A Particle Filter without Dynamics for Robust 3D Face Tracking

Le Lu Xiang-Tian 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 problem. However its convergence greatly depends on the balance between the number of particles/hypotheses and the fitness of the dynamic model. In particular, in cases where the dynamics are complex or poorly modeled, thousands of particles are usually required for real applications. This paper presents a hybrid sampling solution that combines the sampling in the image feature space and in the state space via RANSAC and particle filtering, respectively. We show that the number of particles can be reduced to dozens for a full 3D tracking problem which contains considerable noise of different types. For unexpected motions, a specific set of dynamics may not exist, but it is avoided in our algorithm. The theoretical convergence proof [1, 3] for particle filtering when integrating RANSAC is difficult, but we address this problem by analyzing the likelihood distribution of particles from a real tracking example. The sampling efficiency (on the more likely areas) is much higher by the use of RANSAC. We also discuss the tracking quality measurement in the sense of entropy or statistical testing. The algorithm has been applied to the problem of 3D face pose tracking with changing moderate or intense expressions. 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), to computer vision problems. Applications on parameterized or non-parameterized contour tracking [11, 12, 20], and human tracking [2, 14] 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 are generally determined by the dimension and structure of the state space [5]. A typical 6-DOF tracking problem usually requires thousands of particles [5]; reducing the number of particles by training a finely tuned dynamic model is not trivial [16], sometimes even impossible.

On the other hand, the RANSAC method [6] is often applied as a robust estimation technique. The final stage in RANSAC is to apply a robust estimator that results in a good solution which includes as many non-outliers as possible. However, 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 the 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 convergence property of RANSAC-PF is empirically analyzed.

The evaluation of 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 the statistical quality characteristics of the tracked density function and evaluate it numerically. More importantly, the entropy curve computed during tracking can help us extract some well tracked frames as exemplars [20]. When necessary, these exemplars are then archived to stabilize the tracking[1].

The remainder of this paper is organized as follows. Related work is presented in section 2, followed by a description of a 3D face tracking application that uses our RANSAC-PF algorithm. We also address our entropy and statistical testing based criterion for tracking quality evaluation in this section. Section 4 shows some experimental results. Finally, we offer conclusions and discuss future work.

Notation: $x_i^{(t)}$ is the current state of the *i*-th particle at

[1] Much of computer vision research can be regarded as lying on a continuum between explicit models and exemplars [20]. To extract exemplars which are tracked statistically well, we discuss the tracking quality evaluation issue from the tracked posterior probabilistic distributions.

time t, while $X_i^{(t)}$ represents the current and previous states of the i-th particle at time t. $X_i^{(t)} = \{x_i^{(1)}, \ldots, x_i^{(t)}\}$. Assuming a second order Markov property, $X_i^{(t)}$ can also be represented as $\{x_i^{(t-1)}, x_i^{(t)}\}$, or $\{x_i^{(t)}, v_i^{(t)}\}$, where $v_i^{(t)} = x_i^{(t)} - x_i^{(t-1)}$.

$\omega_i^{(t)}$ is the normalized weight for the i-th particle at time t. These weights collectively represent the particles' posterior probability distribution.

$z^{(t)}$ delegates the current observation (image features, for example). $z^{(t)} = \{z_1^{(t)}, \ldots, z_{M^{(t)}}^{(t)}\}$, where $M^{(t)}$ is the number of detected features at time t.

2 Related Work

Deterministic parameter estimation algorithms 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 [23] and MLESAC [18], follow the strategy of "Winner Takes All" and get the maximum likelihood (ML) result. However, those estimators are not suitable for a sequential estimation problem for a dynamic system because the ML estimation error in each stage can accumulate and result in failure.

It is shown in [10] 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 PF-like techniques are accurate in the 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, random geometric projections with the RANSAC sampling of image features and importance resampling of X^{t-1} guide the time series evolution of state particles to achieve the trade-off between variance and bias.

There are some papers integrating particle filters with variational optimization or observation-based importance function. For the sake of computational efficiency, Sullivan et al. showed in [17] 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 the probabilistic or deterministic tracking engines, which could be problematic. Isard et al. [12] presented an approach (ICondesation) to combine low-level and high-level information by importance resampling with a particle filter. Our most related work is Torr and Davidson's research [19] on 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 with significant observation noise[2]. 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.

