Elsevier Editorial(tm) for Image and Vision Computing
Manuscript Draft

Manuscript Number:

Title: FPIV'04:Efficient Particle Filtering Using RANSAC with Application to 3D Face Tracking

Article Type: Special Issue Paper

Keywords: Random Projection; RANSAC; Particle Filtering;
Robust 3D Face Tracking

Corresponding Author: Le Lu

Other Authors: Xiangtian Dai; Gregory D. Hager ,

Efficient Particle Filtering Using RANSAC with Application to 3D Face Tracking

Le Lu Xiangtian Dai Gregory Hager

Computational Interaction and Robotics Lab
Computer Science Department
the Johns Hopkins University
Baltimore, MD 21218, USA

Abstract

Particle filtering is a very popular technique for sequential state estimation. However, in high-dimensional cases where the state dynamics are complex or poorly modeled, thousands of particles are usually required for real applications. This paper presents a hybrid sampling solution that combines RANSAC and particle filtering. In this approach, RANSAC provides proposal particles that, with high probability, represent the observation likelihood. Both conditionally independent RANSAC sampling and boosting-like conditionally dependent RANSAC sampling are explored. We show that the use of RANSAC-guided sampling reduces the necessary number of particles to dozens for a full 3D tracking problem. This is method is particularly advantageous when state dynamics are poorly modeled. We show empirically that the sampling efficiency (in terms of likelihood) is much higher with the use of RANSAC. The algorithm has been applied to the problem of 3D face pose tracking with changing expression. We demonstrate the validity of our approach with several video sequences acquired in an unstructured environment.

Key words: Random Projection, RANSAC, Particle Filtering, Robust 3D Face Tracking.

1 Introduction

In recent years there has been a great deal of interest in applying Particle Filtering (PF), also known as Condensation or Sequential Importance Sampling (SIS [16, 17]), to computer vision problems. Applications on parameterized or non-parameterized contour tracking [13, 14, 28], and human tracking [3, 18] have demonstrated its usefulness.

However, the performance of SIS depends on both the number of particles and the accuracy of the dynamic model. Given a specific error margin, the number of the particles required is generally determined by the dimension and structure of the state space [7]. A typical 6-DOF tracking problem usually requires thousands of particles [7]; reducing the number of particles by training a finely tuned dynamic model is not trivial [20] and sometimes not even possible.

On the other hand, the RANSAC algorithm [8] is often applied as a robust estimation technique. The basic idea behind RANSAC is to evaluate many small batches of observations, and from those to choose a larger set of “good” observation from which a robust state estimate can be produced. Although well-suited for producing single state estimates, RANSAC by itself does not preserve multiple solutions from frame to frame in a probabilistic inference framework.

In our proposed algorithm RANSAC-PF (or RANSAC-SIS), randomly selected feature correspondences are used to generate state hypotheses between pairs of frames in video sequences. However, instead of looking for a single best solution, the projections are used to guide the propagation of resampled particles. These particles are then reweighted according to a likelihood function and resampled. Consequently, the combined process not only serves as a robust estimator for a single frame, but provides stability over long sequences of frames.

The evaluation of solution quality is an issue of critical importance for all tracking problems, stochastic or deterministic. With the sampling concept, it is straightforward to infer the tracking quality from the state parameters’ posterior probabilistic distribution. We define an entropy-based criterion to evaluate the statistical quality of the tracked density function. In addition to providing a quality measure, the entropy values computed during tracking can help us extract some well tracked frames as exemplars [28]. When necessary, these exemplars can be archived as ”key frames” which are used to further stabilize the tracking.

The remainder of this paper is organized as follows. Related work is presented in section 2, followed by a description of the RANSAC-PF algorithm in section 3. In section 4, we discuss the sampling efficiency of RANSAC-PF. Section 5 describes a 3D face tracking application that uses our RANSAC-PF algorithm and presents experimental results. Finally, we offer conclusions and discuss future work.

2 Related Work

Deterministic parameter estimation algorithms, $e.g.$ linear mean-square methods, normally produce more direct and efficient results when compared with Monte Carlo-style sampling methods. On the other hand, deterministic algorithms are unfortunately easily biased and cannot recover from accumulated estimation errors. Robust estimators, such as LMedS [33] and MLESAC [25], follow the strategy of ”Winner Takes All” to get, with high probability, the maximum likelihood (ML) estimate from contaminated data. However, as discussed in the next section, those estimators are are often not suitable for a sequential estimation problem for a dynamic system because the estimation error in each stage can accumulate and result in failure on sequential data.

Particle filters (or, more generally, sequential important sampling [**?**]), are a family of techniques for recursive estimate that use sampling methods to approximate the optimal Bayesian filter. However, it is shown in [12] that sequential importance sampling or particle filtering may have non-zero probability to condense into an incorrect absorbing state when the number of samples are finite, even though particle filtering techniques can, under suitable assumptions, be accurate in an asymptotic sense. In our work, we propose a hybrid sampling approach to achieve a good balance of sampling efficiency and dynamic stability. From a particle filter viewpoint, both random geometric projections from RANSAC sampling of image features and importance resampling guide the time series evolution of state particles. Generally, simple resampling includes many low weight particles and thus generates poor results. Tu et al. showed that using data-driven techniques (like RANSAC in our case), such as clustering and edge detection, to compute importance proposal probabilities (for particle filter in our case), effectively drives the Markov chain dynamics and achieves tremendous speedup in comparison to the traditional jump-diffusion method [29].

There are several other papers integrating particle filters with variational optimization or observation-based importance functions. Sullivan et al. showed in [24] that random particles can be guided by a variational search, with good convergence when the image differences between frames are low. They used a predefined threshold to switch between the probabilistic and deterministic tracking engines. Isard et al. [14] presented an approach (ICondesation) to combine low-level and high-level information by importance resampling with a particle filter.

In more closely related work, Torr and Davidson [26] compute structure from motion by hybrid sampling (IMPSAC). They built a hierarchical sampling architecture with a RANSAC-MCMC estimator at the coarse level and a SIR-MCMC estimator at the finer levels. In this paper, we use sequential sampling-importance-resampling (SIR) technique to regularize and smooth the object pose estimation from spatial RANSAC sampling. As such, our technique considers the robust estimation problem from the viewpoint of time series analysis, while Torr et al. constrained the output of RANSAC with a MCMC formulated building model in 3D scene reconstruction.

To demonstrate the efficacy of our technique, we develop a system for 3-D face pose tracking. Although there are many solutions for 3-D face pose tracking, most algorithms are some variation on direct methods (e.g. SSD tracking) or feature-based matching. SSD-like methods [9, 15, 32, 1] have attracted much interest and have become a standard technique for many tracking problems. However, a classical deterministic SSD tracker requires a good prediction of target location in order to converge reliably. As a result, unexpected motion jumps can cause loss of tracking, and thus require additional apparatus to guarantee robustness [27].

On the other hand, feature-based methods have the advantage of potentially providing good solutions, provided stable features and good correspondences are available. The advantage of particle-filtering-based methods is that such a correspondence need not be produced explicitly, but is implicit in the likelihood function. Thus, good sampling combined with a good likelihood function can yield good tracking results, even in cluttered situations [18]. However, in cases where motions are abrupt and poorly modeled, and features are unstable, particle-based methods can still fail unless extremely large numbers of samples are generated. For this reason, the majority of particle-filter-based visual tracking consider restricted cases of two-dimensional motion. By including an explicit correspondence search as part of the sampling process, we are able to achieve comparable tracking results [15] under many strong distractions, with much fewer ($50 \sim 200$) particles, auto-recoverable for a full 3-D pose tracking problem.

