

Bayesian Filtering for Tracking Pose and Location of Rigid Targets

Anuj Srivastava

Florida State University, Tallahassee, FL 32306

ABSTRACT

Tracking of target pose is important for ATR in situations where there is a relative motion between the targets and the sensor(s). Taking a Bayesian approach, we formulate the problem of jointly tracking the target positions and orientations (as elements of $SE(3)$) as a problem in nonlinear filtering. Combining pertinent ideas from importance sampling and sequential methods, we apply an iterative Monte Carlo approach to solve for MMSE solutions. This tracking algorithm is demonstrated for tracking individual targets in a simulated environment.

Keywords: ATR, Target pose tracking, nonlinear filtering, sequential Monte Carlo methods, Newtonian dynamics

1. INTRODUCTION

In automated target recognition (ATR), the goal is to analyze observed images and recognize targets of interest contained in them. In the process of target recognition, estimation of nuisance parameters, such as target position and orientation, plays an important role. It is well known that a more accurate parameter estimator will lead to a better performance in target recognition; this relationship has been made precise using asymptotic considerations in Grenander et al.¹ In the case of dynamic scenes containing moving targets, a time sequence of these parameters has to be estimated and in this paper, we focus on the estimation (or tracking) of this time-series. In view of the nonlinearities inherent to the imaging process and complications in the resulting probability models, it is generally difficult to derive analytical estimates for positions and orientations. Hence, the goal is to devise a computational technique to track the positions and orientations of a dynamic rigid target, as the images are received at regular observation times.

For estimation problems with linear relationships among the time-varying parameters and between the observations and the parameters, under assumptions of additive Gaussian noise, one can utilize a Kalman filter for optimal (conditional-mean) estimator. Using recursive formulae, the Kalman filter provides both the optimal estimate and the error bound associated with that optimal estimate. Due to its computational simplicity and the optimality of results, Kalman filter is a widely used tool in inferences involving linear Gaussian systems. In situations with nonlinear relationships and/or non-additive, non-Gaussian noise, Kalman filter has often been extended to obtain desired results. One basic issue in these problems is that the posterior distribution is non-Gaussian, and is often multi-modal and non-symmetric. One such method relies on treating the time propagation of the posterior density directly, instead of trying to propagate the estimates. A pair of equations characterize the change in the posterior density between the discrete observation times. We will utilize these equations and the Monte Carlo sampling techniques to propagate the posterior, not as a complete function but approximately in terms of a finite number of samples obtained from it. These samples can then be averaged to estimate expectations (MMSE estimates) under the posterior.

As an illustration of the tracking problem, consider a target (a car) undergoing a random motion (translation and rotation), as shown in the upper panels of Figure 1. At each time, the target is imaged at a certain resolution and the resulting images form the observations for tracking. The lower panels display the (simulated) observed images of target, each image collected at the camera resolution of 32×32 pixels. In general, the problem of pose tracking is challenging due to the issues including:

Further author information: (Send correspondence to A.S.)

A.S: E-mail: anuj@stat.fsu.edu



Figure 1. Upper panels: the target rendered at several positions and orientations along its motion. Lower panels: the corresponding noisy images of target.

1. The functional relationship between the target pose parameters and the observed images is highly nonlinear. The observed pixels result from a physical process (which includes projection, obscuration, reflection etc.), and hence are nonlinear functions of pose parameters. For moving targets, the relationship between the target attributes from one time to the next may not be linear. Hence, the state equation is also a nonlinear differential equation. These two sources of nonlinearities result in a non-Gaussian posterior density on the parameter space.
2. For tracking rigid motion, the parametric representations of unknowns belong to spaces which are not vector spaces. The target orientation is represented by a 3×3 orthogonal matrix, an element of $SO(3)$ that is not a vector space, but instead a curved Lie group. Therefore, the sampling procedures and diagnostic techniques available for solutions on \mathbb{R}^n may not directly apply.
3. In many recognition setups, the dimensionality of the observation space (in other words, the image size) is much larger compared to the number of parameters being estimated. This implies that the resulting probabilities on the parameter space are restricted only to small subsets, i.e. the problem analysis is asymptotic. Due to target motion, these subsets are constantly changing in time and a major difficulty is to reach these high-probability sets from one time to another, in an efficient fashion. It is an important issue for the computational approaches since they mostly involve finite evaluations to approximate a limiting quantity and it becomes important to have some number of samples from these high-probability subsets.

In addition to these specific issues, there are a number of other factors which make the task of tracking difficult. We restrict ourselves to the above-mentioned issues by making the following assumptions, and briefly reference the techniques for possible extensions that can handle these other factors.

- A1** The geometry of the target is assumed to be rigid and known completely. In general, a target can be semi-rigid or flexible with geometry changing in time. Such cases can be handled by defining high-dimensional deformations for variations in shape and stating the Bayesian formulation on the space of these deformations (see for example Miller et al.,² Trounev,^{3,4} and Matejic⁵), instead of the low-dimensional, rigid transformations considered in this paper.
- A2** The texture of target surface is assumed fixed and known. A large class of textures can be modeled using Markov random fields with parametric descriptions. Therefore, by adding the texture parameters to the list of unknowns a more general problem can be formulated.
- A3** It is well known that the changes in illumination can cause dramatic changes in the images, and estimating illumination variables is an important step in target recognition. However, in this paper we avoid this discussion by assuming that the locations of the light sources are fixed and known. In general situations, the parameter space will get augmented by the random variables representing light variability leading to a larger tracking problem, see Hager et al.⁶

A4 We will assume that the target motion is relatively smooth. Stated more precisely later, it roughly implies that the velocities (translational and rotational) take larger variations from their predicted values only with smaller probabilities. This condition can be relaxed at the cost of a larger computational effort associated with the tracking.

