image plane. Let $\boldsymbol{r} = \begin{bmatrix} r_1, r_2, r_3 \end{bmatrix}^T$ be the position of a point in the inertial frame. Any point in the inertial frame can be transformed to camera frame coordinates by a transformation matrix $\begin{bmatrix} R & \boldsymbol{t} \end{bmatrix}$ as $\tilde{\boldsymbol{r}}^{\mathcal{C}} = \begin{bmatrix} R & \boldsymbol{t} \end{bmatrix} \tilde{\boldsymbol{r}}$. Without loss of generality, we assume that the camera frame is aligned with the inertial frame so that $\begin{bmatrix} R & \boldsymbol{t} \end{bmatrix} = \begin{bmatrix} I & \boldsymbol{0} \end{bmatrix}$.

Position estimate: The centroid measurement on a camera image plane with focal length f (in pixels) as a function of target center position $\boldsymbol{r}$ is

$$\begin{bmatrix} u \\ v \end{bmatrix} = f \begin{bmatrix} r_1/r_3 \\ r_2/r_3 \end{bmatrix} + D\eta \tag{8}$$

where η is a two dimensional Gaussian noise vector for u and v and $D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. The estimates of r_1 and r_2 depend on r_3, the distance of a point along the optical axis. This uncertainty is inherent to monocular images and, given a noise covariance matrix in measurements, impacts the uncertainty in r_1 and r_2. We also include the information that our target cannot go outside the test environment in the position likelihood function.

The likelihood function for location of a single measurement $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{r})$ is

$$P_{loc}(\boldsymbol{u}(\boldsymbol{r})|\boldsymbol{r}) = \begin{cases} \mathbb{N}(\boldsymbol{u}(\boldsymbol{r}); \boldsymbol{u}, \Sigma)\mathbb{U}_{tank} & \text{if detected} \\ \mathbb{U}_{tank} & \text{otherwise.} \end{cases} \tag{9}$$

$\mathbb{N}(\boldsymbol{u}(\boldsymbol{r}); \boldsymbol{u}, \Sigma)$ denotes a normal distribution function over the image plane with mean $\boldsymbol{u}$ and noise covariance matrix Σ. $\mathbb{U}_{tank}$ denotes a uniform distribution within the tank. For example, consider the inertial frame $\mathcal{I}$ shown in Fig. 1. Given a polar representation of a 3D point $\boldsymbol{r} = [r, \varphi, z]^T$, we have

$$\mathbb{U}_{tank} = \mathbb{U}(r; 0, T_d/2)\mathbb{U}(z; 0, T_h)\mathbb{U}(\varphi; -\pi, \pi)$$

T_d and T_h are the tank diameter and tank height respectively.

Pose estimate: We make the following assumptions about the shape and motion of each submarine:

- We model the sub as an ellipsoid with semi-major, -medium and -minor axes of length 0.4889 m, 0.0665 m, 0.0635 m, respectively.
- The sub motion has a low angle of attack which implies that the pose is aligned with the body frame $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$.
- The image used to estimate pose has no occlusions.

These assumptions allow us to estimate the pose of a sub using a quadric surface and its conic image projection.

We use MATLAB^{TM} image processing toolbox to fit an ellipse around the target region and extract the following measurement parameters: (1) Image-plane coordinates of the target centroid, $\boldsymbol{u}$; (2) The orientation of the bounding ellipse, θ; (3) Eccentricity of the bounding ellipse, ϵ.

