AUV/ROV POSE AND SHAPE ESTIMATION OF TETHERED TARGETS WITHOUT FIDUCIALS

Sean Augenstein[1] saugenst@stanford.edu

Stephen Rock[1,2] rock@stanford.edu

[1] Dept. of Aeronautics and Astronautics
Stanford University
496 Lomita Mall
Stanford, CA 94305

[2] Monterey Bay Aquarium Research Institute
7700 Sandholdt Road
Moss Landing, CA 95039

Abstract

This paper presents an algorithm (SPEAR) for visually estimating the pose and shape of an underwater target without fiducials. The goal is to enable AUV/ROV stationkeeping with respect to targets about which nothing is known *a priori*. It will also enable relative positioning with respect to any face on the 3-D body. It uses a particle filter to simultaneously track the pose and reconstruct a 3-D point cloud model of the target, based on concepts from the SLAM community (in particular, the FastSLAM algorithm [1], [2]).

The algorithm for target estimation is explained and applied to the case of monocular sensing. Results are given showing successful estimation of an underwater target's pose along with the 3-D locations of features on the body, based on experiments conducted in Monterey Bay, California with the ROV *Ventana* and a tethered underwater target.

1 Introduction

This paper presents a novel method for visual estimation of an unknown underwater object (the 'target'), relative to an AUV/ROV. The algorithm continually estimates relative pose to handle rotating and translating targets (an issue since tethered targets can move unpredictably due to ocean currents). In addition, the algorithm forms and updates an estimate of the 3-D shape of the body (i.e., 'reconstruction'), so tracking/proximity operations are not constrained by a single view of the target.

This research is motivated by the need for AUV/ROV stationkeeping on objects with no fiducials, or in situations where existing markers have been obscured (for example, by biogrowth). Beyond fiducial-less stationkeeping, a higher-level goal is the ability to command a vehicle to hold station with respect to any particular face on a 3-D target, and slew between distinct faces, as opposed to being limited to one surface. For this reason, online 3-D reconstruction takes place concurrently with relative pose tracking.

The algorithm presented in this paper is inspired by algorithms from the Simultaneous Localization and Mapping (SLAM) field. For this reason, the algorithm is referred to as SLAM-inspired Pose Estimation and Reconstruction ('SPEAR') [3]. The SPEAR framework is broad and of use with any vision sensing configuration (i.e, single or multiple cameras). It can be used with either stereo or monocular vision systems. Stereo provides depth data, but involves more complicated image correspondence (matching features between the camera pair *as well as* matching features over time). Monocular vision is simpler, but estimation techniques must overcome the lack of range data in a single measurement. The choice was made in this research to focus on monocular sensing. Figure 1 illustrates the sensing of a tethered target.

In place of fiducials, the work presented here uses natural image features. While the algorithm is independent of the particular choice of image point features used, SIFT [4] was used due to its invariance properties (which allow for feature tracking/matching through a variety of transformations). SIFT has also been applied previously to robotic ap-

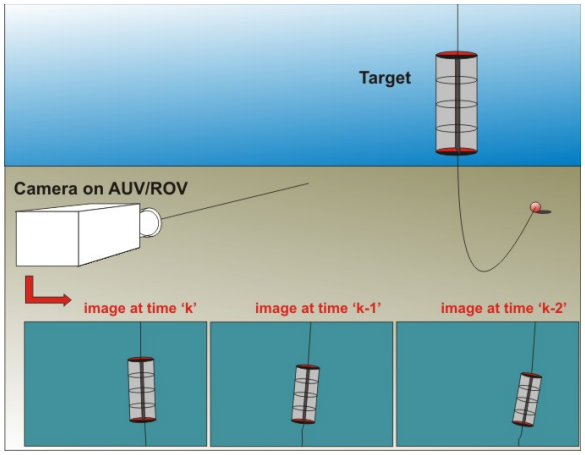

Figure 1: AUV/ROV Imaging an U/W Target

plications in the underwater environment, including SLAM [5] and AUV navigation [6].