A5 In the problem of image-based pose tracking, there may be other man-made or natural objects present in the scene, and therefore in the image. Efficient probability models for these unknown objects, often called clutter objects, are being investigated Mumford et al.^{7,8} and Grenander et al..⁹ In this paper, we avoid clutter-related issues by assuming that the background is completely known beforehand.

In an earlier paper, Grenander et al.,¹⁰ we have studied the problem of statistical inference on target orientation and position, in the context of stationary targets. In case of moving targets, we have investigated the improvement in tracking airplane positions by means of high-resolution images and Newtonian dynamics (Miller et. al.¹¹), using a jump-diffusion based algorithm. For the case of stationary objects, with constant position and orientation, Grenander et al.¹⁰ and Loizeaux et al.¹² have described a method for MMSE estimation of target position and orientation. We will apply these ideas to the case of moving targets and seek computationally efficient ways to update the estimates.

Pose tracking of rigid and articulate objects has been addressed by many researchers in a wide variety of contexts. Most frequent application involves rotation and translation of planar objects in 2-D images to match their occurrence in from one image to another (often called the *registration problem*). In three-dimensional considerations, the pose tracking problem is more natural due to the possible inclusion of the physics of the object motion. The problem can be complicated if there is no apriori assumption on the target geometry (shape). To address that issue, three-dimensional objects are often approximated by means of simplified geometries such as cylinders, cuboids, and ellipsoids, and their combinations, and are tracked using these simpler geometries, see for example.^{13–15} On the other hand, position tracking of point (unresolved) targets is also a well studied problem with some proposed solutions belonging to the nonlinear filtering theory, as described in Bar-Shalom et al..^{16,17}

In Section 2, we describe the representations we have chosen to model the system: the target model, the motion model and the sensor model. Section 3 formulates the target pose tracking as Bayesian nonlinear filtering problem and proposes a sampling-based solution. Section 4 specifies an algorithm for applying this methodology. Some experimental results are presented in Section 5.

2. TARGET AND SENSOR MODELS

An important issue in ATR is the choice of models representing targets and their motion. Taking a top-down approach, we assume the knowledge of three-dimensional target geometry (shape) and impose a projection model on the imager, to estimate motion parameters. It is well known that such a structured model-based representations lead to well-posed inference problem. Of course, it is important to have models that closely resemble the actual situation. Another useful aspect of this representation is the ability to include contextual information, which often goes under-utilized in many ATR algorithms.

As described later, we take a Monte Carlo sampling approach to generate samples from the posterior, and propagate these samples in time. Under this approach, we need to specify two quantities: (i) a functional form of the data-likelihood function, and (ii) an ability to sample from the conditional prior density on parameters (given estimates at the previous time). Next we derive these two forms:

2.1. Target Representation

First we describe our model-based, high-level representation of a target being imaged by a camera. This representation is based on selecting a prototype or a template for the object and transforming it to match its occurrences in a scene. In the context of rigid target-tracking, a template is constructed by choosing a CAD representation (e.g. triangulated mesh) of the two-dimensional surface of a rigid target. Let \mathcal{A} be the finite alphabet of all possible target labels.

$$\mathcal{A} = \{\phi, \alpha_1, \alpha_2, \dots, \alpha_n\}.$$

ϕ denotes no target or the null hypothesis. Associated with each label, $\alpha \in \mathcal{A}$, we assume a template I^α . This template constitutes all the target attributes which affect the sensor output, such as triangulated surface, surface material, and texture (see Grenander et al.¹⁰ for details of this representation). According to the assumption **A1**,

this geometric description of the target is assumed known completely. In the case of textured surfaces, the texture model is also assumed known via assumption **A2**. Shown in Figure 2 are the rendered templates and the triangulated meshes for some examples targets.

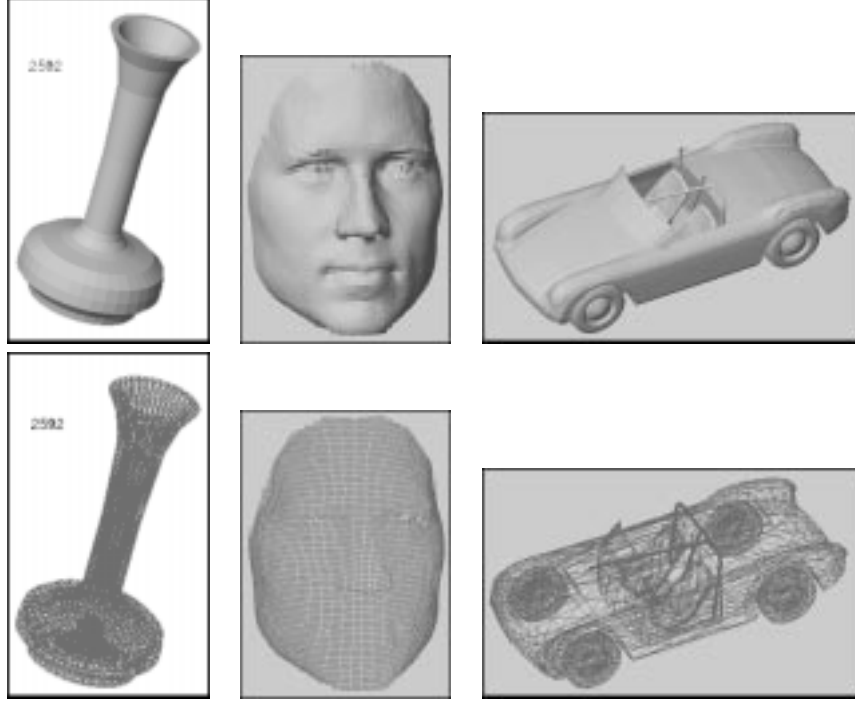


Figure 2. Templates for a vase, a screw, and a car frame: rendered surfaces (top) and the triangulated surfaces (bottom).