If the length of the ellipsoid axes in orthogonal directions are denoted as $2a, 2b, 2c$, the equation of an ellipsoid in the body-frame $\mathcal{B}$ is given by $(r_1^{\mathcal{B}})^2/a^2+(r_2^{\mathcal{B}})^2/b^2+(r_3^{\mathcal{B}})^2/c^2 = 1$. In matrix form, the quadric equation is

$$\tilde{\boldsymbol{r}}^{\mathcal{B}^T} Q^{\mathcal{B}} \tilde{\boldsymbol{r}}^{\mathcal{B}} = 0,\ Q^{\mathcal{B}} = \begin{bmatrix} 1/a^2 & 0 & 0 & 0 \\ 0 & 1/b^2 & 0 & 0 \\ 0 & 0 & 1/c^2 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}. \quad (10)$$

Assuming the $\boldsymbol{y}$ and $\boldsymbol{z}$ vectors lie on the semi-medium and semi-minor axes of the ellipsoid, the pose of the sub can be represented as $R = \begin{bmatrix}\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\end{bmatrix}$. For any $\boldsymbol{r}^{\mathcal{B}}$ and $\boldsymbol{r}$, given $T = \begin{bmatrix} R & \boldsymbol{S} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$, and $T^{-1} = \begin{bmatrix} R^T & -R^T\boldsymbol{S} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$ we have $\tilde{\boldsymbol{r}}^{\mathcal{B}} = T^{-1}\tilde{\boldsymbol{r}}$. Using (10), we have

$$\tilde{\boldsymbol{r}}^T Q^{\mathcal{C}} \tilde{\boldsymbol{r}} = 0,\ Q^{\mathcal{C}} = (T^{-1})^T Q^{\mathcal{B}} T^{-1}. \quad (11)$$

The matrix $Q^{\mathcal{C}}$ represents an ellipsoid in the camera frame and projects an ellipse onto the image plane up to a scale factor. The orientation and eccentricity of the ellipse that is projected onto the image plane is compared to the eccentricity and orientation of the measured elliptical contour. Assuming a normal distribution in measurement, a likelihood function is computed for the estimated values.

As per (7), $\tilde{\boldsymbol{u}}^T C \tilde{\boldsymbol{u}} = 0$ where $C = K[Q_4^{\mathcal{C}} Q_4^{\mathcal{C}^T} - Q_{44}^{\mathcal{C}} Q_{3\times 3}^{\mathcal{C}}]$ and K makes $C_{2\times 2}$, the upper left 2×2 matrix in C, positive definite. $C_{2\times 2}$ represents an ellipse with eccentricity and orientation as follows

$$\epsilon_m = \sqrt{1 - \frac{\lambda_{min}}{\lambda_{max}}},\ \ \theta_m = \mathrm{atan}(\mathrm{v}_2/\mathrm{v}_1), \quad (12)$$

where λ_{min} and λ_{max} are the eigenvalues of $C_{2\times 2}$ and $\boldsymbol{v} = \begin{bmatrix} v_1 & v_2 \end{bmatrix}^T$ is the eigenvector for λ_{min}.

We compare ϵ_m, θ_m to our measurements of the bounding ellipse and build a likelihood function. The likelihood function for pose is $P_{pose} = P_\epsilon P_\theta$, where

$$P_\epsilon(\epsilon|\boldsymbol{r}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \mathbb{N}(\epsilon_m; \epsilon, \sigma_\epsilon^2)$$
$$P_\theta(\theta|\boldsymbol{r}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \mathbb{N}(\theta_m; \theta, \sigma_\theta^2).$$

The uncertainty in our measurements, $(\sigma_\epsilon, \sigma_\theta)$, is due to the position measurement of each pixel projected as a subsurface on the image. We assume Gaussian noise with covariances determined experimentally (see Table I). Note that the pose likelihood function has a forward-backward ambiguity since we do not have information of where the sub is pointing. This results in a bimodal distribution function, which is why we use the mode, and not mean, on rotations.