We also develop a boosting-like RANSAC method to even more efficiently sample particles from the image observations. Okuma et al. [21] utilize Adaboost [31] to generate new detection hypotheses of mixture particle filtering for multi-target tracking. In comparision, we employ the boosting principle to guide the interaction of sampling between two particles. Our method is similar in nature to the iteratively boosted mixture modeling of Pavlovic [23]. He describes an optimization based approach to maximize the data’s likelihood given an estimated Gaussian mixture model. Data are then weighted inversely to their current likelihood values and these weights are then used to to calculate the new kernel of the mixture, and so on. As a result, data poorly represented by the previous mixture model have more weights in the next time step.

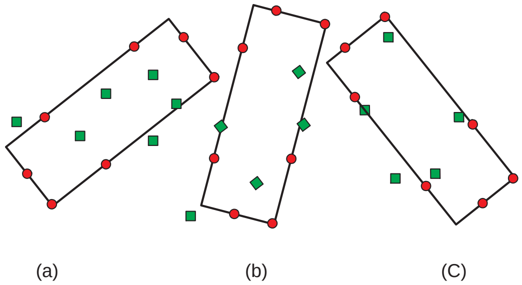


Figure 1: An example of RANSAC sampling of feature points (red dots are inliers, and green squares are outliers.) to track planar motions.

3 The RANSAC-PF Algorithm

3.1 Motivation

In order to motivate RANSAC-PF, consider the situation shown in Figure 1. Here we consider in-plane rotations and translations of a planar object through three frames of a video sequence. In each frame, a set of feature points $\{z_i^{(t)}\}$ are detected on the object. Some of these features are common among frames, and some are inconsistent or spurious. To track the object, we make use of RANSAC [8] to compute the incremental motion between successive frames, and then integrate these solutions over time to compute a state estimate. While this is computationally convenient, the lack of distributional information means that a single incorrect estimation step can be disastrous. More precisely, RANSAC samples some a number, l of subsets of the observed features. Let q denote the probability that one or more of these l subsets yields a correct correspondence which is detected by RANSAC. Although q can, in principle, be made arbitrarily large by increasing l, it will still generally be less than 1. As a result, the overall probability of consistently correct solutions over t frames, q^t, quickly decreases to zero as t grows. One way to avoid this problem is to include a time series model that maintains and regularizes multiple solutions over time. This is exactly the goal of our RANSAC-PF algorithm.

In Figure 2, we show the simulated results of RANSAC while tracking a planar patch. The patch is moving forward while rotating in the plane. The blue line is the ground truth of in-plane rotation angles, the magenta line is the RANSAC tracking result when sub-pixel feature matching accuracy is unavailable, and the black line is the result when matching outliers are introduced. We also test the RANSAC-PF algorithm (introduced formally in the next section) on the synthesized sequence. In the figure, the red line is the trajectory of the particle with maximal weight at each time point (an approximation to the MAP estimate), and the green line is the mean state value at each time point. In this simple case, drift in the parameter estimation from RANSAC is clearly visible, while there is no apparent drift with RANSAC-PF using only 100 particles.

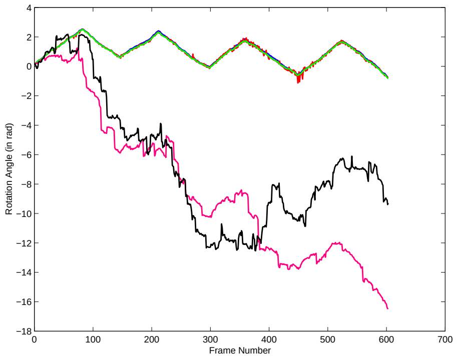


Figure 2: Feature based tracking for a planar patch's cyclic in-plane motion sequence. Blue represents ground truth of rotation angles; black and magenta represent the results of RANSAC with and without outliers, respectively. The results for RANSAC-PF are: red, highest weighted particle, and green, mean state value. In most cases, the latter are closely superimposed with blue.

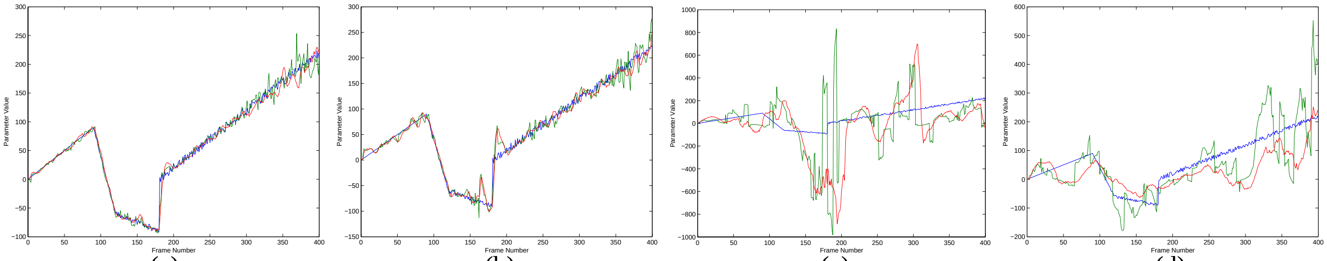
For a second example, we use a particle filter to track a sequence of simulated data. A 2nd order Markov dynamics is adopted and the state is directly observed under Gaussian noise. The observation likelihood function is computed based on the difference with the ground truth state and is also Gaussian. Figure 3 shows the simulation results for a classical particle filter tested on various state dimensionalities. Qualitatively, we see that even when estimating only 2 parameters, a 200-particle filter tends to compute poor solutions after 200 to 300 frames. It is evident the results for 6 DOF tracking are meaningless with 800 particles. In short, the performance of the particle filter degrades dramatically when the number of dimensions of the state space increases. This is consistent with actual practice, where a few thousand particles are used for 2D (four parameter) person tracking applications [7] and also reflect the results of King et al. [12].

3.2 The General Algorithm

We consider the object being tracked as described with known models but unknown parameters (or state) X. Given an observation $Z^{(t)}$ of the object for each image frame t, the objective is to estimate the object state $X^{(t)}$ at every time-step (frame) t. We assume that the underlying observation and dynamic models F and G are known:

$$Z^{(t)} = F(X^{(t)}, \eta) \tag{1}$$
$$X^{(t)} = G(X^{(t-1)}, \zeta) \tag{2}$$

where the noise terms η and ζ have known or assumed distribution. We note that the image likelihood function



(a) (b) (c) (d)

Figure 3: We simulate the tracking accuracy with varying numbers of particles in 2, 4 and 6 state space dimensions. Because data is synthesized, the ground truth is known and used to observe the likelihood via an independent Gaussian process assumption. To illustrate, only the tracking results of parameter 1 (no physical meaning) is shown; similar results are obtained for other parameters. (a) 200 Particles for 2 parameters (b) 200 Particles for 4 parameters (c) 200 Particles for 6 parameters (d) 800 Particles for 6 parameters. Colors code same information as in figure 2.

$L(Z^{(t)}; X^{(t)}) = p(Z^{(t)}|X^{(t)})$ and the state propagation function $p(X^{(t)}|X^{(t-1)})$ can be derived from this stated information.

Let $Z^{(t)}$ be a set of $M^{(t)}$ elements $\{z_i^{(t)}\}$:

$$z_i^{(t)} = f_i(X^{(t)}, \eta),\ i = 1, \ldots, M^{(t)} \tag{3}$$

We assume that $X^{(t)}$ can be computed from Z_p, a subset of all $M^{(t)}$ elements of observation and, optionally, a prior state value $X^{(t-1)}$. Let $\mathcal{R}(Z_p; X^{(t-1)})$ denote this function which is essentially an inverse of (1). A specific instantiation for face tracking is given in section 5.

