Improved Frame-to-Frame Pose Tracking during Vision-Only
SLAM/SFM with a Tumbling Target

Sean Augenstein and Stephen M. Rock

Abstract—A hybrid algorithm for real-time frame-to-frame
pose estimation during monocular vision-only SLAM/SFM is
presented. The algorithm combines concepts from two existing
approaches to pose tracking, Bayesian estimation methods and
measurement inversion techniques, to achieve in real-time a
feasible, smooth estimate of the relative pose between a robotic
platform and a tumbling target. It is assumed that no a priori
information about the target is available, and that only
a monocular camera is available for measuring the relative
motion of the target with respect to the robotic platform. The
rationale for a hybrid approach is explained, and an algorithm
is presented. A specific implementation using a modified Rao-
Blackwellised particle filter is described and tested. Results
from both numerical simulations and field experiments are
included which demonstrate the performance and viability of
the hybrid approach. The hybrid approach to pose estimation
described here is applicable regardless of the method by which
the map/reconstruction is estimated.

I. INTRODUCTION

This paper presents a method for frame-to-frame tracking
of the pose of an unknown moving object, while concurrently
reconstructing its shape, as it is observed by a robotic
platform. The observing platform can be equipped with own-
pose instrumentation, but it is assumed that only a monocular
camera is available for sensing the relative motion of the
target with respect to the platform. This problem is an
instance of “vision-only SLAM” (Simultaneous Localization
and Mapping) or “recursive SFM” (Structure from Motion),
and will be referred to here as “vision-only SLAM/SFM”.

The motivation for developing this method is the need
to determine in real-time a feasible, smooth estimate of the
relative pose between an observing robot and a tumbling
target, using no a priori knowledge about the target. Such
an estimate can be used for real-time control of the robotic
platform with respect to the target, enabling autonomous
rendezvousing and docking with objects for which a priori
shape, mass properties, and appearance information might
not be available. Applications include inspection and re-
pair of damaged space satellites (see Figure 1) or tethered
underwater science instruments. To perform such tasks, an
accurate 6DOF relative pose estimate is necessary as a
reference for closed loop control, and an accurate 3D shape
estimate is necessary for hazard avoidance and planning
robotic interaction.

Significant work has been done in the vision-only
SLAM/SFM field on frame-to-frame pose tracking methods
for conceptually related applications, but no existing tech-
nique is able to provide a smooth, feasible, real-time estimate
for this moving target problem. Existing methods can be
divided into two broad classes: Bayesian filtering (examples
of which include [1]–[7]), and measurement inversion (exam-
ples of which include [8]–[11]). Both address frame-to-frame
pose tracking, that is, the estimation at camera frame-rate of
the relative pose (and possibly its derivatives) between
the target and the camera.

Specifically, the pose $\mathbf{s}$ consists of relative orientation
Euler angles $\mathbf{\hat{\theta}}$, angular rate $\mathbf{\hat{\omega}}$, position $\mathbf{\hat{x}_p}$, and velocity $\mathbf{\hat{v}_p}$
(where subscript $p$ denotes the point on the target serving as
reference):

$$\tilde{s}_t = \begin{bmatrix} \tilde{\theta} \\ \tilde{\omega} \\ \tilde{x}_p \\ \tilde{v}_p \end{bmatrix}_t$$

(1)

Fig. 1. Tumbling Satellite being Imaged by Chase Vehicle

Note that these variables describe the moving target’s pose
(to be estimated), with the camera’s pose known via own-
pose sensors. This is in contrast to a traditional SLAM
formulation, where the pose vector describes the moving
camera’s pose (to be estimated), with the target’s pose known
due to it being a static, unmoving environment.

An (imperfect, incomplete) map of the target is available
for reference as it is constructed simultaneously. The map
is expressed as the positions of $N$ points, relative to the
reference point $p$ on the target, in the target’s frame ($F$):
The pose relates 3D locations between the target’s reference frame $(F)$ and the camera’s reference frame $(C)$:

$$ \bar{x}^C_j = R(\bar{\theta})^{C/F} \bar{x}^F_{j/p} + \bar{x}_p $$  \hspace{1cm} (3)

where the rotation matrix $R(\bar{\theta})^{C/F}$ is a function of the relative orientation $\bar{\theta}$.