B. Particle filter tracking algorithm

We use a generic particle filter [2] to fuse the vision-based likelihood function with self-propelled motion model. For a target, our state vector $\boldsymbol{X}$ consists of position and orientation, described by $g \in SE(3)$, and speed s, which is assumed constant. The particle filtering algorithm is as follows:

i. *Initialize*: Generate N uniformly distributed samples on $SE(3)$ according to

$$g = \exp(\mathbb{U}(-\pi,\pi)E_1 + \mathbb{U}(-\pi,\pi)E_2 + \mathbb{U}(-\pi,\pi)E_3 + \mathbb{U}(-T_d,T_d)E_4 + \mathbb{U}(-T_d,T_d)E_5 + \mathbb{U}(0,T_h)E_6)$$

and set the state variable to $\boldsymbol{X} = \begin{bmatrix}\boldsymbol{r}^T & \boldsymbol{x}^T & \boldsymbol{y}^T & \boldsymbol{z}^T & s\end{bmatrix}^T$.

ii. For each time step k:

a) *Propagate*: Evolve each sample according to (4), using normal random variables as inputs with standard deviations σ_q, σ_h, and σ_w. These values are based on how the target moves and satisfy $\sigma_q > \sigma_h \gg \sigma_w > 0$ (The sub turns more than it pitches or rolls). The speed s is propagated as $s[k] = s[k-1] + d_s$, where $d_s = \mathbb{N}(0, \sigma_s^2)$.

b) *Predict*: For each sample j compute the weights using the product of likelihood functions for location (9) and pose (13), and normalize

$$\tilde{w}_j = P_{loc}P_{pose},\ w_j = \tilde{w}_j[\sum_{i=1}^{N} \tilde{w}_i]^{-1}$$

iii. *Resample*: Estimate effective sample size $N_{eff} = (\sum_{j=1}^{N} \tilde{w}_j)^{-1}$. If this value is less than a threshold N_T, resample using normalized weights w_j. The value for N_T was set to $N/2$ in our algorithm.

iv. *Estimate*: Compute the sample estimate by augmenting the algebraic mean of position with mode on rotations (see Section II-B)

IV. Experimental Methods and Results

A. Experimental setup

We describe tracking results from two deployments of remote-controlled submarines in the University of Maryland Neutral Buoyancy Research Facility, which is 7.6 m deep and 15 m wide. Five high-resolution ZC-YHW701N GANZ cameras, each with an approximate wide viewing angle of 36 degrees and vertical viewing angle of 32 degrees were used. All cameras are mounted on the inside wall of the tank with three cameras at mid-level depth of 3.81 m spaced at 90 degree intervals looking straight into the center of the tank. The remaining two cameras are at 45 degrees interval to the mid-level ones just below the surface of water at 0.61 m looking down at an angle of approximately 15 degrees.

The image acquisition was carried out using FlashBus^{TM} framegrabbers on three separate computers. The acquisition was not synchronous and the time delay between successive frames depended on which cameras were used for measurements. With all five cameras the time delay was an average 70 ms, while with three cameras the time delay was 120 ms. The tracker was fed measurements as they came and sampling time interval Δ was computed at each iteration.

Calibration of cameras was done using a non-coplanar arrangement of balls [23] and the Tsai camera-calibration

TABLE I

PARAMETER VALUES USED FOR TRACKING

Parameter	Avg. value	Value used	Parameter	Value used
σ_u(pixels)	0.1	4.0	σ_q(rad/s)	1.00
σ_v(pixels)	0.05	4.0	σ_h(rad/s)	0.30
σ_ϵ	1.0×10^{-5}	0.15	σ_w(rad/s)	0.08
σ_θ(radians)	0.08	0.25	σ_s	0.02

software [28]. To mitigate the effect of bubbles and changing lighting conditions, the background was updated as a running average [19], $G[k+1] = \alpha T[k] + (1-\alpha)G[k]$, where $T[k]$ is the current image, $G[k]$ is the background image, and the value used for α is 0.05. $G[0]$ is an image without any targets taken just before the experiment is started.

Measurement-noise parameters for each camera were calculated by computing variance in values of centroid position, eccentricity, and orientation of a still sub in water over 300 images. These static values, however, only give a lower bound to our error variances. Different parts of tank have different lighting conditions, and bubbles from propeller as the sub moves through water creates clutter. The actual values used in tracker, therefore, were much larger (see Table I). The submarine speed was set 0.5 m/s.