We introduce a set of particles $\{x_i^{(t)}\}_{i=1}^{N+Q}$ and their relative weights $\{\omega_i^{(t)}\}_{i=1}^{N+Q}$ that are maintained for all timesteps t as a representation of the posterior distribution $p(X^{(t)}|Z^{(t)})$. Naturally, any function e of $X^{(t)}$ can be estimated by

$$e(X^{(t)}) = \sum_{i=1}^{N+Q} \omega_i e(x_i^{(t)}) \tag{4}$$

With these definitions, we can see from Figure 4 that RANSAC-PF operates roughly as follows. For each frame t, particles are generated using $\mathcal{R}(Z_p; X^{(t-1)})$ by randomly selecting Z_p and $X^{(t-1)}$ and computing $X^{(t)}$. Optionally, some other particles are sampled from the dynamic model $p(X^{(t)}|X^{(t-1)})$ using randomly resampled particles of $X^{(t-1)}$. These two sets of particles can be mixed together. Weights $\omega_i^{(t)}$ are then computed using the image likelihood function L as is normally done in importance sampling. Note that our RANSAC-PF algorithm does not necessarily need to be combined with the standard particle filtering, but this combination makes it convenient to compare these two algorithms in the subsequent experiments. A graphical model representation of the algorithm is illustrated in Figure 5.

3.2.1 Conditional independent/ dependent RANSAC sampling

We have also included a boosted version of RANSAC-PF in Figure 4. The simple RANSAC sampling procedure produces current particles in the state space without further evaluating the particles produced in previous samples. In order to increase the sampling efficiency, especially for multi-modal density functions, we have found that a boosting-like strategy improves algorithm performance. To illustrate the idea, consider again the feature based planar tracking problem in Figure 2. Any selection of two pairs of feature correspondences can be used to calculate a solution for the translation and rotation parameters. Assume now that there are three independent motions, with one half of the feature points following one motion and a quarter of feature points following each of the other two motions. Therefore, we can obtain a correctly sampled hypothesis of the motion only when the pair of selected feature correspondences belong to the same motion mode. Otherwise, a faulty particle hypothesis is generated from features of different motions. Thus, the probability that the first motion is sampled is $1/4 = (1/2) \times (1/2)$, the probability of the other two motions being sampled is $1/16 = (1/4) \times (1/4)$, and random faulty hypotheses are generated with the remaining probability of $5/8 = 1 - 1/4 - 1/8$. As a result, a correct state hypothesis has the probability of only $3/8$ of being generated. This is referred to as the sampling efficiency, P_s. In practice, P_s is should be as large as possible.

The boosting-like heuristic in Figure 4 means that the features that are consistent with a current hypothesis have a higher chance to be selected to generate the next hypothesis. In the case when correct and incorrect matches are perfectly distinguished, the corresponding probabilities are $1/2$, $5/9$ or at least $3/8$[1]. As a result, the sampling efficiency of the

[1] Assume that the first mode is first sampled by the particle and all the features associated with this motion are not eligible for the sampling of next particle. Because there are half and half features left for mode 2 and

```
RANSAC Particle Filtering Algorithm
inputs: $\mathcal{Z}^{(t)}$; $\hat{x}^{(1)}$ and $\hat{x}^{(2)}$; $N$; $Q$; BoostFlag;
outputs: $X^{(t)}$

a) From the initial results $\hat{x}^{(1)}, \hat{x}^{(2)}$, construct particles for the first two frames.
    • $x_i^{(1)} = \hat{x}^{(1)},\ i = 1, \ldots, N$
    • $x_i^{(2)} = \mathcal{N}(\hat{x}^{(2)}, \sigma),\ i = 1, \ldots, N$, where $\mathcal{N}$ is a Normal diffusion function.

b) From the previous particle set $\{(x_i^{(t-1)}, \omega_i^{(t-1)})\}_{i=1}^N$ at time $t-1$, construct a new particle set $\{(x_i^{(t)}, \omega_i^{(t)})\}_{i=1}^N$ for time $t$ by
    1. If (BoostFlag = False)
       For $i = 1, \ldots, N$, generate $x_i^{(t)}$ by
        (a) Randomly select $x_i^{(t-1)}$ with probability $\omega_i^{(t-1)}$.
        (b) Uniformly Randomly select a subset $Z_p$ of $Z^{(t)}$ from $M^{(t)}$ features (RANSAC).
        (c) Let $x_i^{(t)} = \mathcal{R}(Z_p; x_i^{(t-1)})$.
    2. If (BoostFlag = True)
       For $i = 1, \ldots, N/2$, generate $x_i^{(t)}$ and $x_{i+N/2}^{(t)}$ by
        (a) Randomly select $x_i^{(t-1)}$ with probability $\{\omega_i^{(t-1)}\}$.
        (b) Uniformly randomly select a subset $Z_p$ of $Z^{(t)}$ from $M^{(t)}$ features (RANSAC).
        (c) Let $x_i^{(t)} = \mathcal{R}(Z_p; x_i^{(t-1)})$.
        (d) For $j = 1, \ldots, M^{(t)}$, weight each feature $j$ as $\nu_j^{(t)'} = 1/p(z_j^{(t)}|x_i^{(t)})$ and $\nu_j^{(t)} = \nu_j^{(t)'}/\sum_{i=1}^N \nu_j^{(t)'}$.
        (e) Randomly select a subset $Z_p$ of $Z^{(t)}$ from $M^{(t)}$ features with probability $\{\nu_j^{(t)}\}$ (Boosted-RANSAC).
        (f) Let $x_{i+N/2}^{(t)} = \mathcal{R}(Z_p; x_i^{(t-1)})$.
    3. $i = N+1, \ldots, N+Q$, generate $x_i^{(t)}$ and by Let $x_i^{(t)} = \mathcal{G}(X^{(t-1)}, \zeta)$.
    4. For $i = 1, \ldots, N+Q$, compute $\omega_i^{(t)'} = L(Z^{(t)}; x_i^{(t)})$ and $\omega_i^{(t)} = \omega_i^{(t)'}/\sum_{i=1}^{N+Q} \omega_i^{(t)'}$.
```

Figure 4: The RANSAC-PF algorithm.

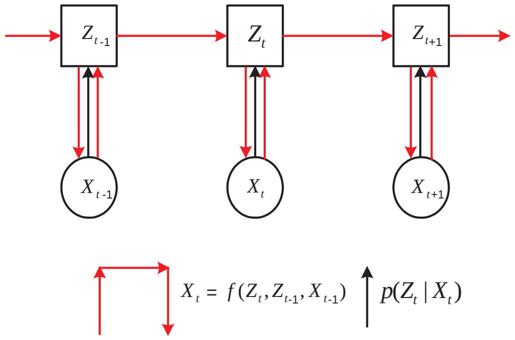


Figure 5: A graphical representation of RANSAC-PF where the new state X_t is a function of new observation Z_t, former observation Z_{t-1} and state X_{t-1}.

next particle is now even higher. In the sequel, the particle sampled from the reweighted image features (Step 2f of Figure 4) is called the dual of the current hypothesis.