### A. Bayesian Filtering

A Bayesian filter is a recursive optimization at each time-step based on the constraints of both a camera measurement model and a process model describing how the relative pose changes from one time-step to the next.

In a standard Bayesian approach to SLAM/SFM the process model $f()$ is used to propagate the pose from one time-step to the next:

$$ \tilde{s}_t = f(\tilde{s}_{t-1}) $$  \hspace{1cm} (4)

and then a correction is performed incorporating the most recent measurement $Z_t$. This happens via comparison against the predicted measurement $\hat{Z}_t$, as calculated based on a model $g()$ of how the measurement is formed:

$$ Z_t = g(\tilde{s}_t, X^F) $$  \hspace{1cm} (5)

Additionally, the map estimate $X^F$ is updated in this last step. The process then repeats.

### B. Measurement Inversion

‘Measurement inversion’ refers to the use of projective geometry to invert the camera’s measurements and calculate relative pose at each time-step. Measurement inversion optimizes based on the constraints of the camera measurement model only, determining the relative pose estimate which gives the best-fit of the map to the measurements.

A typical measurement inversion approach [11] works as follows. Let the function $h()$ represent the inversion of Equation 5 given $X^F$:

$$ \tilde{s}_t = h(Z_t, X^F) = g^{-1}(X^F(Z_t)) $$  \hspace{1cm} (6)

Typically, no process model is used to predict the relative change in pose from one time-step to the next. Instead, at each time-step, a candidate pose is selected via RANSAC [12] (or a similar consensus) algorithm. Random minimal sets of map features are selected (3 features [13], if the map is already initialized), and pose hypotheses $\tilde{s}_t$ are calculated from Equation 6. Each hypothesis is then used with Equation 5 to predict the features’ measurements $\hat{Z}_t$, and these are compared against the actual measurements $Z_t$ to score the hypothesis.

The strongest pose hypothesis is then refined, based on a non-linear minimization of reprojection error (i.e., iterative refinement), to determine a pose estimate $\hat{s}_t$.

### C. Pose Tracking for a Tumbling Target

While measurement inversion can be performed in real-time, it generally does not use a process model to hypothesize pose based on previous filtered pose information. Consequently there is no mechanism to ensure that the target’s motion propagates in a physically realistic manner (i.e., the pose is free of jumps).

In contrast, a Bayesian approach to pose tracking (and in general any approach using a process model) ensures that the target’s motion is physically reasonable. By incorporating a process model, the filter has ‘memory’ of target motion. This effectively serves to prevent jumps by disambiguating between multiple distinct poses that are high-likelihood fits to the measurements, all but one of which are physically erroneous (as is possible when measurements are bearing-only [1]). Process models typically include noise to account for uncertainty in the prediction of motion.

A drawback to such approaches (e.g., non-linear Bayesian filters) is that they operate poorly when noise covariances are large, due to approximations made for handling non-linear models. For example, in an extended Kalman filter formulation, large noise covariance causes slow convergence and increased linearization error. In a particle filter formulation, large noise covariance requires increased numbers of particles, causing non-real-time performance.

The issue is that, for the application described here (i.e., where the map is undergoing unknown motion, and no mass properties are known), the equations of motion are such that a process model does require a large noise covariance (explained in Section II), hence implementing a Bayesian estimator can be problematic.

A hybrid solution to overcome this difficulty is presented in this paper. Section II explains why the equations of motion for a moving target hinder the use of a standard Bayesian filter for real-time pose tracking during vision-only SLAM/SFM. Section III explains an alternative, hybrid approach to frame-to-frame pose tracking that splits the estimation into a partial Bayesian filter (for the rotational pose states) and a partial measurement inversion (for the translational pose states). Section IV explains the details of the translation-only measurement inversion. Section V gives a specific implementation adapted from Rao-Blackwellised particle filters [14], [15]. Section VI gives experimental results, validating improvements in accuracy and computational cost, as well as the real-time capability of this algorithm. Section VII concludes.