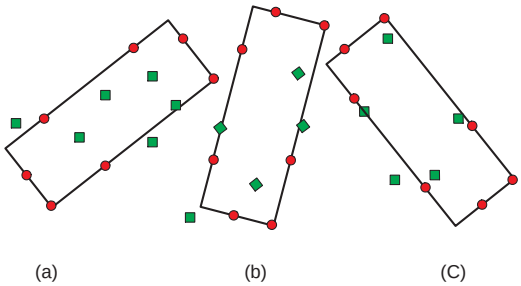


Figure 1: *An example of RANSAC sampling of feature points (red dots are inliers, and green squares are outliers.) to track planar motions.*

There are various solutions for face pose tracking. Here we mainly discuss two techniques: SSD (sum of squared distances) and particle filtering based tracking. SSD [7, 13] has attracted much interest and has become a standard technique for various tracking problems. However, a single template can not cover a large range of motions. Dynamic template updating is usually a non-trivial problem and causes error accumulation. Moreover, the relative motion between frames is required to be smooth enough to satisfy the local linearity prerequisite of Jacobian approximations in SSD tracking. Jumps can cause loss of tracking. Moon et al. [14] proposed a new approach for the image likelihood measurement based on the shape distance between the reprojected eye and mouth curve with the detected curve in the image. This observation model is simple and efficient to be implemented, but may have some problems with large out-of-plane rotations and expressional changes.

We propose a new method to achieve comparable tracking results [13] under many strong distractions, with much fewer ($50 \sim 200$) particles, auto-recoverable for a wide-range full 3D tracking problem[3]. Small jumps (up to 10^o) between consecutive frames is also not an issue for our approach.

3 The RANSAC-PF Algorithm

3.1 Motivation

In order to explain our approach to tracking, consider the situation shown in Figure 1. Here we consider in-plane rotations and translations of a planar object through three frames

[2]We consider three sources of noise in the probabilistic robust tracking work; 1) image feature matching outliers mainly due to expression deformations; 2) the inaccuracy of our 3D face model; 3) manual alignment errors for initialization. The hybrid sampling strategy is used to handle the biased or unbiased noises, playing the similar role as a robust estimator.

[3]Face pose tracking is a well-studied topic, but still remains difficult for a deformable subject with non-semantic, wide-range motions. Here we attempt to get 3D pose estimates from a moving talking face. Expressional deformations are treated as clutter noise. Please refer to http://www.cs.jhu.edu/ lelu/RansacPF.htm for more results.

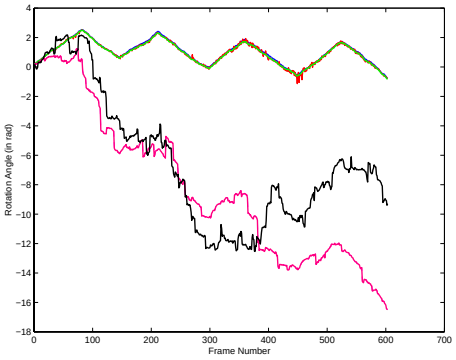


Figure 2: *Feature based tracking for a planar patch's cyclic in-plane motion sequence. Blue represents ground truth of rotation angles; black and pink represent the results of RANSAC with and without outliers; red (MAP) and green (MWP) (closely superimposed with blue) are results for RANSAC-PF .*

of a video. 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. We develop a RANSAC filter for computing *incremental* motion between successive frames (to simplify features matching), 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, note that RANSAC tries a number of subsets of features; while the probability that at least one of the subsets yields an adequately correct result can be set to a high probability[4] q by adjusting the tolerance threshold, the overall probability of 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 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 in the synthesized sequence: the red line is the trajectory of particles with maximal weight (*MAP*) and the green line is the mean trajectory (*MWP*). In this simple case, drift in the parameter estimation from RANSAC is clearly visible, while there is no apparent drift with RANSAC-PF of 100 particles.

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 observation likelihood function is based on the difference with the ground truth[5]. We see that the performance of the particle filter degrades dramatically when the number of dimensions of the state space increases. Figure 3 shows the simulation results for a particle filter tested on various dimensions. 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. This is not inconsistent with actual practice, where a few thousand particles are used for 2D (four parameter) person tracking applications [5]. King et al. [10] also addressed the convergence problem of particle filter with finite samples.

[4] $q = 1$ only happens without existence of any kind of noise.

[5] This observation measurement is near perfect, because the particle weights are penalized respect to their bias with the ground truth.

3.2 The General Algorithm

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 Gaussian 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