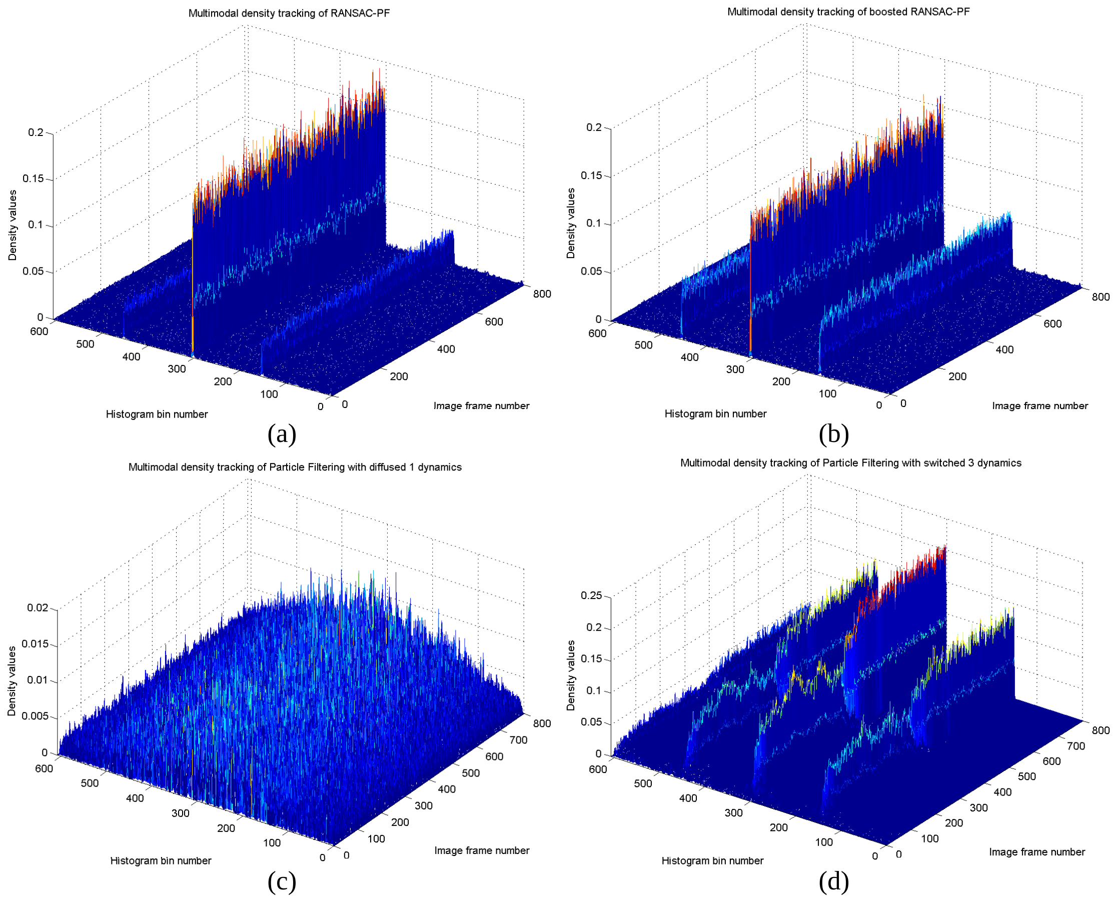
We note that there are several other possible boosting algorithms. For example, in the current algorithm all particles from RANSAC-PF are independently sampled, while the boosted RANSAC-PF particle set consists of half RANSAC produced hypothesis and half their boosted dual particles. An possible alternation is to first sample all RANSAC particles according to their weights (likelihood), then sample their dual particles. For more complex density model, a cascaded boosting scheme may be used [31].

3.2.2 Comparison of multi-modal density tracking

In Figure 6, we show the results of simulating multi-modal density tracking results for with four different algorithms. The underlying motion density function has three modes with $1/2$, $1/4$ and $1/4$ supporting features, respectively. We synthesize a sequence with 800 frames and assume that the positions of feature points are detected without errors. We compare the four trackers' performance on density function approximation accuracy in terms of keeping all the three modes during tracking. For better visualization, we convert the weighted particle representation of the density function into a histogram model using Parzon window integration [5]. From Figure 6 (a) and (b), our proposed RANSAC-PF and boosted RANSAC-PF successfully track the three modes very accurately without any prior knowledge of the underlying density function. Boosted RANSAC-PF further increases the sampling efficiency by having more particle weights concentrated on the three motion modes.

By comparison, a vanilla particle filtering algorithm can theoretically track multi-modal functions, but it normally requires a very large number of particles in the absence of a well tuned dynamical model. In order to generate particles covering the three modes, we use a constant velocity

3, the probability P_s becomes $1/2 = (1/2*1/2+1/2*1/2)$. Similarly, $P_s = (2/3*2/3+1/3*1/3)$ for the cases that mode 2 or 3 is first sampled; $P_s = (1/2*1/2+1/4*1/4+1/4*1/4)$ for the case that no motion mode is first sampled.



(a) (b) (c) (d)

Figure 6: The comparison of multi-modal density tracking on in-plane rotation. There are three modes in the underlying density function. The synthesized sequence contains 800 frames. 600 particles are used for tracking, and the density function is visualized as a histogram representation of 608 bins in each frame. (a) (Conditional independent) RANSAC particle filtering (b) Conditional dependent) boosted RANSAC particle filtering (c) Particle filtering with one diffused dynamics (d) Particle filtering with three switched dynamics.

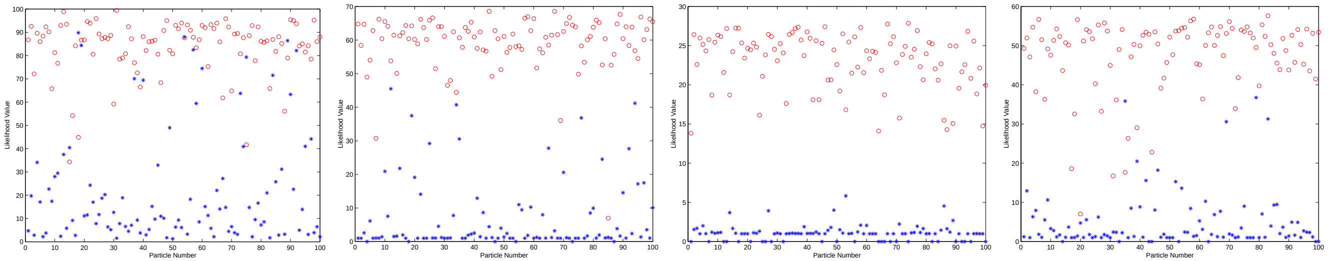
dynamic model with large diffusion. The tracking result in Figure 6 (c) demonstrates its inability to represent the multi-modal density with 600 particles. Assuming we know *a priori* of the three density modes, we can also design a switched model of three dynamics with tight diffusions for each mode [22]. In Figure 6 (d), the density functions converge to have three modes after about 400 frames. However, the results are no better than boosted RANSAC-PF, which has no prior knowledge.

4 Sampling Efficiency and Tracking Evaluation

4.1 Sampling Efficiency

The key of the success of particle filtering is to adaptively concentrate particles in regions of high posterior/likelihood probability by resampling [2], while still guaranteeing fair sampling of the space. In Figure 7, we show the particles' likelihood values (weights in factored sampling [13]) extracted from a face video sequence. For comparison, red circles represent particles driven by RANSAC-PF and blue stars are particles propagated through a second order Markov dynamics. In Figure 7 (a), there are a few high weight blue stars appearing in the cloud of red dots. As time passes, both red and blue particles initially decrease their weights in (b), then stabilize their weights at a reasonable level. (c) depicts a hard-to-track frame with very poor object appearance[2] resulting in even lower weights. However, the clear recovery is found in (d) where the particles' likelihood values return to the same level as (b). The likelihood distribution of blue particles is normally a very few high stars with mostly low ones, while the red particles maintained by RANSAC-PF have an opposite distribution.

[2] The subject's face is turning down deeply, so the face region is small and highly tilted. It causes difficulties for any face tracker.



(a) Frame 3 (b) Frame 105 (c) Frame 360 (d) Frame 425

Figure 7: The likelihood distribution of particles in a video sequence. There are 100 (blue star) dynamics driven particles and 100 (red circle) RANSAC-PF guided particles).)