To generate motion, we introduce rigid transformations, translation and rotation, on these templates. Let p be a translation vector in \mathbb{R}^3 , then the notation pI^α implies that the vertices and normals in I^α are all translated by p . The template I^α , which was initially centered at the origin of a fixed reference frame, is now centered at p . Similarly for a rotation matrix $o \in SO(3)$, oI^α stands for rotated template with all the vertices and normals rotated by matrix o . Their combined action is represented by the special Euclidean group $SE(3) \equiv SO(3) \otimes \mathbb{R}^3$ according to

$$s = \begin{bmatrix} o & p \\ 0 & 1 \end{bmatrix}, \quad s \in SE(3).$$

A transformed target is denoted by sI^α . In computer simulations, this is often accomplished by appropriately changing the view point of the observer (keeping the target stationary). For a dynamic target, the rendered target is modeled as $s_t I^\alpha$ for a process, labeled by the observation index t , with s_t taking values in $SE(3)$.

2.2. Newtonian Motion Model

Since our goal is to track the target motion, a precise model on temporal relationship between s_{t-1} and s_t can be quite useful. Estimates from time $t-1$ are commonly used to initialize the estimation procedure at time t , but a dynamic model can further improve the performance. To build a probability model on target motion, we utilize the rigid-body dynamics as dictated by Newton's second law. Our approach is to impose a probability model on the forces and torques driving the target, and inherit a probability model on the resulting positions and orientations. Let $v_t = [v_{1,t} \ v_{2,t} \ v_{3,t}] \in \mathbb{R}^3$ and $\omega_t = [\omega_{1,t} \ \omega_{2,t} \ \omega_{3,t}] \in \mathbb{R}^3$ be the translational and rotational velocities of the target, respectively, in the body-reference frame. Inertial-frame positions, $p_t \in \mathbb{R}^3$, and orientations, $o_t \in SO(3)$, relate to the velocities according to,

$$\frac{do_t}{dt} = o_t \Omega_t, \quad \frac{dp_t}{dt} = o_t v_t, \quad (1)$$

where Ω_t is the skew-symmetric matrix made up of the angular velocities of the target,

$$\Omega_t = \begin{bmatrix} 0 & -\omega_{3,t} & \omega_{2,t} \\ \omega_{3,t} & 0 & -\omega_{1,t} \\ -\omega_{2,t} & \omega_{1,t} & 0 \end{bmatrix}. \quad (2)$$

According to Newton's second law, the rate of change of velocities depend upon the external forces and torques according to

$$\begin{aligned} \frac{dv_t}{dt} + \Omega_t v_t &= \frac{1}{m} F_t, \\ I_m \frac{d\omega_t}{dt} + \Omega_t I_m \omega_t &= T_t, \end{aligned}$$

where m is the target mass, and I_m is a 3×3 , diagonal matrix with entries given by the target's moment of inertias around the three axes. For discrete observation times $t = 1, 2, \dots$, the corresponding difference equations are given by

$$v_{t+1} = v_t - \Omega_t v_t + \frac{1}{m} F_t, \quad \omega_{t+1} = \omega_t - I_m^{-1} \Omega_t I_m \omega_t + T_t. \quad (3)$$

A probability model on the forces and torques, $\{(F_t, T_t) : t = 1, 2, \dots\}$, imposes a probability model on the resulting $\{s_t : t = 1, 2, \dots\}$. Assume that F_t and T_t are normally distributed with means \bar{F}_t and \bar{T}_t , and covariances Σ_1 and Σ_2 , respectively. Let Λ_1 and Λ_2 be the choleski square-roots of these covariances, respectively. This results in the random vectors, ω_{t+1} and v_{t+1} , having the conditional probability (given ω_t and v_t), as $N(v_t - \Omega_t v_t, \frac{1}{m^2} \Sigma_1)$ and $N(\omega_t - I_m^{-1} \Omega_t I_m \omega_t, \Sigma_2)$, respectively. The discrete form of Eqn. 1 is, for $t = 1, 2, \dots$,

$$o_{t+1} = o_t \exp(\Omega_t), \quad p_{t+1} = p_t + o_t v_t, \quad s_{t+1} = \begin{bmatrix} o_{t+1} & p_{t+1} \\ 0 & 1 \end{bmatrix}, \quad (4)$$

where \exp denotes the matrix exponential. For algorithmic implementation, it is useful to define the inverse map, given s_t 's the velocities are given by

$$\Omega_t = \log(o_{t+1} o_t^T), \quad v_t = o_t^T (p_{t+1} - p_t). \quad (5)$$

Using this transformation, back and forth, we can write down the conditional density on s_t given s_{t-1} , $f(s_t | s_{t-1})$. Explicit formulation of this density requires calculation of the Jacobian of the transformation, from velocities to positions and orientations, as described in Srivastava et al.¹⁸ However, instead of an explicit expression, we only need to be able to generate samples of s_t given s_{t-1} . This can be accomplished by first sampling for v_t and ω_t from their normal densities and then substituting them in Eqn. 4 to convert the velocities into s_t .