## II. Equations of Motion of Tumbling Target

The process model Equation 4 for the rigid body problem considered here can be written as:

\[
X^F = \begin{bmatrix} \bar{x}_{1/p} & \bar{x}_{2/p} & \cdots & \bar{x}_{j/p} & \cdots & \bar{x}_{N/p} \end{bmatrix}^F
\]  \hspace{1cm} (2)
\[
\begin{align*}
\tilde{\theta}_t &= \tilde{\theta}_{t-1} + \Delta t \cdot M(\tilde{\theta}_{t-1})\hat{\omega}_{t-1} \\
\hat{\omega}_t &= \hat{\omega}_{t-1} + \Delta t \cdot J^{-1}T_{total} \\
\bar{x}_{p_{1}} &= \bar{x}_{p_{1-1}} + \Delta t \cdot \bar{v}_{p_{1-1}} \\
\bar{v}_{p_{1}} &= \bar{v}_{p_{1-1}} + \Delta t \left( \frac{1}{m} \bar{F}_{total} + J^{-1}T_{total} \times \bar{r}_p + \hat{\omega}_{t-1} \times \hat{\omega}_{t-1} \times \bar{r}_p \right)
\end{align*}
\] (7)

In Equation 7, a matrix function \( M(\tilde{\theta}) \) of the Euler angles maps angular rates \( \hat{\omega} \) to Euler angle derivatives \( \tilde{\omega} \) [16]. Also, \( J \) is the inertia matrix, \( m \) is the mass, \( \bar{r}_p \) is the positional offset of the reference point from the center of mass, \( T_{total} \) is the total external torque, and \( \bar{F}_{total} \) is the total external force acting through the center of mass.

If the external forcing terms \( T_{total} \) and \( \bar{F}_{total} \) are not known (as in the case considered here), then Equation 7 can be written as a set of kinematic relationships driven by noise (\( \nu_\omega \) and \( \nu_v \)):

\[
\begin{align*}
\tilde{\theta}_t &= \tilde{\theta}_{t-1} + \Delta t \cdot M(\tilde{\theta}_{t-1})\hat{\omega}_{t-1} \\
\hat{\omega}_t &= \hat{\omega}_{t-1} + \nu_\omega \\
\bar{x}_{p_{1}} &= \bar{x}_{p_{1-1}} + \Delta t \cdot \bar{v}_{p_{1-1}} \\
\bar{v}_{p_{1}} &= \bar{v}_{p_{1-1}} + \nu_v
\end{align*}
\] (8)

In Equations 8 and 9, \( \nu_\omega \) and \( \nu_v \) are zero-mean, Gaussian random forcing terms (drawn from \( N(0, \Sigma_\omega) \) and \( N(0, \Sigma_v) \), respectively). Covariances \( \Sigma_\omega \) and \( \Sigma_v \) are chosen to be large enough to encompass the full range of feasible accelerations.

The issue with a target about which no \textit{a priori} information is available is that the term \( \bar{r}_p \), which locates the center of mass of the target, is not known. In this case, Equation 9 must be modelled as:

\[
\begin{align*}
\tilde{\theta}_t &= \tilde{\theta}_{t-1} + \Delta t \cdot M(\tilde{\theta}_{t-1})\hat{\omega}_{t-1} \\
\hat{\omega}_t &= \hat{\omega}_{t-1} + \nu_\omega \\
\bar{x}_{p_{1}} &= \bar{x}_{p_{1-1}} + \Delta t \cdot \bar{v}_{p_{1-1}} \\
\bar{v}_{p_{1}} &= \bar{v}_{p_{1-1}} + \nu_v
\end{align*}
\] (9)

In Equation 10, The covariance \( \Sigma_v, Total \) associated to \( \nu_v, Total \) must be chosen large enough to encompass the full range of feasible accelerations as well as the deterministic effects of the term involving \( \bar{r}_p \).

The effect of this is to make a Bayesian approach inaccurate and/or intractable for real-time implementation. Even though the covariance associated with the rotational equations of motion (\( \Sigma_\omega \)) can be kept small, the covariance associated with the translational equations of motion (\( \Sigma_v, Total \)) must be made large. As discussed in Section I-C, this has various detrimental effects, due to the approximations made for non-linear Bayesian filters.

Note that, in the case of traditional SLAM (moving camera, static target), this issue is typically not a problem since \( \bar{r}_p \) is known: it is the physical offset on the robotic platform between the camera and the center of mass, which can be measured.