4.2 An Entropy-Based Criterion for Tracking

While the mean of weighted particles can serve as a representative or estimate of the set of particles, it is the entire set that does the tracking. To evaluate how well it is doing, we introduce an entropy based criterion. The entropy of a probability distribution $p(x)$ is defined as

$$H = -\int_x p(x) \log_2 p(x) \, \mathrm{d}x \tag{5}$$

Since we are tracking only one object configuration, a single sharp peak of the posterior distribution is ideal, while a broad peak probably means poor or lost tracking. Entropy can be used as a scale to discriminate between these two conditions. Low entropy means less tracking uncertainty, thus better performance.

Nevertheless, the weighted particles $\{x_i, \omega_i\}_{i=1}^{N}$ are only a set of samples from a probability distribution $p(x)$, not the distribution itself. There are a number of ways to estimate the entropy of underlying distribution. The simplest method is to compute the entropy directly from weights of discrete samples:

$$H' = -\sum_{i=1}^{N} \omega_i \log_2 p(x_i) = -\sum_{i=1}^{N} \omega_i \log_2 \omega_i \tag{6}$$

H' converges to H when N approaches infinity, but they may have a significant difference when N is small. An alternative in this case is to include a window function to spread the support of a particle like a kernel. We have performed numerical evaluations that suggest there is no significant difference between these two methods with $50 \sim 200$ particles. While the entropy itself is a good indicator, we sometimes need to better discriminate between unimodal and multi-modal distributions. To do so, we artificially merge any pair of particles into one super-particle provided they are near enough in state space. In this way, we further lower the outputs of entropy-estimate functions for single mode distributions; thus promoting them.

5 Experiments on 3-D face tracking

The diagram of our face tracking system is shown in Figure 8 (a). We use a generic triangle based face model, which is highly parameterized and can be easily manipulated with geometric modeling software. Different from [30], the approximate 3D face models are sufficient to achieve the reasonable good tracking results in our experiments.

When initiating tracking, we register the generic 3D model to the first video frame by manually picking 6 fiducial (mouth and eye) corners in the face image. A two-view geometric estimate [6, 33] is then computed for the face pose on the next frame, followed by a Gaussian diffusion. Consequently, $\widehat{x}^{(1)}$ and $\widehat{x}^{(2)}$ are obtained as the state vectors that encode the face pose (three components for rotation and three components for translation). Tracking in subsequent frames proceeds as described below.

5.1 Feature Detection and Random Projection

In our face tracking application, we first detect Harris-like [10] image corner features in two frames. Then, a cross correlation process for feature matching and a rough feature clustering algorithm based on epipolar geometry are performed to form an initial set of corresponding feature pairs. To compute a relative pose change, we spatially and uniformly sample 9 matched image features between two frames by RANSAC, illustrated in Figure 9 (c). We now obtain a set of feature matches $\{(\mathbf{m}_j^{(t-1)}, \mathbf{m}_j^{(t)})\}$, where $\mathbf{m}_j^{(t-1)}$ and $\mathbf{m}_j^{(t)}$ are a pair of (possible non-perfectly) matched points in two successive images. For each point $\mathbf{m}_j^{t-1}$ in the reference image, we cast a 3D ray from the camera center through that point, and compute the intersection $\mathbf{Z}_j$ of that ray with the face mesh model, using a resampled pose state $x_i^{(t-1)}$ at frame $(t-1)$. The relative pose $\hat{\mathbf{T}}_\mathrm{i} = \begin{pmatrix} \hat{\mathbf{R}}_i & \hat{\mathbf{t}}_i \\ \mathbf{0}^T & 1 \end{pmatrix}$ can then be computed according to

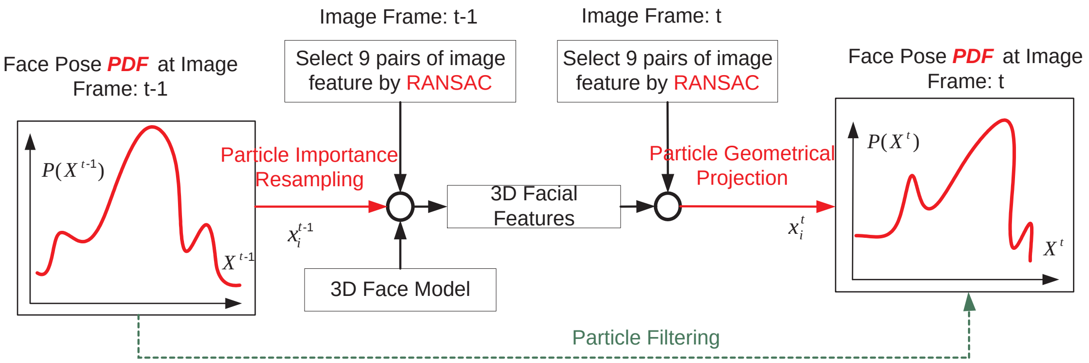


Image Frame: t-1
Image Frame: t
Face Pose PDF at Image Frame: t-1
Select 9 pairs of image feature by RANSAC
Select 9 pairs of image feature by RANSAC
Face Pose PDF at Image Frame: t
$P(X^{t\text{-}1})$
Particle Importance Resampling
3D Facial Features
Particle Geometrical Projection
$P(X^t)$
$X^{t\text{-}1}$
$x_i^{t\text{-}1}$
x_i^t
X^t
3D Face Model
Particle Filtering

Figure 8: Diagram of RANSAC-PF as applied to 3D face pose tracking. (b) The graphical representation of RANSAC-PF where the new state X_t is a function of new observation Z_t, former observation Z_{t-1} and state X_{t-1}.

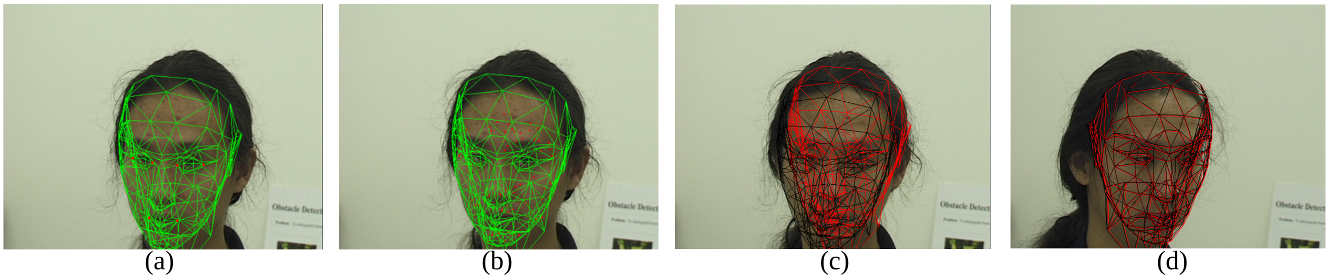


Figure 9: The initialization process of face tracking. (a) The initial frame is manually aligned with a 3D face model using 6 fiducial corners. (b) The next frame is tracked through the two-view motion estimation. The RANSAC-PF tracking begins from the third frame. We show the Maximum A Posterior (*MAP*) result with a red color reprojected face mesh overlaid on the images, while the mean of weighted particles (*MWP*) with a black color. (c)*MAP* and *MWP* are different at the beginning frames of the RANSAC-PF tracking. (d) *MAP* and *MWP* converge together quickly.

the following equation

$$\mathcal{A}\mathcal{P}\hat{\mathbf{T}}_{\mathrm{i}}\tilde{\mathbf{Z}}_j = \lambda\tilde{\mathbf{m}}_j^t \tag{7}$$

where $\tilde{\mathbf{Z}}_j = (\mathbf{Z}_j^T, 1)^T$ and $\tilde{\mathbf{m}}_j = (\mathbf{m}_j^T, 1)^T$. The intrinsic matrix $\mathcal{A}$, the standard projection matrix $\mathcal{P}$, $\mathbf{Z}_j$ and $\mathbf{m}_j^t$ are known. Each of the above equations gives two constraints on $\hat{\mathbf{T}}_{\mathrm{i}}$. We compute $\hat{\mathbf{T}}_{\mathrm{i}}$ with a linear least-squares technique[3] described in [6]. A pair of $(\hat{\mathbf{R}}_i, \hat{\mathbf{t}}_i)$ corresponds to a certain particle as $X_i^{(t)}$. Therefore, this linear geometric projection behaves as a bridge between the propagation of state particles from $X_i^{(t-1)}$ to $X_i^{(t)}, i = 1, \ldots, N_p$ on frame t. We call this process random projection *(RP)*. An illustration of the random projection procedure is demonstrated in Figure 10 (a,b,c).