2 Background

Much previous vision-based AUV/ROV relative positioning research has relied on known fiducials on the object [7,8]. This necessarily constrains station-keeping to specific targets. Recently an alternative algorithm has been presented [9] in which a user-selected group of natural SIFT features act as the reference marker. It is intended for ROV teleoperation about stationary targets. These methods use for reference a single face of the target, which must remain in view of the camera to determine relative pose.

For broader capability, i.e., the ability to hold station with respect to any face, or slew from one face to another, some information on the shape is necessary. This could come in the form of a CAD model; however, this constrains station-keeping capability to objects for which *a priori* information is available. Alternatively, the 3-D shape of the target can be visually reconstructed. As the targets are rotating and translating on tethers in the midwater, this reconstruction takes place concurrently with (and thus factors in) the relative pose estimation.

The problem of bearings-only reconstruction of an object up to a scale factor, along with the relative poses of the camera, is addressed in the field of Structure-from-Motion (SFM). However, much of the SFM research focuses on offline, batch processing, to find an optimal solution for the shape and poses based on all the images acquired [10]. While the solution obtained is accurate, an offline batch algorithm is of no use in a real-time estimation/control loop.

There are also SFM algorithms that recursively online process each new image, and might be applied in a real-time situation. The dominant approach to performing recursive SFM has been to use extended Kalman filters (EKFs) [11–13]. However, the selection of an EKF makes a constraining assumption on the probability distribution of the states of the filter, namely, that they are Gaussian-like (i.e., unimodal). In addition, many of these methods have various computational drawbacks, such as requiring inversion of large matrices and/or requiring a batch process for initialization.

More recently, research has been done on particle filter-based SFM, which relaxes the assumption of a Gaussian posterior. Previous particle filter methods either use fiducials for state initialization [14] and periodic state correction [15], or are motivated by issues of unknown feature correspondence between images [16] and have not run in real-time.

The SPEAR algorithm presented here is a SLAM-inspired, efficient particle-filter approach to solving the problem of real-time recursive reconstruction and pose estimation without fiducials or other *a priori* information. Specifically, the approach is based on the FastSLAM algorithm [1,2,17]. It uses a type of particle filter (known as a Rao-Blackwellized Particle Filter [18]) which efficiently partitions the state space, and appears not to have been used previously in the SFM field.

SPEAR takes advantage of the strong structural similiarites between SLAM and the pose estimation and reconstruction task. SLAM refers to the task of determining own-vehicle location/pose along with a map of the vehicle's surroundings [19]. Where SLAM algorithms are concerned with processing measurements to form a map of the external world, SPEAR is concerned with processing measurements to form a 3-D 'map' of a distinct external target. Where SLAM algorithms are concerned with the kinematic states of the own-robot making the measurements, SPEAR is concerned with the kinematic states of the external target.

3 Problem Definition

SPEAR entails two concurrent estimation problems: *pose estimation*, to determine the kinematic states $\overline{s}$ (which evolve over time), and *reconstruction*, to determine the 3-D location of the features in the body frame, $\mathbf{X}$ (constants, since a rigid body is assumed). These are discussed in turn. For notation, the discrete time steps are indexed by k, and the features

are distinguished by index j.

3.1 Pose Estimation

Pose estimation refers to the filtering of the target's kinematic state $\overline{s}$ as it evolves over time. Explicitly, the kinematic state vector at time k is:

$$\overline{s}_k = \left[\, \phi \;\; \theta \;\; \psi \;\; \omega_{x_R} \;\; \omega_{y_R} \;\; \omega_{z_R} \;\; x_t \;\; y_t \;\; z_t \;\; \dot{x}_t \;\; \dot{y}_t \;\; \dot{z}_t \,\right]_k^T \quad (1)$$

The Euler angles ϕ, θ, and ψ give the rotation to the body frame from the reference frame; ω_{x_R}, ω_{y_R}, and ω_{z_R} form the target's angular rate vector, expressed in the reference frame; x_t, y_t, z_t, $\dot{x}_t$, $\dot{y}_t$, $\dot{z}_t$ are the position and velocity of the target's center of rotation in the reference frame. The reference frame can be either the camera's frame (in which case the position is in non-dimensional coordinates), or an inertial frame if position/orientation sensors are available on the observing AUV/ROV.