III. HYBRID ESTIMATION

A hybrid solution to overcome this difficulty is presented in this paper. Rotation is predicted in a Bayesian filter using the available process model. Translation is predicted via measurement inversion, to avoid using the part of the process model with large covariance noise. Using the rotational process model preserves a feasible smooth trajectory for relative orientation. Further, as is explained below (and shown in Section IV), given relative orientation, a translation-only measurement inversion is not susceptible to jumps and non-smoothness, and is a fast calculation. Therefore the hybrid solution yields a real-time, smooth, and physically realistic method for tracking pose.

Specifically, the pose at time-step \( t \) is now defined as:

\[
\bar{s} = \begin{bmatrix}
\tilde{\theta} \\
\hat{\omega} \\
\bar{x}_p
\end{bmatrix}
\] (11)

For a given relative position \( \bar{x}_{p_{1}} \), map \( X^F \), and set of bearings-only measurements \( Z_t \) (i.e., a monocular image) at a time-step, there can exist more than one relative orientation \( \tilde{\theta}_t \) of the target which is a close fit when aligning the map and measurements. That is, the probability distribution of orientation (given position, map, and measurements) based on the measurement model is multimodal (See [1]). An estimate based only on the measurement model would be susceptible to jumping between different peaks of the probability distribution from one time-step to the next.

By using a process model, the filter is ‘guided’ towards the correct peak in probability, i.e., the one that agrees with the system’s underlying dynamics. Equation 8 is the process model used for predicting the rotational states, \( \tilde{\theta}_t \) and \( \hat{\omega}_t \).

In contrast, for a given \( \tilde{\theta}_t \), \( X^F \), and \( Z_t \), it is possible to rearrange the measurement equations so that there is a single close fit when aligning the map and measurements, i.e., the estimate of \( \bar{x}_{p_{1}} \) is the maximum likelihood of a unimodal uncertainty distribution. This is shown in Section IV, where measurement inversion for relative position is performed by convex optimization, specifically, a fast linear least squares calculation.

To predict position \( \bar{x}_{p_{1}} \), an equation similar to Equation 6 is formed (explained in detail in Section IV), but with the relative orientation at the current time-step also assumed to be given:

\[
\bar{x}_{p_{1}} = h_{\bar{x}_p} (Z_t, X^F, \tilde{\theta}_t) = g^{-1}_{\|X^F, \tilde{\theta}_t\|} (Z_t)
\] (12)

Given the orientation predicted by Equation 8, the map estimate \( X^F \), and the latest measurement \( Z_t \), the relative position is predicted by Equation 12.

Finally, in the same way as Bayesian Estimation, the overall pose estimate \( \hat{s} \) is corrected via comparison of the predicted latest measurement \( \hat{Z}_t \) and the actual latest measurement \( Z_t \) (the map estimate \( X^F \) can also be corrected). The process then repeats.

Note that this paper focuses solely on the frame-to-frame pose tracking aspect of vision-only SLAM/SFM. Specifically, two aspects of the field are not covered. The first is loop-closure. Much recent work in SLAM/SFM has been focused on addressing large-scale mapping/loop-closure [17, 18]; no improvements are presented in that area, as the
pose tracking algorithm in this paper is independent of and intended to be used in conjunction with these techniques.

The second topic is the use of bundle adjustment versus a purely causal approach. The decision is independent of whether or not a process model is used during frame-to-frame tracking; as [11] succinctly states, “... bundle adjustment as well as [causal] filtering for the application of camera tracking can be used both with or without a camera motion [process] model.” The choice of using a process model or not is dictated by the accuracy of the information it might provide. The choice of using bundle adjustment is discussed thoroughly in [19].

IV. TRANSLATION VIA MEASUREMENT INVERSION

Translation is predicted via measurement inversion, assuming a map has been previously initialized and the orientation for the upcoming time-step has already been predicted based on Equation 8.