2.3. Observation Model

Next we specify the probability model for the observed images. We will assume that remote sensing corresponds to a mapping, denoted by T , from the space of target descriptions $\{s_t I^\alpha\}$ to the images, $T : s_t I^\alpha \mapsto I_t$, $I_t \in \mathcal{I}$ is the measurement space, assumed to be a finite dimensional vector space. T is mostly an orthographic or a perspective projection from the three-dimensional volume containing the target to the two-dimensional image space, as shown in Figure 3. I_t is the observed image of the target α at the transformation s_t and is modeled as a random realization with mean given by $T s_t I^\alpha$. The exact nature of I_t depends upon the sensor being used. For example, in case of CCD imaging I_t is Poisson distributed with mean given by $T s_t I^\alpha$, as in Snyder et al.¹⁹ in case of LADAR imaging I_t is modeled as $T s_t I^\alpha$ attenuated by Rayleigh weights (see Shapiro et al.²⁰), and for video imagers I_t is a Gaussian field with mean given by $T s_t I^\alpha$ and covariance K (see Grenander et al.^{10,1}). with the mapping T depending upon the sensor used. For the video case, let the observation space be the two-dimensional focal plane of the camera lens perpendicular to the line of sight. The image formed on the focal plane is sampled on a finite, square lattice of size d . We will use the following notation: for a matrix X with total elements d , and a $d \times d$ matrix A , let $\|X\|_A$ denote the quadratic form $X_v^\dagger A X_v$ where X_v is the column-vector formed by arranging columns of X one below another and \dagger denotes the transpose. Under Gaussian model, the image-likelihood function is given by,

$$f(I_t | s_t) = \frac{1}{Z} \exp\left(-\frac{1}{(2\pi\sigma^2)^{d/2}} \|I_t - T s_t I^\alpha\|_{K^{-1}}^2\right). \quad (6)$$

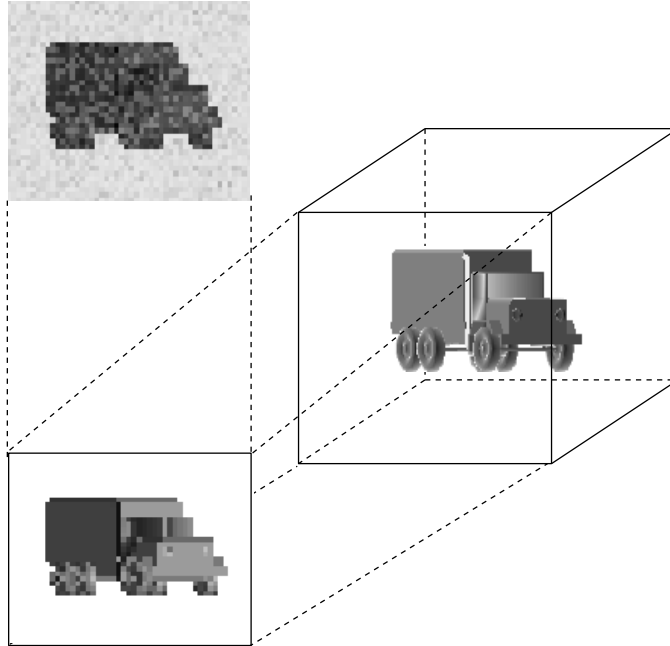


Figure 3. Depiction of imaging as a projection operator.

Having specified a prior probability model on s_t and the likelihood function, we can write down the posterior using Bayes' rule (*posterior* \propto *prior* \times *likelihood*). For writing densities on $SE(3)$, we choose the reference measure to be the direct product (denoted by $\gamma(ds)$) of the unit Haar measure on $SO(3)$ and the Lebesgue measure on \mathbb{R}^3 .

3. POSE TRACKING

In this section we state the Bayesian, nonlinear-filtering formulation of the pose tracking problem and propose a Monte-Carlo sampling solution.

3.1. Bayesian Non-linear Filtering

In the context of pose tracking, the Bayesian filtering problem can be described as follows. For the discrete observation times $t = 1, 2, \dots$, let the target motion be represented by the sequence $s_1, s_2, \dots \in SE(3)$, and the observed image sequence be given by $I_1, I_2, \dots \in \mathcal{I}$.

Problem: Given the observations $I_{1:t} = \{I_1, \dots, I_t\}$, estimate the sequence $s_{1:t} = \{s_1, s_2, \dots, s_t\}$ using a minimum mean-squared error (MMSE) criterion.

In this Bayesian approach, we have to derive a posterior probability model on the unknown sequence and seek computational techniques to generate MMSE estimates. As t increases, the underlying parameter space grows and the posterior density changes. In recent papers,^{23,22,21,24} an efficient procedure to solve such Bayesian problems has been suggested. This procedure restricts inference to only the last time t and utilizes a Monte Carlo technique to sample from the posterior probability associated with s_t . Furthermore, a recursive algorithm takes samples from posterior density of s_{t-1} and generates samples from the posterior density of s_t . MMSE estimates are based on the expectations of the type: for some function $g : SE(3) \rightarrow \mathbb{R}$,

$$\theta_t = \int_{SE(3)} g(s_t) f(s_t | I_{1:t}) \gamma(ds_t). \quad (7)$$

We will estimate θ_t by averaging samples obtained from the posterior $f(s_t | I_{1:t})$, for each time t .

In a time-series analysis, there is a standard characterization of the time-varying posterior density, in a convenient recursive form. This characterization is via the following two filtering equations, for $t = 2, 3, \dots$

$$f(s_t|I_{1:t-1}) = \int_{SE(3)} f(s_t|s_{t-1})f(s_{t-1}|I_{1:t-1})\gamma(ds_{t-1}), \quad (8)$$

$$f(s_t|I_{1:t}) = \frac{f(I_t|s_t)f(s_t|I_{1:t-1})}{f(I_t|I_{1:t-1})}. \quad (9)$$

Eqn. 8 is called the prediction equation and Eqn. 9 is called the update equation. The denominator in Eqn. 9 is difficult to compute and, for a given observation set, is a constant; we will denote it by Z_t . One distinct advantage of the Monte Carlo approaches is that this normalizing constant need not be explicitly evaluated. This relationship between Eqns. 8 and 9 suggests a recursive form for the solutions derived from the posteriors $f(s_{t-1}|I_{1:t-1})$ and $f(s_t|I_{1:t})$.