The kinematic state's time evolution is modeled by the following kinematic motion update equation:

$$\overline{s}_k = f(\overline{s}_{k-1}) + \overline{n}_p$$

$$\begin{bmatrix} \phi \\ \theta \\ \psi \\ \omega_{x_R} \\ \omega_{y_R} \\ \omega_{z_R} \\ x_t \\ y_t \\ z_t \\ \dot{x}_t \\ \dot{y}_t \\ \dot{z}_t \end{bmatrix}_k = \begin{bmatrix} \phi_{(k-1)} + \Delta t * \dot{\phi}_{(k-1)} \\ \theta_{(k-1)} + \Delta t * \dot{\theta}_{(k-1)} \\ \psi_{(k-1)} + \Delta t * \dot{\psi}_{(k-1)} \\ \omega_{x_R(k-1)} \\ \omega_{y_R(k-1)} \\ \omega_{z_R(k-1)} \\ x_{t(k-1)} + \Delta t * \dot{x}_{t(k-1)} \\ y_{t(k-1)} + \Delta t * \dot{y}_{t(k-1)} \\ z_{t(k-1)} + \Delta t * \dot{z}_{t(k-1)} \\ \dot{x}_{t(k-1)} \\ \dot{y}_{t(k-1)} \\ \dot{z}_{t(k-1)} \end{bmatrix} + \overline{n}_p \quad (2)$$

where

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_k = \begin{bmatrix} 1 & tan(\theta_k)sin(\phi_k) & tan(\theta_k)cos(\phi_k) \\ 0 & cos(\phi_k) & -sin(\phi_k) \\ 0 & \frac{sin(\phi_k)}{cos(\theta_k)} & \frac{cos(\phi_k)}{cos(\theta_k)} \end{bmatrix} \mathbf{R}^{B/R} \begin{bmatrix} \omega_{x_R(k)} \\ \omega_{y_R(k)} \\ \omega_{z_R(k)} \end{bmatrix} \quad (3)$$

In Eq. 2, $\overline{n}_p$ is additive noise, with Gaussian distribution $\mathcal{N}(0, P_{[4:12,4:12]})$ on the rates, position, and velocity components of $\overline{s}_k$. A Fisher distribution ($\mathcal{F}(\overline{\mu}, \kappa)$), a probability model for directions in space [20], is used as the basis for applying noise on the Euler angle components of $\overline{s}_k$. This ensures perturbations to the target's 3-D orientation are identically distributed over the entire range of Euler angle parameters (further discussed in [3]).

In Eq. 3, $\mathbf{R}^{B/R}$ is the rotation matrix which rotates vectors from the reference frame R into the body frame B.

3.2 Reconstruction

SPEAR also reconstructs the target's 3-D shape; this is represented as a point cloud, with the points being the locations where the observed SIFT features sit on the target. A key assumption is that the target is a rigid body, so that the feature locations are fixed with respect to each other.

The matrix of feature locations on the body is:

$$\begin{aligned} \mathbf{X} &= \begin{bmatrix} X_1 & X_2 & \cdots & X_j & X_{j+1} & \cdots \end{bmatrix} \\ \mathbf{X} &= \begin{bmatrix} x_1 & x_2 & & x_j & x_{j+1} & \\ y_1 & y_2 & \cdots & y_j & y_{j+1} & \cdots \\ z_1 & z_2 & & z_j & z_{j+1} & \end{bmatrix} \end{aligned} \quad (4)$$

This matrix grows as new features are initialized.

As the object is observed by a calibrated camera, SIFT features are extracted from the images. These measurements ($\overline{z}$) are in pixel coordinates (u, v). A pinhole model is assumed:

$$\begin{aligned} \overline{z}_j &= g(\overline{s}, X_j) + \overline{n}_c \\ \begin{bmatrix} u_j \\ v_j \end{bmatrix} &= \frac{foc}{z^C_{j/cam}} \begin{bmatrix} x \\ y \end{bmatrix}^C_{j/cam} + \overline{n}_c \end{aligned} \quad (5)$$

In Eq. 5, $\overline{n}_c$ is additive Gaussian noise with distribution $\mathcal{N}(0, r^2 I)$.