A pinhole projection model of the camera is assumed, so a single feature’s measurement is expressed as:

\[
\begin{bmatrix}
    u_j \\
    v_j
\end{bmatrix} = f_c \begin{bmatrix}
    x^C_j \\
    y^C_j
\end{bmatrix} = \frac{f_c}{z_j} \begin{bmatrix}
    x^C_j \\
    y^C_j
\end{bmatrix} \tag{13}
\]

In Equation 13, \( f_c \) is the focal length (an intrinsic parameter of the camera), and \( x^C_j, y^C_j \), and \( z_j \) are the components of feature \( j \)'s position in the camera frame. The measurements taken \((u_j, v_j)\) are pixel coordinates on the image plane. Rearranging Equation 13 gives:

\[
\begin{bmatrix}
    0 \\
    0
\end{bmatrix} = \begin{bmatrix}
    u_j z_j^C - f_c x^C_j \\
    w_j z_j^C - f_c y^C_j
\end{bmatrix} \tag{14}
\]

Equation 14 can be stacked to form one large linear expression involving all the measurements at a given time, \( Z_t \). An initial estimate of the feature locations in the camera frame, denoted \( X^C_0 \), is to be corrected by translational offsets \( \Delta x, \Delta y, \) and \( \Delta z \), while satisfying as closely as possible the constraints of Equation 15.

\[
\begin{bmatrix}
    0 \\
    0 \\
    0 \\
    \vdots \\
    0 \\
    0
\end{bmatrix} \begin{bmatrix}
    u_1 (z^C_{1o} + \Delta z) - f_c (x^C_{1o} + \Delta x) \\
    v_1 (z^C_{1o} + \Delta z) - f_c (y^C_{1o} + \Delta y) \\
    \vdots \\
    u_j (z^C_{jo} + \Delta z) - f_c (x^C_{jo} + \Delta x) \\
    v_j (z^C_{jo} + \Delta z) - f_c (y^C_{jo} + \Delta y) \\
    \vdots \\
    u_N (z^C_{no} + \Delta z) - f_c (x^C_{no} + \Delta x) \\
    v_N (z^C_{no} + \Delta z) - f_c (y^C_{no} + \Delta y)
\end{bmatrix} = 0 \tag{15}
\]

Note that because the features are part of a single rigid body, the translational offset is common. Also note that, based on Equation 3, \( X^C_0 \) is a function of the orientation estimate at the latest time-step (as predicted by Equation 8), \( \hat{\delta}_t \), and an initial guess at target position \( \hat{\bar{x}}_{p_o} \).

Rewriting Equation 15 as a linear equation in the translational offset vector:

\[
\begin{bmatrix}
    f_c \hat{x}^C_{x} - u_1 \hat{z}^C_{1o} \\
    f_c \hat{x}^C_{y} - u_1 \hat{z}^C_{1o} \\
    \vdots \\
    f_c \hat{x}^C_{x} - u_j \hat{z}^C_{jo} \\
    f_c \hat{x}^C_{y} - u_j \hat{z}^C_{jo} \\
    \vdots \\
    f_c \hat{x}^C_{x} - u_N \hat{z}^C_{no} \\
    f_c \hat{x}^C_{y} - u_N \hat{z}^C_{no}
\end{bmatrix} = \begin{bmatrix}
    -f_c & 0 & u_1 \\
    0 & -f_c & v_1 \\
    \vdots & \vdots & \vdots \\
    -f_c & 0 & u_j \\
    0 & -f_c & v_j \\
    \vdots & \vdots & \vdots \\
    -f_c & 0 & u_N \\
    0 & -f_c & v_N
\end{bmatrix} \begin{bmatrix}
    \Delta x \\
    \Delta y \\
    \Delta z
\end{bmatrix} \tag{16}
\]

\[
\hat{\bar{x}}_p = \hat{\bar{x}}_{p_o} + \Delta \bar{x}^C_{WLS} \tag{19}
\]

Because this is a linear optimization problem, the initial guess at the vehicle position this time-step \( \hat{\bar{x}}_{p_o} \) is \textit{unimportant}; the optimization process will yield a value for \( \Delta \bar{x}^C_{WLS} \) which always adjusts to the same optimal vehicle position \( \bar{x}_p \).

To summarize, the position estimate of the rigid body reference point \( \bar{x}_p \) is based on a best fit alignment of the feature positions’ image plane projections to the latest measurements. This inversion of the pinhole camera model is represented in functional form by Equation 12.