3.2. Sequential Importance Sampling/Resampling

Motivated by the temporal structure in the problem, we propose a sequential sampling algorithm utilizing importance sampling ideas. The general framework is laid out in Liu²¹ while these methods have previously been used in tracking as described in the papers.^{22,23} In general, a sampling approach is to generate samples $\{s_t^{(i)} : i = 1, 2, \dots, M\}$ from the posterior, $f(s_t|I_{1:t})$, and approximate the integral in Eqn. 7 by the sample average according to

$$\int_{SE(3)} g(s_t)f(s_t|I_{1:t})\gamma(ds_t) \approx \frac{1}{M} \sum_{i=1}^M g(s_t^{(i)}) \equiv \hat{\theta}_{t,M}. \quad (10)$$

$\hat{\theta}_{t,M}$ is a random variable with mean θ and variance

$$\text{var}(\hat{\theta}_{t,M}) = \frac{1}{M} \left[\int_{SE(3)} g(s_t)^2 f(s_t|I_{1:t})\gamma(ds_t) - \theta^2 \right].$$

For an appropriate function g , the estimate $\hat{\theta}_{t,M}$ converges to θ_t as M gets larger. In view of the complicated relationship between the parameter s and the observation I , it is difficult and often inefficient to sample directly from the posterior $f(s_t|I_{1:t})$. A recursive formulation, which takes samples from $f(s_{t-1}|I_{1:t-1})$ and transforms them into samples from $f(s_t|I_{1:t})$ in an efficient fashion, is desirable. Assume that, for time $t - 1$, we have the set of M samples from the posterior $f(s_{t-1}|I_{1:t-1})$, forming the sample set,

$$S_{t-1} = \{s_{t-1}^{(i)} \in SE(3) : i = 1, 2, \dots, M\}.$$

Following are the steps which utilize elements of S_{t-1} to generate the set S_t .

1. The first step is to sample from $f(s_t|I_{1:t-1})$ given the samples from $f(s_{t-1}|I_{1:t-1})$. We take a *compositional approach* by treating $f(s_t|I_{1:t-1})$ as a mixture density. According to Eqn. 8, $f(s_t|I_{1:t-1})$ is the integral of the product of a marginal and a conditional density. This implies that, for each element $s_{t-1}^{(i)} \in S_{t-1}$, by generating a sample from the conditional, $f(s_t|s_{t-1}^{(i)})$, we can generate identically distributed samples from $f(s_t|I_{1:t-1})$. This method is useful only when there is an efficient technique to sample from the prior density $f(s_t|s_{t-1})$. In our case, Section 2.2 describes a simple technique to sample from this prior. Now we have samples $\{\tilde{s}_t^{(i)}\}$ from $f(s_t|I_{t-1})$; corresponding to the Kalman-filtering notation, these samples are called *predictions*.
2. Next step is to sample from the posterior $f(s_t|I_{1:t})$ given the predictions $\tilde{s}_t^{(i)}$, $i = 1, 2, \dots, M$. Towards that end, rewrite Eqn. 7 as

$$\frac{1}{Z_t} \int_{SE(3)} g(s)f(I_t|s_t)f(s_t|I_{1:t-1})\gamma(ds_t). \quad (11)$$

This form motivates the use of *importance sampling* (see Roberts,²⁴ pg 80 for a description). Importance sampling essentially implies generating samples from $f(s_t|I_{1:t-1})$ and multiplying appropriate weights to generate

sample averages from $f(s_t|I_{1:t})$. The likelihood function, $f(I_t|s_t)$ as a function of s_t , is called the *importance function* and provides the weights. Then, the importance sampling approximates the desired expectation according to

$$\int_{SE(3)} g(s) f(s_t|I_{1:t}) \gamma(ds_t) \approx \frac{1}{M} \sum_{i=1}^M g(\tilde{s}_t^{(i)}) \frac{f(I_t|\tilde{s}_t^{(i)})}{Z_t}. \quad (12)$$

Since $Z_t = f(I_t|I_{1:t-1}) = \int_{SE(3)} f(I_t|s_t) f(s_t|I_{1:t-1}) \gamma(ds_t)$, it can be approximated according to

$$Z_t \approx \frac{1}{M} \sum_{I=1}^M f(I_t|\tilde{s}_t^{(i)}).$$

Substituting back,

$$\int_{SE(3)} g(s) f(s_t|I_{1:t}) \gamma(ds_t) \approx \sum_{i=1}^M g(\tilde{s}_t^{(i)}) \beta_{t,i}, \quad \text{where } \beta_{t,i} = \frac{f(I_t|\tilde{s}_t^{(i)})}{\sum_{j=1}^M f(I_t|\tilde{s}_t^{(j)})}. \quad (13)$$

This calculation shows why it is not necessary to evaluate the constant in the likelihood function.

3. In such imaging applications, characterized by large-sized data for estimating a small number of parameters, it is often possible that the likelihood of certain samples is very small, resulting in negligible values of $p_{t,i}$ for certain i . Although the corresponding samples $\tilde{s}_t^{(i)}$ do not contribute much in the computation of the estimates, they are still utilized to generate the samples for the next time and hence are carried forward despite having very small probabilities. One way to approach this problem is through resampling (Liu²¹ has named it SISR, sequential importance sampling with resampling). This step takes the sample set $S_t = \{\tilde{s}_t^{(i)} : i = 1, 2, \dots, M\}$ and generates M samples from it according to the probability mass function $\beta_{t,i}$ (as defined in Eqn. 13). The resampled set is denoted by

$$S_t = \{s_t^{(i)} \in SE(3) : i = 1, 2, \dots, M\},$$

and the estimate is calculated as

$$\hat{\theta}_{t,M} = \frac{1}{M} \sum_{i=1}^M g(s_t^{(i)}). \quad (14)$$

Note that the resampled set will have values that appear multiple times since they have higher probabilities $\beta_{t,i}$. Similarly, the values with negligible $\beta_{t,i}$'s may not be present in the resampled set.