Single target: To establish ground truth in the single sub trial we use a least squares estimate to minimize the pixel error in all five cameras treating 3 successive measurements as synchronous. We project the least squares estimate back on to the image planes and verify that it also lies on the sub. We then smooth the estimate using moving averages with a span of five data points. Note that any errors inherent in the camera measurement system will not be detected by this method. An alternate way would be to verify the estimates using a different sensor that is not part of the measurement system.

Multi-target: In the multi-target tracking experiment we deployed two subs in the NBRF. The subs are tracked for ten seconds using three cameras. Data association is performed at each time-step using the nearest neighbor filter [4]. The measurement that minimizes the weighted distance between a measurement and a projected estimate of a target on the image plane is assigned to that target.

B. Experimental results

Single target: For a single sub we use a ground truth dataset to characterize tracker performance. Measurements are taken from three cameras such that the sub is at least viewed in two cameras. All three cameras (two top-level and one mid-level) span an angle of 135 degrees. A comparison with the ground-truth is shown in Fig. 3. The error in each orthogonal direction is less than a meter. For pose estimates we compare the orientation and eccentricity of our estimate projected back on to the image plane with the actual measurement taken at that time step (Fig. 4). Orientation and eccentricity are only used for weighting the samples if the sub is detected well within the frame and not at it's edges. Error in eccentricity is generally high compared to that in orientation. This is primarily due to clutter that prevents a tight ellipse fit around the sub contour. We also show frames with the projected ellipse overlaid on the sub.

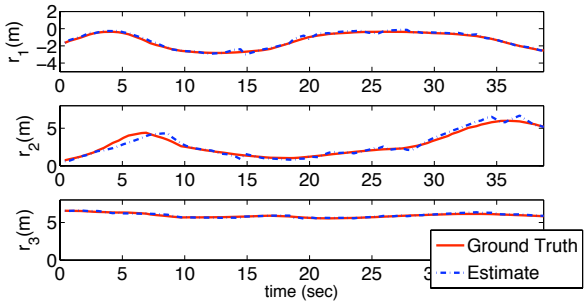

Fig. 3. Position estimates for a single target. The ground truth was created using measurements from five cameras. The tracker was run on measurements from three cameras.

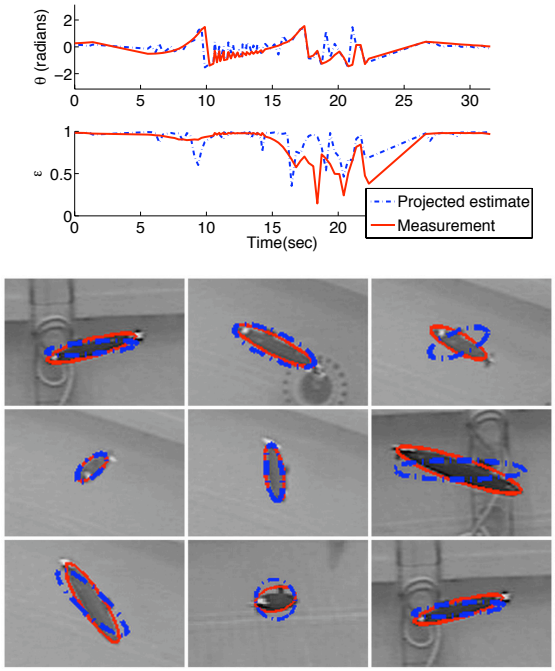

Fig. 4. Pose estimates. The plots on top show orientation and eccentricity values for time-steps when the sub was completely visible in the camera. The measured values are compared with the projected values of the estimate on the image plane of the same camera. Also shown are nine frames with isolated target overlaid with a scaled ellipse with the same eccentricity and orientation as the estimate projected on to the image plane.