V. IMPLEMENTATION WITH RAO-BLACKWELLISED PARTICLE FILTERS

Sections III and IV described the general approach to hybrid pose tracking. A specific vision-only SLAM/SFM implementation is now given, modified from a Rao-Blackwellised particle filter [14], [15]. Rao-Blackwellised particle filters have advantages to alternative Bayesian forms such as the extended Kalman filter, in that they allow for multi-modal probability distributions and can maintain a much larger map of features. They have been previously
applied in the vision-only SLAM/SFM problem [1], [3], although with pose tracking done via standard Bayesian estimation. Accurate operation was achieved at frame-rate (30Hz) with 50 particles [3].

The pose prediction step will be run for each particle $i$ in the filter (containing $K$ particles in total). Each particle contains its own estimate of the feature locations $\hat{X}[i]$ and its own estimate of the robot’s pose $\hat{s}[i]$. First the rotational states are predicted, using Equation 8 (without noise):

$$\begin{align*}
\hat{\theta}[i]_{t} &= \hat{\theta}[i]_{t-1} + \Delta t \cdot M(\hat{\theta}[i]_{t-1})\hat{\omega}[i]_{t-1} \\
\hat{\omega}[i]_{t} &= \hat{\omega}[i]_{t-1}
\end{align*}$$

Using the latest measurements available, $Z_t$ (which is common to all particles), each particle uses its predicted orientation $\hat{\theta}[i]_{t}$ and estimate of feature locations $\hat{X}[i]$ to perform the calculations described in Section IV, yielding $\hat{x}_{pi}^{[i]}$:

$$\hat{x}_{pi}^{[i]} = h_{\hat{x}} \left( Z_t, \hat{X}[i], \hat{\theta}[i]_{t} \right)$$

The state vector is now perturbed with process noise, to generate particle diversity. Following the approach of the FastSLAM 2.0 algorithm [20], this process noise takes into account the latest measurement $Z_t$. FastSLAM 2.0 helps focus the proposal particle cloud towards the highest likelihood regions of the state space, and is used for the experiments in Section VI. The pose sampling step in FastSLAM 2.0 has $O(K\xi_{obs})$ computational cost.

Note that estimating the translational states via measurement inversion (Equation 21) for all the particles has $O(K\xi_{obs})$ total cost, so this new hybrid approach does not increase the computational complexity of FastSLAM 2.0 above the traditional, fully Bayesian approach.

After the pose prediction step, the Rao-Blackwellised particle filter proceeds to estimate the 3D locations of the features (the ‘reconstruction’ or ‘map’) and weight the particles, and then to the resampling step, where likely particles are maintained and unlikely particles are discarded.

Map initialization is an important topic in monocular vision-only SLAM, as map features must be observed more than once before they can be initialized, due to the lack of a depth measurement from the sensor. For the algorithm explained in this paper, map initialization is performed in the same way as [1].

VI. Experiments

To demonstrate the performance of this algorithm, two sets of experiments were conducted. The first used a synthetic target, with known locations of body features as well as known relative pose, for comparison against truth (VI-A). The second used a tethered underwater target, and demonstrated the algorithm’s performance in the field (VI-B).

For the field experiments reported in Section VI-B, SIFT image features [21] were used, extracted from the image stream via a graphical processing unit (GPU) [22] so as to run in real-time (30Hz). SIFTs have the advantage of being robust and recognizable, aiding matching and avoiding problems of mis-correspondence. Note that any other type of image feature could be used alternatively, provided it meets the demands for consistent correspondence.

A. Synthetic Tests

Testing the algorithm against synthetic targets in software allows for validation against truth data, both in the target map $X^F$ and the relative pose $s$. A target was formed of 200 features, randomly located on four sides of a cube. Figure 2 shows a plot of the target and the relative motion of the camera in target-body centered coordinates.

Figures 3 and 4 show the true (black) and estimated (green) pose vs. time-step, for hybrid estimation with 50 particles, for 10 trials. Figure 5 shows the RMS error of the estimated map vs. time-step for the same trials. The orientation (Fig. 4) is expressed in radians, and the position (Fig. 3) and map RMS error (Fig. 5) are expressed in dimensionless distance units.

Figures 6 and 7 show the true (black) and estimated (green) pose vs. time-step, for full Bayes estimation with 500 particles, for 10 trials. Figure 8 shows the RMS error of the estimated map vs. time-step for the same trials. The orientation (Fig. 7) is expressed in radians, and the position (Fig. 6) and map RMS error (Fig. 8) are expressed in dimensionless distance units.