5.2 Dynamics and Image likelihood

Image observations are modelled as a Gaussian process. With each $x_i^{(t)}$ and its former state history $x_i^{(t-1)}$, we can project the position of image point features at image $(t-1)$ to image (t). The reprojection errors are the 2D Euclidean distances d_m^2 between image features $(u_m^{(t)}, v_m^{(t)})$ at frame t and reprojected image features $(\tilde{u}_m^{(t)}, \tilde{v}_m^{(t)})$.

$$d_m^2 = (u_m^{(t)} - \tilde{u}_m^{(t)})^2 + (v_m^{(t)} - \tilde{v}_m^{(t)})^2 \tag{8}$$

Then the conditional probability for likelihood is

$$p(z|x) \propto \frac{1}{\sqrt{2\pi}\sigma} \sum_m e^{-\frac{d_m^2}{2\sigma^2}} \tag{9}$$

where the standard derivation σ can be estimated from the set of all feature reprojection distances $\{d_m\}$ for each pair

[3] We use 9 as the number of image features for the random projection in our algorithm. In theory, 3 is the minimal possible number to compute the 3D object pose. By considering the sub-pixel matching errors, too few (ie, 3) features can not provide stable geometric estimates normally. On the contrary, too many features lose the advantage of robustness by random sampling. We empirically find 9 is a good number for the trade-off. More theoretical and experimental analysis will be explored for future work.

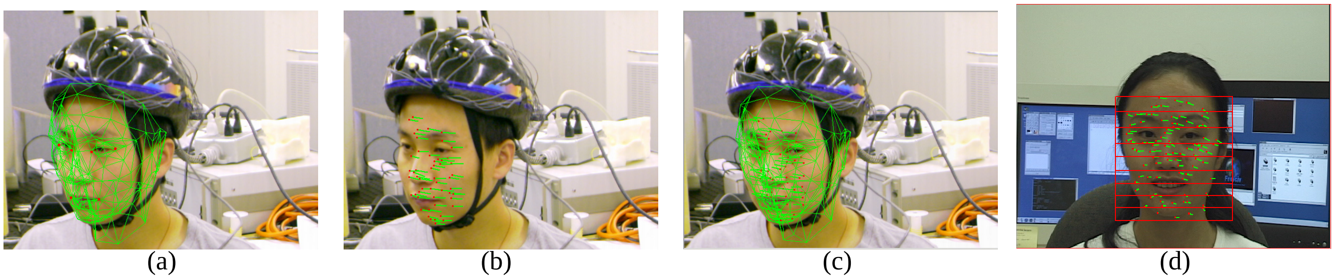


Figure 10: (a) A particle of head pose is overlaid in current frame. (b) Some facial image feature correspondences between the current and next frame are established. (c) The particle is projected into the next frame via random selected image features. (d) Image features are selected spatially and uniformly via RANSAC.

of $x_i^{(t-1)}$ and $x_i^{(t)}$. In the experiments, we set σ to 2.5 pixels for simplicity. No apparent improvement was found when estimating σ from data.

5.3 Tracking Results

We use a simple constant velocity model to guide the temporal evolution of particle filtering.

In Figure 9, we show a short face tracking video *(Comparison.avi)*[4] with large out-of-plane rotations. In this case, a subject's face is considered as a rigid object without facial expressions. From this figure, reasonable tracking accuracy[5] is achieved, although the generic face model is not very accurate for the given subject and feature mismatching does exist. After few frames, the estimates of *MAP* and *MWP* estimates converge together.

For the convenience of comparison, we generalize our RANSAC-PF algorithm with particles guiding by a second order Markov (constant velocity) dynamics in parallel (see the dashed line in Figure 8). The results on the above short sequence *(Comparison.avi)* is shown in Figure 11. We name the particles driven by RANSAC-PF *RP (random projection)* particles, and the others as *DP (dynamic propagation)* particles. Note that the constant velocity dynamics can be considered as a reasonable assumption for this simple yawing video sequence. Nevertheless, the tracking in Figure 11 (c) is quickly lost due to the relatively small number of particles according to the 6-DOFs required by 3D tasks. On the other hand, our algorithm performs better with the same or smaller number of total particles. From Figure 11 (a) and *(Comparison.avi)*, good tracking results are obtained with 100 *RP* particles and 100 *DP* ones. When reducing the *RP* particles to 10 in Figure 11 (d), slight tracking accuracy is lost for *MWP* and the *MAP* results become to flicker around *MWP* estimates. It means that the computed *MWP* is stable and *MWP* is not. Here 10 can be thought of as a lower bound for the number of *RP* particles. The decrease of *DP* particles (comparing (b) to (a) in figure 11) does not apparently influence tracking quality.

We also tested our algorithms on tracking people faces from different races. Two video sequences *(cher.avi, donald.avi)* are linked in author's website. *(Cher.avi)* has moderate expression changes and results in better tracking, compared to *(donald.avi)*, where intensive expressional deformations occur. Both of the videos (tracked with 80 particles) have long rotation ranges over $20 \sim 30$ seconds, and subjects move their face arbitrarily. Automatic recoveries from poorly tracked frames can also be found. To test the robustness to misalignments, we manually align the first frame in the tracking sequence with some moderate errors. Our algorithm shows the remarkable stability from Figure 12. The initial registration errors do not increase with time, and a significant accumulation of tracking errors is not observed. A general 3D face model is used for tracking though particular adjustments of face model to a subject may improve the tracking.

For quantitative evaluation of the tracking results, we capture 2 video sequences of a subject moving his head with motion capture device. Six optical markers in the subject's helmet are tracked during head moving (for instance, in Figure. 10 (a,b,c)), and the subject's head poses are then computed as the ground truth. As shown in figure 13, our method can keep track of the subject's head poses over the large ranges. The average tracking errors of yaw, pitch and roll are 4.6790^o, 3.4715^o and 4.3466^o in the first video sequence (rigid face motions only) and 6.8714^o, 3.6158^o and 5.7456^oin the second video sequence (moderate facial deformations).

The relative motion between successive frames is not required to be very smooth in our experiments. We have concluded that random projection is most successful when handling rotations of 0^o to 5^o degrees. One way to test this robustness is to simply leave frames out. In experiments, images used for tracking can be sub-sampled every 3 to 10

[4] All the video can be accessed from http://www.cs.jhu.edu/ lelu/RansacPF.htm

[5] Since we do not have the ground truth for tracking, no explicit numerical comparison is provided. The validity is shown by overlaying the 3D face mesh model to images.