Multi-target: For multi-target tracking we do a visual comparison of measurements against estimates projected back onto the image plane on all three cameras where the subs were seen. Two of those cameras ar shown (See Fig. 5). Tracks of individual targets are maintained during occlusions. Direction of motion in the last frame shown is computed using pose estimates.

C. Discussion

Single target: For the single target tracking case, we obtain estimates within the body length of the target (1 m). The difference in eccentricity and orientation measurements is due to the sensitivity of these values to clutter in the image. There were several instances when the sub appeared as a disconnected region in the image after background

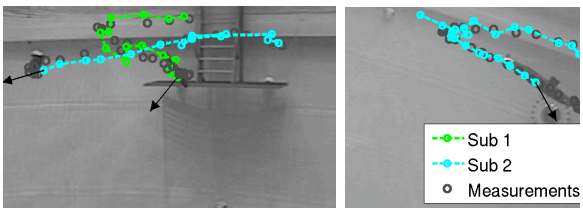

Fig. 5. Image frames from two cameras are shown. Estimates are projected back on to the frame along with measurements. Black arrows representing estimated direction of motion at the time step for the frame are also shown. The cameras are orthogonal to each other. The second camera had only one sub in view.

subtraction. Furthermore a tight bound around the sub is required for precise measurements. In the case of noisy images this was not the case. In addition, pose estimate is based on low angle-of-attack assumption which is violated when the sub looses RF communication and automatically cuts off the propeller, floating to the surface.

Multi-target: The multi-target test contained challenges that we seek to address in ongoing work including occlusions and reflections. One way to handle reflections is to track every object that is detected after a mild background subtraction and then probabilistically identify the target [21]. Data association using nearest-neighbor distance is sensitive to clutter and affected the estimate at instances when there were bubbles in the tank.

V. Conclusion

We describe a model-based probabilistic framework to track multiple underwater vehicles in a test environment. The framework combines a three-dimensional motion model for steered vehicles with a likelihood function that assimilates monocular image data based on target geometry and behavior. We model the target shape as an ellipsoid to augment our estimates of pose and velocity. Results are described from an experiment using multiple targets in the university Neutral Buoyancy Research Facility.

There are at least three research directions to improve tracking performance. Firstly, to make it robust to a variety of maneuvers a stack of motion models inheriting from the same dynamics can be used [4]. Secondly, we can model the input variances as part of the state vector and perform combined parameter estimation [12]. Thirdly in order to track more targets within clutter a data association scheme such as Joint Probabilistic Data Association Filter (JPDAF) [4] may be used.

VI. ACKNOWLEDGMENTS

We gratefully acknowledge the Space Systems Laboratory at University of Maryland, especially Kate McBryan for getting the image-acquisition system to work. Thanks to Adam Mirvis, Nick Limparis and Massimiliano Di Capua for experiment support. Thanks to our own team at Collective Dynamics and Control Laboratory, Seth Napora, Nitin Sydney, Billy Lang, Sarah Beal, Steve Sherman, Alexander Leishman and Adam Reese for building the subs and running the trials.

References

[1] Neutral Buoyancy Research Facility at Space Systems Laboratory, University of Maryland http://www.ssl.umd.edu/html/facilities.html.

[2] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Processing*, 50(2):174–188, Feb 2002.

[3] B.A.A.P. Balasuriya, M. Takai, W.C. Lam, T. Ura, and Y. Kuroda. Vision based autonomous underwater vehicle navigation: underwater cable tracking. *OCEANS '97. MTS/IEEE Conference Proceedings*, 2:1418–1424 vol.2, Oct 1997.

[4] Y. Bar-Shalom. *Tracking and data association*. Academic Press Professional, Inc., San Diego, CA, USA, 1987.

[5] Geoffrey Cross and Andrew Zisserman. Quadric reconstruction from dual-space geometry. *IEEE Int. Conf. on Computer Vision*, 0:25, 1998.

[6] Song De Ma. Conics-based stereo, motion estimation, and pose determination. *Int. J. Comput. Vision*, 10(1):7–25, 1993.