Comparing Figures 3, 4, and 5 against Figures 6, 7, and 8, it can be observed that the hybrid estimator is more accurate than the Bayesian estimator, even though the Bayesian estimator is using 10 times more particles.

Figure 9 shows the RMS error (of the map features’ positions) vs. time via Bayesian estimation, and Figure 10 shows the same data when using hybrid estimation. Different curves reflect the number of particles $K$ used. The computational complexity of FastSLAM 2.0 is equivalent for both standard Bayesian estimation and hybrid estimation, and linear in $K$. Note that it takes many more particles for standard Bayesian estimation to achieve the same level of accuracy as hybrid estimation.

From these plots, it can be concluded that hybrid estimation is more accurate and less computationally burdensome than Bayesian estimation for this problem.
B. Underwater Field Tests

To test in the field with real vision and complex lighting conditions, the algorithm was deployed to track and reconstruct an underwater target from a camera mounted on a Remotely Operated Vehicle (ROV). Testing was conducted in conjunction with the Monterey Bay Aquarium Research Institute (MBARI), using the research vessel Point Lobos and the ROV Ventana (shown in Figures 11 and 12). The target (shown in Figure 13) is undergoing unknown rotation and translation, due to forcing and torquing by ocean currents and buoyancy effects.

Figures 14 and 15 each show the target’s map estimate after 700 frames, and the relative pose estimate of the camera (in the target’s body frame) up to that time. The estimates in Figure 14 were calculated by the hybrid frame-to-frame pose tracking approach explained in this paper, with 100 particles in the particle filter. The estimates in Figure 15 were calculated by a purely Bayesian Rao-Blackwellised particle filter, with 1000 particles in the particle filter.

The true motion of the ROV was in a clockwise loop about the target (looking down on the target). Figure 14 correctly shows a smooth, feasible clockwise ROV path (the green line) and a reconstruction (the blue point cloud) which resembles the actual target (Fig. 13). By contrast, Figure 15 shows the Bayesian filter producing an incorrect pose estimate (with a counter-clockwise path around the target), and a reconstruction which does not resemble the true target.

As with the synthetic results, it can be observed that the hybrid estimator is more accurate for this problem than the Bayesian estimator, even though the Bayesian estimator is using 10 times more particles.

VII. Conclusion

A new method of performing pose tracking during monocular vision-only SLAM/SFM was described in this paper. The method is a hybridization of existing techniques for pose tracking. The motivation was real-time, smooth relative pose tracking of a tumbling object, based on observations from a camera-equipped robotic platform (while mapping is also taking place concurrently).

The results show the hybrid algorithm’s successful estimation of target pose (and shape), in situations where existing pose tracking methods for vision-only SLAM/SFM failed to estimate the target’s relative motion accurately and in real-time.

Acknowledgements

The authors thank the Monterey Bay Aquarium Research Institute for support, and NASA grant #Z627401-A for partial funding. Additionally, the authors thank the MBARI ROV pilots (Knute Brekke, Mike Burcynski, Craig Dawe, and D.J. Osborne) and crew of the Point Lobos, as well as Kiran Murthy and Peter Kimball, for their assistance in ROV operations which supported the research goals of this paper.

References

Fig. 3. Synth. Target, Hybrid Est., 50 particles: Position (-) vs. Time

Fig. 4. Synth. Target, Hybrid Est., 50 particles: Orientation (rad.) vs. Time

Fig. 5. Synth. Target, Hybrid Est., 50 particles: Map Error (-) vs. Time

Fig. 6. Synth. Target, Bayes Est., 500 particles: Position (-) vs. Time

Fig. 7. Synth. Target, Bayes Est., 500 particles: Orientation (rad.) vs. Time

Fig. 8. Synth. Target, Bayes Est., 500 particles: Map Error (-) vs. Time
Fig. 9. Synthetic Target, Bayes Estimation: Map Error (-) vs. Time

Fig. 10. Synthetic Target, Hybrid Estimation: Map Error (-) vs. Time

Fig. 11. MV Point Lobos

Fig. 12. ROV Ventana

Fig. 13. Field Target

Fig. 14. Field Results: Hybrid Estimator with 100 particles

Fig. 15. Field Results: Bayesian Estimator with 1000 particles