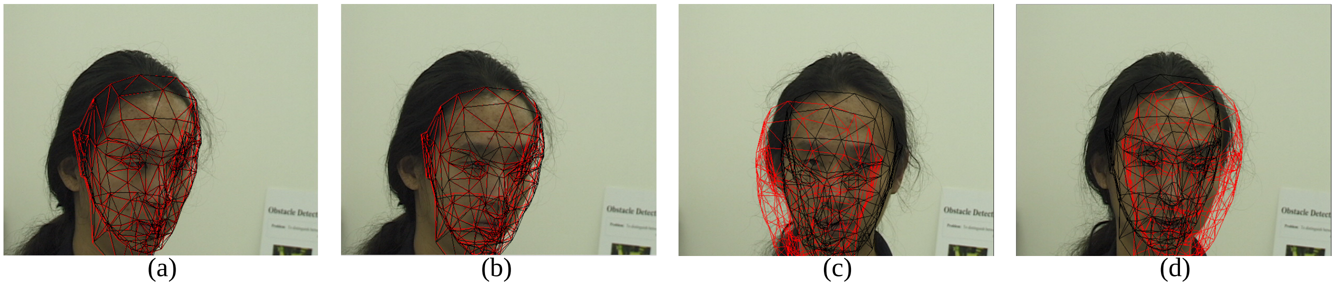


Figure 11: The tracking result comparison of RANSAC-PF under different configurations. (a) 100 *RP* particles and 100 *DP* particles (b) 100 *RP* particles and 10 *DP* particles (c) 200 *DP* particles (d) 10 *RP* particles and 100 *DP* particles

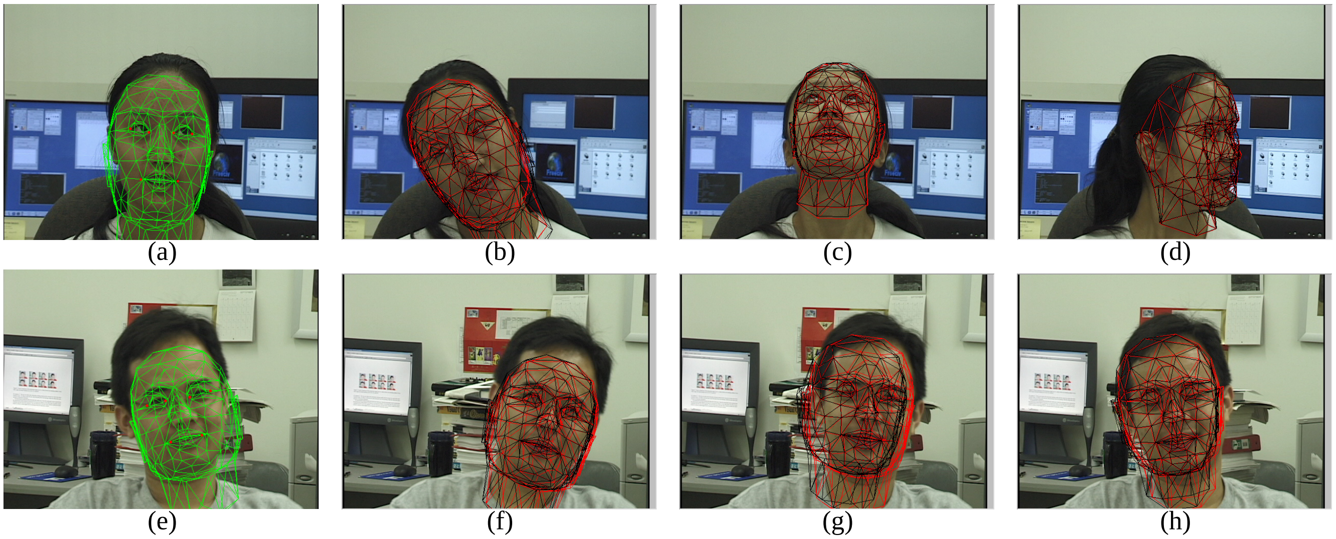


Figure 12: Robustness testing for two misaligned video sequences. (a) The initial frame with the visible alignment error of subject 1 (b) Frame 195 (c) Frame 628 (d) Frame 948 (e) The initial frame with the visible alignment error of subject 2 (f) Frame 185 (g) Frame 255 (h) Frame 285.

frames and we do not observe the evident degrading tracking results.

In our experiments, the entropy curve is very stable most of the time as expected, indicating the stable tracking performance. For the extreme cases (Figure 7 (c)), the entropy value does increase. In Figure 14, we show three entropy curves computed from equation 6 based on the discrete distribution representation (with a variation of merging the close particles into a super ”particle” also) and the continuous distribution approximation by Parzon window [5]. The testing images are from *cher.avi*.

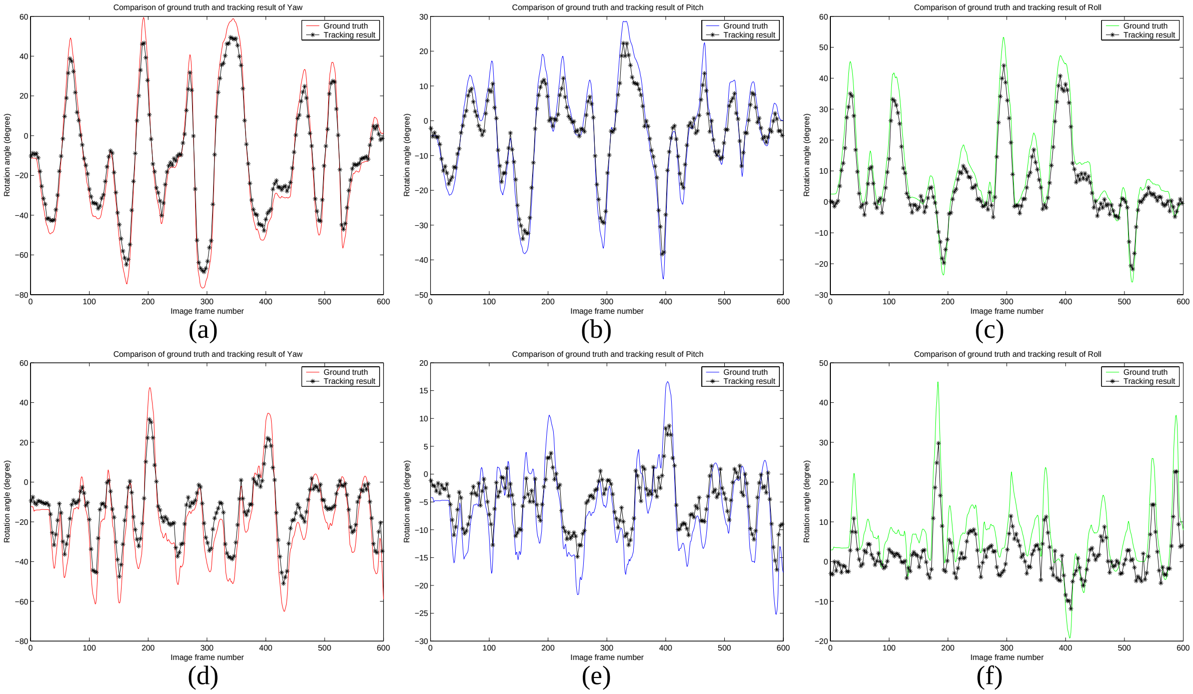
6 Summary and Future Work

In this paper, we have presented a stochastic technique for full 3D face tracking with a small number of particles and a weak dynamical model. The main feature of this algorithm is a RANSAC-based image feature selection which greatly improves the efficiency of the samples used by the algorithm.

This algorithm presents several directions for further research. Most importantly, we hope to offer a convergence proof for this algorithm along the lines suggested in [4, 2]. In particular, we hope to show that some form of boosted RANSAC is formally convergent. The latter shows great promise at alleviating some of the problems identified in [12] when tracking multi-modal densities. Further improving the boosting method is another direction that offers great promise.

We also intend to improve and extend our work to multi-face tracking. Our local feature matching algorithm is expected to distinguish features from different faces by appearance and spatial neighborhood constraints. This step can help RANSAC generate proposals from each person’s matched feature set respectively, while boosting will ensure the multiple modes are maintained.

Finally, finding suitable methods to compute importance



(a) (b) (c)

(d) (e) (f)

Figure 13: Comparison of the tracked rotations and the ground truth. The first row is a video sequence containing the rigid face motions only; the second row is a video sequence containing some moderate facial deformations. (a) Yaw (b) Pitch (c) Roll (d) Yaw (e) Pitch 2 (f) Roll.

proposal probabilities for Monte Carlo-style algorithms is another area of future work.