The relationship between the feature location w.r.t. the camera in the camera frame ($\overline{X}^C_{j/cam}$) and the feature location w.r.t. the target in the target's body frame ($X_j = \overline{X}^B_{j/target}$) is given by:

$$\begin{aligned} \overline{X}^C_{j/cam} &= \mathbf{R}^{C/R} * \overline{X}^R_{j/cam} \\ &= \mathbf{R}^{C/R} * (\overline{X}^R_{j/target} + \overline{t}_{target/cam}) \\ &= \mathbf{R}^{C/R} * (\mathbf{R}^{R/B} * \overline{X}^B_{j/target} + \overline{t}_{target/cam}) \end{aligned} \quad (6)$$

In Eq. 6, $\mathbf{R}^{R/B}$ is the rotation matrix which rotates vectors from the body frame B into the body frame R, and $\mathbf{R}^{C/R}$ is the rotation matrix which rotates vectors from the reference frame R into the camera frame C (if these frames are not the same).

4 Algorithm

To solve the estimation problem presented in the previous section, a particle filter is implemented. The particle filter consists of many particles (referred to by index i), each containing a hypothesis for the kinematic state at the latest time step k, $\overline{s}^{[i]}_k$.

For a given particle, each feature's position can be estimated independently of the others [1, 21]. This algorithm duplicates the approach of FastSLAM, which has each particle maintain a set of small EKFs, one for each feature. Within each particle i, the distribution $p(X_j|\overline{s}^{[i]})$ is assumed to be Gaussian, $\mathcal{N}(\overline{\mu}_j^{[i]}, \Sigma_j^{[i]})$.

Each particle contains N EKFs; that is, for particle i, every feature j has a mean $\overline{\mu}_j^{[i]}$ and covariance $\Sigma_j^{[i]}$ to estimate its true 3-D location in the body frame (X_j).

At every time-step, the particle filter proceeds through three distinct updates: a measurement update, a weighting/resampling of the particles based on their likelihoods, and a motion update to predict the kinematic state at the next time step. These are described in turn.

4.1 Measurement Updates

The algorithm depends on consistently tracking features over several images. Feature correspondence is handled 'upstream' of the filter. Any type of point features can be used. SIFT features [4] were used for the experiments in this paper because they have a few beneficial properties. Each SIFT feature has a unique 128-vector descriptor which greatly aids correspondence. Also, SIFTs are somewhat robust to orientation (including out-of-plane) changes, making them useful for tracking on a rotating body.

During the SPEAR process, a library of previously observed SIFT features is maintained. When a new image is taken, each newly observed SIFT is compared against this library to find a potential match. Three steps are performed to find the correspondence: a pixel window is applied to narrow the search, then the standard SIFT correspondence algorithm is applied [4], and finally a threshold is applied on the dot product between the new SIFT and any potential match. See [3] for more information on the feature correspondence procedure.

To improve the number of SIFT features detected in the underwater environment, Contrast Limited Adaptive Histogram Equalization [22] (CLAHE) is applied to the raw image, prior to feature detection taking place.

The pixel locations $\overline{z}_j$ of the SIFT features in the image plane are passed to the algorithm as measurements. In the case of monocular vision, a feature's 3-D location estimate cannot be initialized until the feature has been observed more than once; this is necessary in order to establish range from the bearings-only measurements. The feature initialization [3] is done via least-squares estimation.