It can be shown that, as the sample size $M \rightarrow \infty$, the estimator

$$\hat{\theta}_{t,M} = \frac{1}{M} \sum_{i=1}^M g(s_t^{(i)})$$

converges to θ_t in the total variation norm.

3.3. MMSE of Target Pose

The remaining issue is to generate MMSE estimates from the samples $\{s_t^{(i)} : i = 1, 2, \dots, M\}$ generated from the posterior. Since $SE(3)$ is a curved Lie group, we have to choose a definition of error to make this goal precise. In papers,^{12,10} we have chosen the Euclidean distance function to define the estimation error, and under this derivation established MMSE estimators on $SO(3)$ and $SE(3)$. In addition to an MMSE estimator, a lower bound on the estimation error (in the expected squared error sense) is also derived and it is established that the MMSE estimator achieves this lower bound. For the posterior density $f(s_t|I_{1:t})$, the MMSE is defined to be

$$\hat{s} = \operatorname{argmin}_{s \in SE(3)} \int_{SE(3)} d(s, s_t)^2 f(s_t|I_{1:t}) \gamma(ds). \quad (15)$$

With $g(s_t) = d(s, s_t)^2$, the integral is approximated by the sample mean

$$\hat{s}_{t,M} = \operatorname{argmin}_{s \in SE(3)} \frac{1}{M} \sum_{i=1}^M d(s, s_t^{(i)})^2. \quad (16)$$

The MMSE estimator, in its matrix form, is given by

$$\hat{s}_{t,M} = \begin{bmatrix} \hat{o}_{t,M} & \hat{p}_{t,M} \\ 0 & 1 \end{bmatrix}, \quad \text{where } \hat{p}_t = \frac{1}{M} \sum_{i=1}^M p_t^{(i)}, \quad \text{and } \hat{o}_t = u \Delta v^\dagger, \quad (17)$$

for $a = \frac{1}{M} \sum_{i=1}^M o_t^{(i)}$, and the singular value decomposition $a = u \sigma v^\dagger$. Δ is a diagonal matrix with all elements +1 except the last element which is -1 if $\operatorname{determinant}(a) < 0$. Please refer to Loizeaux et. al.¹² for the proof.

4. POSE TRACKING ALGORITHM

In this section we state a precise algorithm for target pose tracking.

An important issue that we have not addressed yet is that of initializing the tracking algorithm, i.e. how to sample the initial pose and velocities. We will utilize a jump-diffusion sampling (see Miller et al.¹¹ for details) approach based on global search in $SE(3)$, even though it is going to be computationally expensive. Once the samples for initial pose are available, they can be used to initialize the algorithm.

Assume that for any time $t - 1$, we have the samples $S_{t-1} = \{s_{t-1}^{(i)} : i = 1, 2, \dots, M\}$ and the corresponding velocities $\{(v_{t-2}^{(i)}, \omega_{t-2}^{(i)}) : i = 1, 2, \dots, M\}$. The following algorithm outlines the steps to generate the samples for t :

Algorithm

1. **Sample Conditional:** Draw $\{(v_{t-1}^{(i)}, \omega_{t-1}^{(i)}), i = 1, 2, \dots, M\}$ from the conditional prior as follows. Set $i = 1$.

(a) Generate two random vectors $x_i, y_i \sim N(0, Id) \in \mathbb{R}^3$.

(b) Set $\omega_{t-1}^{(i)} = \omega_{t-2}^{(i)} - I_m^{-1} \Omega_{t-2}^{(i)} I_m \omega_{t-2}^{(i)} + \Lambda_2 x_i$, and $v_{t-1}^{(i)} = v_{t-2}^{(i)} - \Omega_{t-2}^{(i)} v_{t-2}^{(i)} + \frac{1}{m} \Lambda_1 y_i$. Form $\Omega_{t-1}^{(i)}$ according to Eqn. 2.

(c) Compute $\tilde{o}_t^{(i)} = o_{t-1}^{(i)} \exp(\Omega_{t-1}^{(i)})$, $\tilde{p}_t^{(i)} = p_{t-1}^{(i)} + v_{t-1}^{(i)}$, and form

$$\tilde{s}_t^{(i)} = \begin{bmatrix} \tilde{o}_t^{(i)} & \tilde{p}_t^{(i)} \\ 0 & 1 \end{bmatrix}.$$

(d) Set $i = i + 1$. If $i < M$, go to Step (a).

2. **Importance Weights:** Compute the incremental weights as $w_t^{(i)}$ according to

$$w_t^{(i)} = \exp\left(\frac{-1}{2\sigma^2} \|I_t - \tilde{s}_t^{(i)} I^\alpha\|^2\right),$$

and form the probabilities

$$\beta_{t,i} = \frac{w_t^{(i)}}{\sum_{j=1}^M w_t^{(j)}}.$$

3. **Resampling:** Generate M samples from the set $\{\tilde{s}_t^{(i)}, i = 1, 2, \dots, M\}$ with the associated probabilities $\{\beta_{t,i}, i = 1, 2, \dots, M\}$. Denote these samples by $\{s_t^{(i)}, i = 1, 2, \dots, M\}$.

4. **MMSE Averaging:** The estimated position of the target at t is given by:

$$\hat{p}_{t,M} = \frac{1}{M} \sum_{i=1}^M p_t^{(i)},$$

and the estimated orientation is given by

$$\hat{o}_{t,M} = u \Delta v^\dagger, \quad \text{for } a = u \sigma v^\dagger, \quad \text{and } a = \frac{1}{M} \sum_{i=1}^M o_t^{(i)}.$$

5. Using the inverse mapping update the velocities according to

$$\Omega_{t-1}^{(i)} = \log(o_{t-1}^{(i)} (o_t^{(i)})^T), \quad v_{t-1}^{(i)} = (o_{t-1}^{(i)})^T (p_t^{(i)} - p_{t-1}^{(i)}).$$

Extract the elements of $\omega_{t-1}^{(i)}$ from $\Omega_{t-1}^{(i)}$.