References

[1] J. Xiao, S. Baker, I. Matthews, and T. Kanade, Real-Time Combined 2D+3D Active Appearance Models, *IEEE Conf. on Computer Vision and Pattern Recognition*, June, 2004.

[2] D. Crisan and A. Doucet, A Survey of Convergence Results on Particle Filtering for Practitioners, *IEEE Trans. Signal Processing*, vol. 50, no. 3, pp. 736-746, 2002.

[3] J. Deutscher, A. Blake and I. Reid, Articulated Body Motion Capture by Annealed Particle Filtering. CVPR'00.

[4] D. Crisan and A. Doucet, Convergence of Sequential Monte Carlo Methods. CUED/F-INFENG/TR381, 2000.

[5] R. Duda, P. Hart and D. Stork, *Pattern Classification (2nd)*, Wiley Interscience, 2002.

[6] O. Faugeras, *Three-Dimensional Computer Vision: a Geometric Viewpoint*, MIT Press, 1993.

[7] D. Fox, KLD-Sampling: Adaptive Particle Filters. *NIPS*'01.

[8] M.A. Fischler and R.C. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography. *Commun. Assoc. Comp. Mach.*, vol. 24:381-95, 1981.

[9] G. Hager and P. Belhumeur, Efficient Region Tracking With Parametric Models of Geometry and Illumination. *IEEE Trans. PAMI*, **20:10**, 1998.

[10] C. Harris and M. Stephens, A combined corner and edge detector. *4th Alvey Vision Conf.*, pp. 189-192, 1988.

[11] T. Hastie and R. Tibshirani, Discriminant Analysis by Gaussian Mixtures. *Journal of Royal Statistical Society Series B*, 58(1):155-176.

[12] O. King and D. Forsyth, How does CONDENSATION behave with a finite number of samples? ECCV'00, pp. 695-709.

[13] M. Isard and A. Blake, CONDENSATION – conditional density propagation for visual tracking. *IJCV* **29:1**, pp. 5-28, 1998.

[14] M. Isard and A. Blake, ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework. ECCV'98, pp. 893-908.

[15] M. La Cascia, S. Sclaroff, and V. Athitsos, Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Robust Registration of Texture-Mapped 3D Models. *IEEE Trans. PAMI*, **22:4**, April, 2000.

[16] J.S. Liu and R. Chen. Sequential Monte Carlo for Dynamic System. *Journal of the American Statistical Association* **93**, pp. 1031-1041 .

[17] J.S. Liu, R. Chen, and T. Logvinenko. A Theoretical Framework for Sequential Importance Sampling and Resampling.

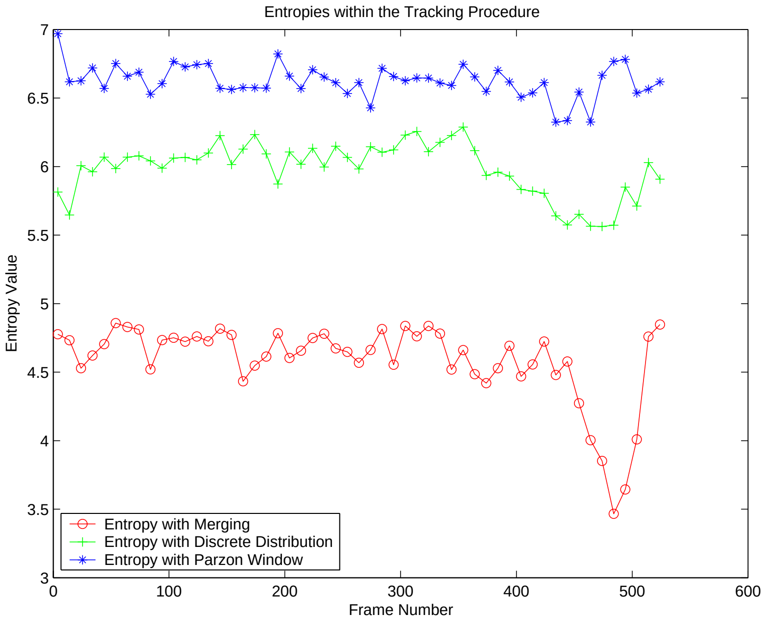


Figure 14: The entropy curves based on merging, discrete distribution and Parzon windows [5] during tracking.

Sequential Monte Carlo in Practice, A. Doucet, N. deFreitas, and N. Gordon, Eds., New York: Springer-Verlag, 2000.

[18] H. Moon, R. Chellappa, and A. Rosenfeld, 3D Object Tracking using Shape-Encoded Particle Propagation. ICCV'01.

[19] L. Morency, A. Rahimi and T. Darrell, Adaptive View-Based Appearance Models. CVPR'03.

[20] B. North, A. Blake, M. Isard, and J. Rittscher, Learning and classification of complex dynamics. *IEEE Trans. PAMI*, **22:9**, pp. 1016-1034, Sep., 2000.

[21] K. Okuma, A. Taleghani, N. De Freitas, J. Little, D. Lowe, A boosted particle filter: multitarget detection and tracking, *European Conference on Computer Vision*, May 2004.

[22] V. Pavlovic, J. Rehg and J. MacCormick, Title Learning Switching Linear Models of Human Motion, *In Neural Information Processing Systems*, 2000.

[23] V. Pavlovic04, Model-based motion clustering using boosted mixture modeling, CVPR, 2004.

[24] J. Sullivan and J. Rittscher, Guiding random particles by deterministic search. ICCV, I:323-330, 2001.

[25] P.H. Torr and A. Zisserman, MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. *CVIU*, **78**:138-156, 2000.

[26] P.H. Torr and C. Davidson, IMPSAC: Synthesis of Importance Sampling and Random Sample Consensus. *IEEE Trans. PAMI*, **25:3**, March, 2003.

[27] K. Toyama and G. Hager, Incremental focus of attention for robust vision-based tracking, *International Journal of Computer Vision*, **35(1)**:45-63, Nov. 1999.

[28] K. Toyama and A. Blake, Probabilistic Tracking in a Metrix Space. ICCV'01.

[29] Z. Tu and S.C. Zhu, Image Segmentation by Data-Driven Markov Chain Monte Carlo. *IEEE Trans. PAMI*, **24:5**, May 2002.

[30] L. Vacchetti, V. Lepetit, P. Fua, Fusing Online and Offline Information for Stable 3D Tracking in Real-Time. CVPR'03.

[31] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[32] J. Xiao, T. Kanade, and J. Cohn, Robust Full-Motion Recovery of Head by Dynamic Templates and Re-registration Techniques. FG'02.

[33] Z. Zhang, et al., A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry, *Artificial Intelligence J.*, **78** pp: 87-119, Oct. 1995.

* Abstract

Particle filtering is a very popular technique for sequential state estimation. However, in high-dimensional cases where the state dynamics are complex or poorly modeled, thousands of particles are usually required for real applications. This paper presents a hybrid sampling solution that combines RANSAC and particle filtering. In this approach, RANSAC provides proposal particles that, with high probability, represent the observation likelihood. Both conditionally independent RANSAC sampling and boosting-like conditionally dependent RANSAC sampling are explored. We show that the use of RANSAC-guided sampling reduces the necessary number of particles to dozens for a full 3D tracking problem. This is method is particularly advantageous when state dynamics are poorly modeled. We show empirically that the sampling efficiency (in terms of likelihood) is much higher with the use of RANSAC. The algorithm has been applied to the problem of 3D face pose tracking with changing expression. We demonstrate the validity of our approach with several video sequences acquired in an unstructured environment.