To update the features' 3-D location estimates, the measurement model (Eq. 5) is used to calculate the predicted measurement and its Jacobians:

$$\hat{\overline{z}}_j^{[i]} = g(\overline{s}^{[i]}, \overline{\mu}_j^{[i]}) \tag{7}$$

$$G_{X_j} = \frac{\partial g}{\partial X_j}\Big|_{\left(\overline{s}^{[i]}, \overline{\mu}_j^{[i]}\right)} \tag{8}$$

$$G_s = \frac{\partial g}{\partial \overline{s}}\Big|_{\left(\overline{s}^{[i]}, \overline{\mu}_j^{[i]}\right)} \tag{9}$$

This is followed by a standard EKF measurement update, performed on the feature's mean and covariance (Eqns. 10-13). Note that r is the noise standard deviation. This parameter is a measure of the importance with which the latest measurement is treated.

$$Q = \left(G_s P {G_s}^T + G_{X_j} \Sigma_j^{[i]} {G_{X_j}}^T + r^2 I\right) \tag{10}$$

$$K = \Sigma_j^{[i]} {G_{X_j}}^T Q^{-1} \tag{11}$$

$$\overline{\mu}_j^{[i]} = \overline{\mu}_j^{[i]} + K(\overline{z}_j - \hat{\overline{z}}_j^{[i]}) \tag{12}$$

$$\Sigma_j^{[i]} = \left(I - K G_{X_j}\right) \Sigma_j^{[i]} \tag{13}$$

4.2 Particle Weighting and Resampling

Following the measurement update step, the particle filter updates an importance weighting for each of the particles. This is a measure of the likelihood of the particle, given the measurements observed up to the given time step. The importance weighting of particle i is denoted $w^{[i]}$. There are many choices available for weighting measures. For example, inlier/outlier counts based on pixel error thresholding have been used in previous SFM particle filters [14, 16].

Given the choice made above to model $p(X_j|\overline{s}^{[i]})$ as 'Gaussian-like' ($\mathcal{N}(\overline{\mu}_j^{[i]}, \Sigma_j^{[i]})$), a natural choice of weighting is the Gaussian likelihood (as is done in the original FastSLAM algorithm [1,17]). The measurement of feature j, $\overline{z}_j$, has likelihood denoted $w_j^{[i]}$, and calculated by:

$$w_j^{[i]} = |2\pi Q|^{-\frac{1}{2}} exp\left\{-\frac{1}{2}(\overline{z}_j - \hat{\overline{z}}_j^{[i]})^T Q^{-1} (\overline{z}_j - \hat{\overline{z}}_j^{[i]})\right\} \tag{14}$$

The overall likelihood of a specific particle, $w^{[i]}$, is the cumulative likelihoods of that particle's measurements:

$$w^{[i]} = \prod_j w_j^{[i]} \tag{15}$$

Periodically, the particles are resampled, in order to remove the least likely hypotheses, and focus on the most likely. This algorithm resamples based on a calculation of effective sample size N_{eff} [18]:

$$N_{eff} = \frac{1}{\sum_i (w^{[i]})^2} \tag{16}$$

When N_{eff} is less than half the total number of particles, resampling takes place. In order to resample efficiently, low-variance resampling is used [17]. After resampling, all particles are set to equal weight.

4.3 Motion Updates

For each particle i, the proposed kinematic state $\overline{s}^{[i]}_{(k+1)}$ for the next time-step is drawn from the following proposal distribution:

$$\overline{s}^{[i]}_{(k+1)} \sim \begin{bmatrix} \mathcal{F}(\mu^{[i]}_{prop_{[7:9]}}, (\Sigma^{[i]}_{prop_{[1,1]}})^{-1}) \\ \mathcal{N}(\mu^{[i]}_{prop_{[4:12]}}, \Sigma^{[i]}_{prop_{[4:12,4:12]}}) \end{bmatrix} \tag{17}$$

This proposal distribution is a Fisher distribution over the Euler angles, and a Gaussian distribution over the other components of the kinematic state vector. It takes into account a motion model prediction ($f(\overline{s}^{[i]}_k)$), *as well as* the latest measurements taken ($\overline{z}_{j(k+1)}$) [2]. The covariance ($\Sigma^{[i]}_{prop}$) and mean ($\mu^{[i]}_{prop}$) of the proposal distribution are:

$$\Sigma^{[i]}_{prop} = \left[\sum_j \tilde{G}_s^{[i]^T} \left(Q_j^{[i]} \right)^{-1} \tilde{G}_s^{[i]} + P^{-1} \right]^{-1} \tag{18}$$