[7] Nando De Freitas and Neil Gordon. *Sequential Monte Carlo Methods in Practice*. Birkhuser, 2001.

[8] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, March 2004.

[9] E.W. Justh and P.S. Krishnaprasad. Natural frames and interacting particles in three dimensions. *IEEE Conf. on Decision and Control*, pages 2841–2846, Dec. 2005.

[10] J. Kwon, M. Choi, F. C. Park, and C. Chun. Particle filtering on the Euclidean group: framework and applications. *Robotica*, 25(6):725–737, 2007.

[11] Naomi Ehrich Leonard, Derek Paley, Francois Lekien, Rodolphe Sepulchre, David Fratantoni, and Russ Davis. Collective motion, sensor networks, and ocean sampling. *IEEE Proc., Emerging technology of networked control systems*, (95):48–74, 2007.

[12] J. Liu and M. West. Combined parameter and state estimation in simulation-based filtering. 2000.

[13] R.L. Marks, S.M. Rock, and M.J. Lee. Automatic object tracking for an unmanned underwater vehicle using real-time image filtering and correlation. *Int. Conf. on Systems, Man and Cybernetics*, pages 337–342 vol.3, Oct 1993.

[14] Frederick Martin and Radu Horaud. Multiple-Camera Tracking of Rigid Objects. *The Int. J. of Robotics Research*, 21(2):97–113, 2002.

[15] D.T. Meldrum and T. Haddrell. Gps in autonomous underwater vehicles. *Int. Conf. on Electronic Engineering in Oceanography*, pages 11–17, Jul 1994.

[16] M. Moakher. Means and averaging in the group of rotations. *SIAM Journal on Matrix Analysis and Applications*, 24(1):1–16, 2002.

[17] R. M. Murray, S. S. Sastry, and L. Zexiang. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., 1994.

[18] Derek A. Paley. Stabilization of collective motion on a sphere. *Automatica*, 45(1):212 – 216, 2009.

[19] M. Piccardi. Background subtraction techniques: a review. *Int. Conf. on Systems, Man and Cybernetics*, 4:3099–3104, Oct 2004.

[20] X. Rong Li and V.P. Jilkov. Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Trans. Aerospace and Electronic Systems*, 39(4):1333–1364, Oct. 2003.

[21] D.J. Salmond and H. Birch. A particle filter for track-before-detect. *In Proc. of American Control Conference*, 5:3755–3760 vol.5, 2001.

[22] L. Scardovi, N.E. Leonard, and R. Sepulchre. Stabilization of collective motion in three dimensions: A consensus approach. *IEEE Conf. on Decision and Control*, pages 2931–2936, Dec. 2007.

[23] J. R. Smithanik, E. M. Atkins, and R. M. Sanner. Visual positioning system for an underwater space simulation environment. *Journal of Guidance, Control, and Dynamics*, 29:858–869, 2006.

[24] B. Stenger, P.R.S. Mendonca, and R. Cipolla. Model-based 3d tracking of an articulated hand. *In Proc. IEEE Computer Vision and Pattern Recognition CVPR*, 2:II–310–II–315 vol.2, 2001.

[25] O. Tuzel, R. Subbarao, and P. Meer. Simultaneous multiple 3d motion estimation via mode finding on Lie groups. In *Proc. IEEE Int. Conf. on Computer Vision ICCV*, volume 1, pages 18–25, October 17–21, 2005.

[26] K. Vickery. Acoustic positioning systems. a practical overview of current systems. *Proc. of 1998 Workshop on Autonomous Underwater Vehicles*, pages 5–17, Aug 1998.

[27] Yunfeng Wang and G.S. Chirikjian. Error propagation on the euclidean group with applications to manipulator kinematics. *IEEE Trans. on Robotics*, 22(4):591–602, Aug. 2006.

[28] Reg Willson. Tsai camera calibration software http://www.cs.cmu.edu/ rgw/tsaicode.html, 2008.