6. Set $t \leftarrow t + 1$ and go to step 1.

There are two sources of error in this estimation procedure. First, the MMSE estimate as defined in Eqn. 15 may not be same as the true underlying target pose, and second, this MMSE estimate is further approximated by its sample mean according to Eqn. 16 introducing a sampling error. The expected value of the first error is lower bounded by Hilbert-Schmidt bounds as described in papers.^{10,12} For the second error, we propose to derive Chebyshev type inequality to quantify the confidence interval associated with the sampling error.

5. EXPERIMENTAL RESULTS

To illustrate the algorithm we present some experimental results. Consider the problem of tracking the pose of a toy car as shown in Figure 1. According to assumption **A1**, we assume that the geometry of this car is completely known. The light sources are kept stationary through this experiment. The motion is created using Newtonian dynamics model described in Section 2.2. Shown in Figure 4 upper panels is a sequence of pictures showing car motion in a three-dimensional scene. Every unit time the car is imaged by the frame grabber at a chosen camera resolution (32×32 or 64×64). Middle panels in Figure 4 display the noisy images of the moving car at 32×32 resolution. These images are then used according to algorithm given in Section 4. In this experiment, for initializing the algorithm, the target positions and orientation at $t = 1$ and $t = 2$ as sampled using a jump-diffusion algorithm. Rendered car at the estimation parameters \hat{s}_t for a sample size $M = 50$ is shown in the lower panels.

Figure 5 shows the estimation results for another example. This time a couple clutter objects have been added in the scene to generate non-homogeneous noise.

ACKNOWLEDGMENTS

This work was supported in part by ARO DAAG55-98-1-0102, NSF-9871196, ARO CIS DAA-H04-95-1-0494 and DAAD19-99-1-0267.

REFERENCES

1. U. Grenander, A. Srivastava, and M. I. Miller, "Asymptotic performance analysis of bayesian object recognition," *to appear in IEEE Transactions of Information Theory*, June, 2000.
2. M. Miller, G. Christensen, Y. Amit, and U. Grenander, "Mathematical textbook of deformable neuroanatomies," *Proceedings of the National Academy of Science* **90**, Dec. 1993.
3. A. Troune, "An infinite dimensional group approach for physics based models in pattern recognition," *in preparation*, 1999.
4. A. Troune, "Diffemorphisms groups and pattern matching in image analysis," *International Journal of Computer Vision* **28**(3), pp. 213–221, 1998.
5. L. Matejic and U. Grenander, "Group cascades for modeling anatomical variability in human brain," *personal communication*, 1997.

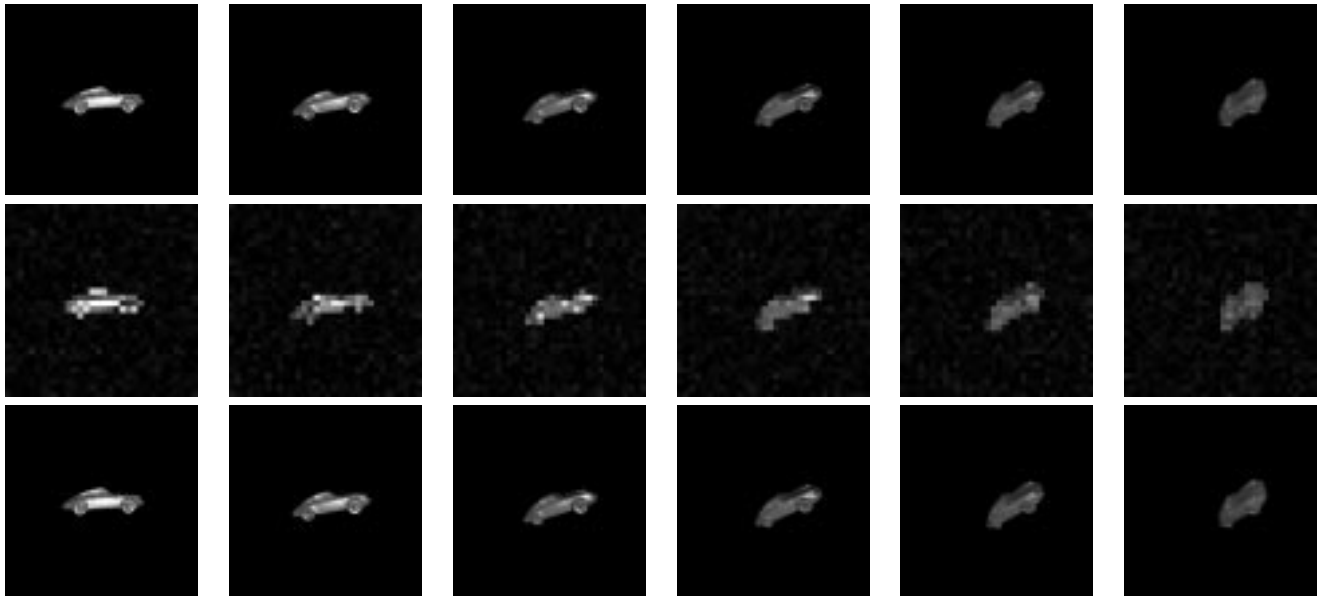


Figure 4. Upper panels: the target rendered at several positions and orientations along its motion. Middle panels: the corresponding noisy images of target. Lower panels: target rendered at the estimated positions and orientations.