$$\mu^{[i]}_{prop} = \Sigma^{[i]}_{prop} \sum_j \tilde{G}_s^{[i]^T} \left(Q_j^{[i]} \right)^{-1} \left(\overline{z}_j - \tilde{\overline{z}}_j^{[i]} \right) + f(\overline{s}^{[i]}_k) \tag{19}$$

where:

$$\tilde{\overline{z}}_j^{[i]} = g(f(\overline{s}^{[i]}_k), \overline{\mu}_j^{[i]}) \tag{20}$$

$$\tilde{G}_{X_j} = \left. \frac{\partial g}{\partial X_j} \right|_{\left(f(\overline{s}^{[i]}_k), \overline{\mu}_j^{[i]} \right)} \tag{21}$$

$$\tilde{G}_s = \left. \frac{\partial g}{\partial \overline{s}} \right|_{\left(f(\overline{s}^{[i]}_k), \overline{\mu}_j^{[i]} \right)} \tag{22}$$

$$Q_j^{[i]} = R_{prop} + \tilde{G}^{[i]}_{X_j} \Sigma_j^{[i]} \tilde{G}^{[i]^T}_{X_j} \tag{23}$$

Once the kinematic states have been proposed, the filter then advances to the next time-step and begins the measurement update again.

5 Experiments

Field testing was conducted in conjunction with the Monterey Bay Aquarium Research Institute (MBARI). Using the MBARI research vessel *Point Lobos* and the ROV *Ventana* (shown in Figures 2 and 3), underwater tethered targets were imaged.

Figure 2: MV *Point Lobos*

Figure 3: ROV *Ventana*

An example of an underwater target of interest is shown in Figure 4. No *a priori* information exists, and autonomous proximity operations are desired. The purpose of the field experiments is to track its pose in the water while simultaneously reconstructing the shape of the target (a precursor to positioning control algorithms).

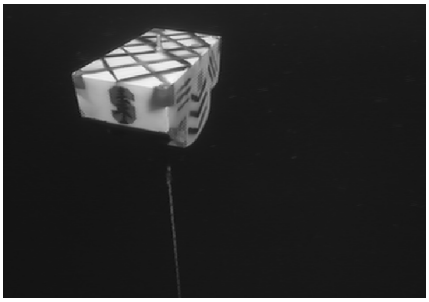

Figure 4: Example of an Underwater Target

A set of images was obtained with the main science camera on *Ventana*, at 10 Hz rate. Samples of the sequence of 150 images, at 1 second intervals, are shown in Figure 5.

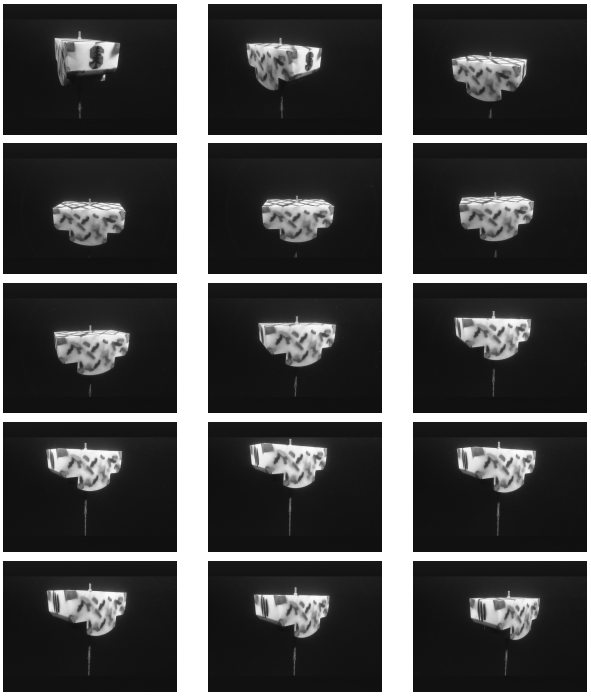

Figure 5: Sample Views from Image Sequence

Figures 6, 7, 8, 9, and 10 present the results of applying particle filter SPEAR on the set of images above. The reconstructed target is shown in Figure 6; compared with the views of the target shown in Figure 5, it shows accurate representation of the 3-D shape.

The relative pose estimate time history is shown in Figures 7-10. Figure 7 shows the Euler angles, Figure 8 shows the angular rates, Figure 9 shows the position, and Figure 10 shows the velocity.

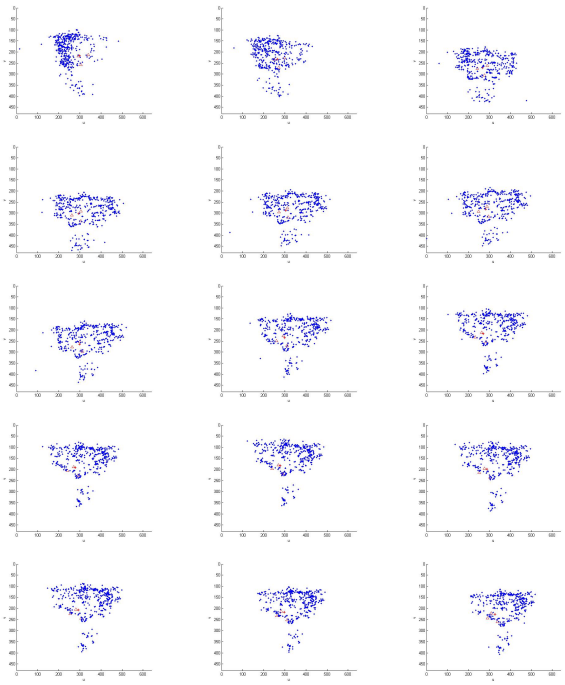

Figure 6: Reconstruction of the Target

6 Summary

A technique has been presented to estimate the pose and 3-D shape simultaneously of an underwater tethered object. The algorithm described here is based on an efficient particle filter algorithm from the SLAM community (FastSLAM). In particular, the algorithm was applied to a problem with bearings-only measurements and no *a priori* information about the target.

The results show the successful estimation of an underwater target's shape and pose, validating the particle filter approach of FastSLAM as an effective tool for the SPEAR problem.

Acknowledgements

The authors would like to thank the Monterey Bay Aquarium Research Institute for support, and the Stanford Graduate Fellowship and NASA grant #Z627401-A for partial funding. Additionally, the authors would like to thank the MBARI pilots (Knute Brekke, Mike Burczynski, Craig Dawe, and D.J. Osborne) and crew of the *Point Lobos* for their assistance in conducting ROV operations in support of the research goals of this paper.