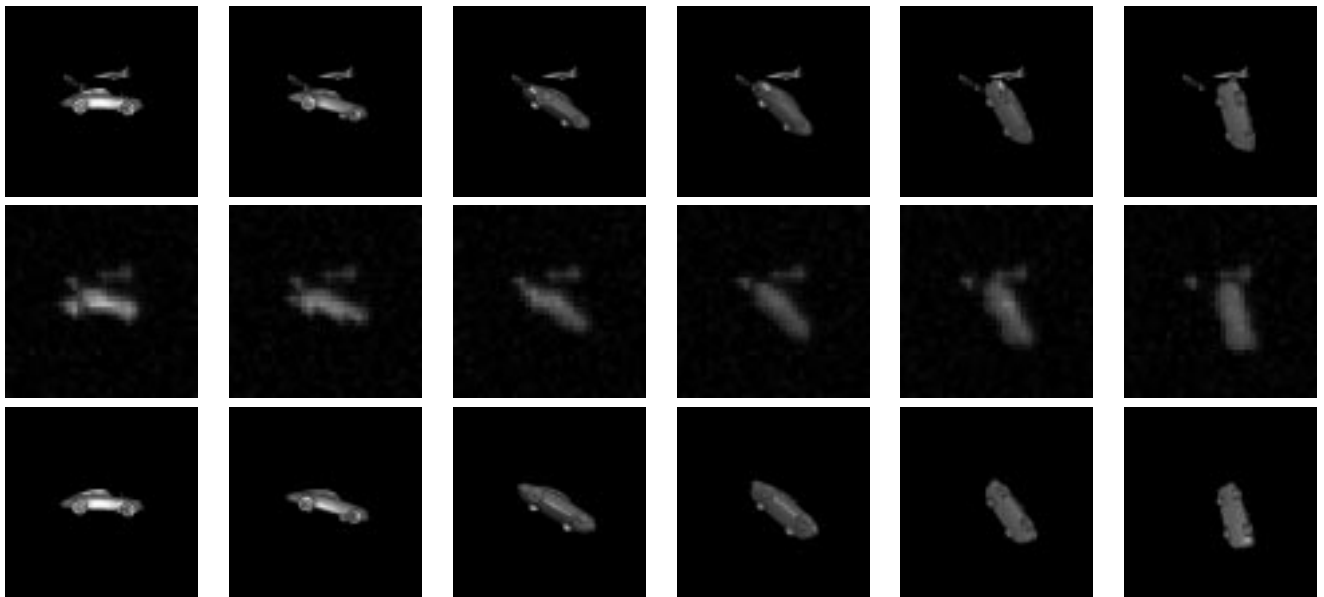


Figure 5. Upper panels: the target rendered at several positions and orientations along its motion. Middle panels: the corresponding noisy images of target. Lower panels: target rendered at the estimated positions and orientations.

6. G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**, October 1998.
7. A. B. Lee and D. Mumford, "An occlusion model generating scale-invariant images," *Personal Communication* , 1999.
8. J. Huang and D. Mumford, "Statistics of natural images and models," *Personal Communication* , 1999.
9. U. Grenander and A. Srivastava, "Probability models for background clutter in natural images," *Monograph of Department of Statistics, Florida State University* , 2000.
10. U. Grenander, M. I. Miller, and A. Srivastava, "Hilbert-schmidt lower bounds for estimators on matrix lie groups for atr," *IEEE Transactions on PAMI* **20**(8), pp. 790–802, 1998.
11. M. I. Miller, A. Srivastava, and U. Grenander, "Conditional-expectation estimation via jump-diffusion processes in multiple target tracking/recognition," *IEEE Transactions on Signal Processing* **43**, pp. 2678–2690, November 1995.
12. M. Loizeaux, A. Srivastava, and M. I. Miller, "Pose/location estimation of ground targets," in *Proceedings of SPIE*, (Orlando, FL), April 1999.
13. E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau, "Robust visual tracking by coupling 2d motion and 3d pose estimation," *Proceedings of ICIP* **4**, pp. 98–102, 1999.
14. Q. Delamarre and O. Faugeras, "3d articulated models and multi-view tracking with silhouettes," *Proceedings of seventh ICCV* **2**, pp. 716–721, 1999.
15. S. Clippingdale and T. Ito, "A unified approach to video face detection, tracking and recognition," *Proceedings of ICIP* **1**, pp. 662–666, 1999.
16. Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, 1988.
17. E. Y. Bar-Shalom, *Multitarget-Multisensor Tracking*, Artech House, 1990.
18. A. Srivastava, U. Grenander, G. R. Jensen, and M. I. Miller, "Jump-diffusion markov processes on orthogonal groups for object recognition," *accepted for publication by Journal of Statistical Planning and Inference* , December, 1999.
19. D. Snyder, A. Hammoud, and R. White, "Image recovery from data acquired with a charge-coupled-device camera," *Journal of the Optical Society of America A* **10**, pp. 1014–1023, May 1993.
20. J. H. Shapiro, B. A. Capron, and R. C. Harney, "Imaging and target detection with a heterodyne-reception optical radar," *Applied Optics* **20**(19), pp. 3292–3313, 1981.
21. J. S. Liu and R. Chen, "Sequential monte carlo methods for dynamic systems," *Journal of the American Statistical Association* **93**, pp. 1032–44, September 1998.
22. A. Blake and M. Isard, *Active Contours*, Springer, 1998.
23. N. J. Gordon, D. J. Salmon, and A. F. M. Smith, "A novel approach to nonlinear/non-gaussian bayesian state estimation," *IEEE Proceedings on Radar Signal Processing* **140**, pp. 107–113, 1993.
24. C. P. Robert and G. Castella, *Monte Carlo Statistical Methods*, Springer Text in Statistics, 1999.