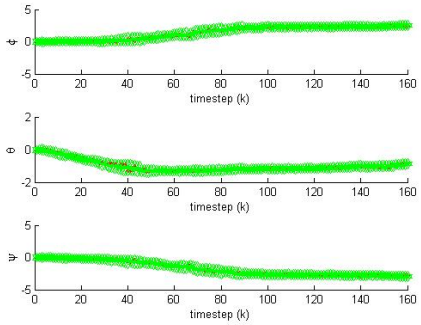

Figure 7: Orientation of the Mock Target

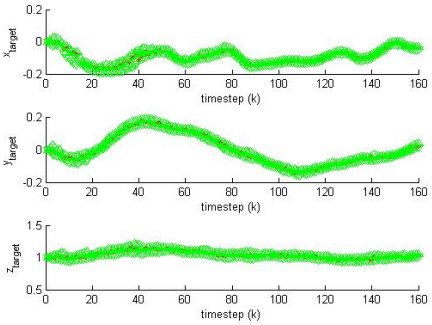

Figure 9: Position of the Mock Target

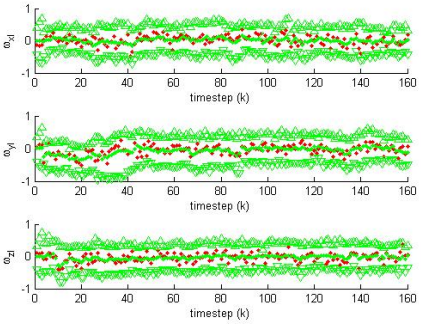

Figure 8: Angular Rate of the Mock Target

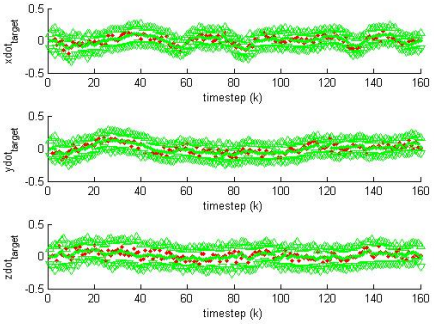

Figure 10: Velocity of the Mock Target

References

[1] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam: a factored solution to the simultaneous localization and mapping problem," in *Eighteenth National Conference on Artificial Intelligence*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 2002, pp. 593–598.

[2] ——, "Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *International Joint Conference on Artificial Intelligence*, vol. 18, 2003, pp. 1151–1156.

[3] S. Augenstein and S. M. Rock, "Simultaneous estimation of target pose and 3-d shape using the fastslam algorithm," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*. Chicago, IL: AIAA, August 2009.

[4] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] J. Salvi, Y. Petillot, S. Thomas, and J. Aulinas, "Visual SLAM for Underwater Vehicles using Video Velocity Log and Natural Landmarks," in *Proceedings OCEANS 2008*, 2008.

[6] R. Eustice, O. Pizarro, and H. Singh, "Visually Augmented Navigation for Autonomous Underwater Vehicles," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 2, pp. 103–122, 2008.

[7] P.-M. Lee, B.-H. Jeon, and S.-M. Kim, "Visual servoing for underwater docking of an autonomous underwater vehicle with one camera," *OCEANS 2003. Proceedings*, vol. 2, pp. 677–682 Vol.2, Sept. 2003.

[8] A. Plotnik and S. Rock, "Visual Servoing of an ROV for Servicing of Tethered Ocean Moorings," in *Proceedings of the MTS/IEEE OCEANS Conference, Boston, MA*, 2006.

[9] P. Jasiobedzki, S. Se, M. Bondy, R. Jakola, and S. MDA, "Underwater 3D Mapping and Pose Estimation for ROV Operations," in *Proceedings OCEANS 2008*, 2008.

[10] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer, 2003.

[11] T. Broida, S. Chandrashekhar, and R. Chellappa, "Recursive 3-d motion estimation from a monocular image sequence," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 4, pp. 639–656, Jul 1990.

[12] A. Azarbayejani and A. Pentland, "Recursive estimation of motion, structure, and focal length," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 562–575, Jun 1995.

[13] A. Chiuso, P. Favaro, H. Jin, and S. Soatto, "Structure from motion causally integrated over time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 523–535, 2002.

[14] M. Pupilli and A. Calway, "Real-time camera tracking using a particle filter," in *In Proc. British Machine Vision Conference*, 2005, pp. 519–528.

[15] D. Marimon, Y. Maret, Y. Abdeljaoued, and T. Ebrahimi, "Particle filter-based camera tracker fusing marker and feature point cues," in *Proc. SPIE*, vol. 6508, no. 1, 2007. [Online]. Available: http://link.aip.org/link/?PSI/6508/65080O/1

[16] P. Chang and M. Hebert, "Robust tracking and structure from motion through sampling based uncertainty representation," in *Proceedings of ICRA '02*, May 2002.

[17] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. MIT press, Cambridge, Massachusetts, USA, 2005.

[18] A. Doucet, N. De Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[19] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, Jun 2001.

[20] N. Fisher, T. Lewis, and B. Embleton, *Statistical analysis of spherical data*. Cambridge Univ Pr, 1993.

[21] K. P. Murphy, "Bayesian map learning in dynamic environments," in *In Neural Info. Proc. Systems (NIPS)*, 1999, pp. 1015–1021.

[22] K. Zuiderveld, *Graphics Gems IV*. Cambridge, MA: Academic Press, 1994, ch. VIII.5, "Contrast Limited Adaptive Histogram Equalization", pp. 474–485.