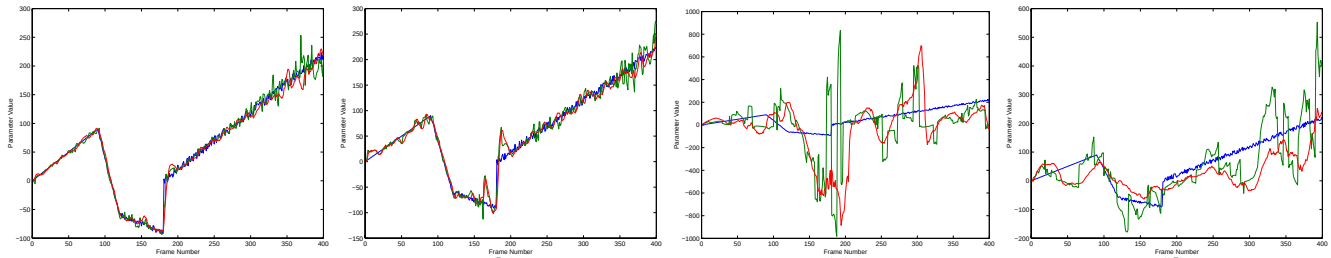
1. For $i = 1, \ldots, N$, generate $x_i^{(t)}$ by
   (a) Randomly select $x_i^{(t-1)}$ with probability $\omega_i^{(t-1)}$
   (b) Randomly select a subset Z_p of $M^{(t)}$ features from $z^{(t)}$ by RANSAC.
   (c) Let $x_i^{(t)} = R(Z_p; x_i^{(t-1)})$.
2. For $i = 1, \ldots, N$, compute $\omega_i^{(t)'} = L(z^{(t)}; x_i^{(t)})$ and $\omega_i^{(t)} = \omega_i^{(t)'} / \sum_{i=1}^N \omega_i^{(t)'}$.
3. Compute the empirical entropy criterion $H^{(t)} = -\sum_{i=0}^N \omega_i^{(t)} \log_2 \omega_i^{(t)}$, or perform other statistical testings.

Figure 4: *The RANSAC-PF algorithm*

We consider the object being tracked as described with known models but unknown parameters (or states) 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) from $Z^{(t)} = \{z^{(1)}, \ldots, z^{(t)}\}$. Assume that the underlying observation and dynamic models F and G are known:

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

where the noise terms η and ζ have known or assumed distribution. We note that the image likelihood function $L(z^{(t)}; X^{(t)}) = p(z^{(t)}|X^{(t)})$ and the state propagation function $p(x^{(t)}|X^{(t-1)})$ can be derived from this stated information.



(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.*

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. Let $R(Z_p; X^{(t-1)})$ represent this inverse of the above equation .

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

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

With these definitions, we can see from Figure 4 that RANSAC-PF operates roughly as follows. For each frame t, particles are generated using R by randomly selecting Z_p and $X^{(t-1)}$ and computing $X^{(t)}$. A graphical model representation of the algorithm is illustrated in Figure 6 (b).

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. An entropy criterion or other testing to evaluate the tracking result from the particles is also computed. 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 following experiment section.

3.3 Convergence Analysis

There are two important assumptions for the convergence analysis of sequential importance sampling, or particle filtering-style algorithms [3, 1]. First, the importance function is chosen so that the weights of particles are bounded above. No super-nodes with unbounded high weights exist that may dominate the distribution and resampling. Second, selection scheme does not introduce too strong a discrepancy.

The first prerequisite is easily satisfied by the independent Gaussian process assumption for likelihood measurement (i.e. computing the particle weights in factor sampling [11]). Each process returns a real value between 0 and 1, and the number of processes is limited. On the other hand, the computational efficiency and convergence of particle filter algorithms heavily depend on the selection (resampling) scheme. The ideal case will be that the limited number of particles are sampled from the high peak areas in the posterior density resulting in an estimated expectation[6] with a tight variance. The limited computational resource is not wasted in the unlikely zones. We also require having low bias to obtain good tracking results, so randomness is introduced around high density peaks. The trade-off between variance and bias must be subtly handled.

From Figure 5, we show the particles' likelihood values (weights in factor sampling [11]) 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 5 (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[7] 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. In our experiments, multi-modal estimates of particles are converged to have a dominant result with minor (low-weighted) noisy estimates most of the time. When distractions are very strong, we obtain many estimates floating around the true value, and no dominant estimate exists.

Our intention is not to give a theoretical convergence proof, but attempt to illustrate the basic concept of how RANSAC-PF works by an example. The possibility of poor sampling is lower for a distribution with more high like-

[6] The expectation is also a random variable.

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

lihood particles [3, 1]. Most of the time, the resampling process on very low weight particles generates a poor result. Similarly, 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 [21].

3.4 Tracking Evaluation

Once the tracking results are obtained, our next step is evaluating the soundness of the results. By looking into the likelihood function $L(z^{(t)}; x_i^{(t)})$, we can calculate a fitness value for each particle $x_i^{(t)}$. An intuitive argument can be whether there exists at least one particle $x_i^{(t)}$ with the fitness $L(z^{(t)}; x_i^{(t)})$ above a certain threshold. However, we prefer to find a more sound answer if possible, using the property of the obtained posterior density $\{x_i^{(t)}, \omega_i^{(t)}\}$ of the state variables.

3.4.1 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 can introduce an entropy based criterion. Since we are tracking only one object configuration, a single and 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 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:

$$H_1 = -\sum_{i=1}^{N} \omega_i \log_2 p(x_i) = -\sum_{i=1}^{N} \omega_i \log_2 \omega_i \qquad (5)$$

H_1 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 multimodal 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.

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 5 (c)), the entropy value does increase.

3.4.2 Statistical Testing for Mode Detection

Though particle filtering is well known to tackle the non-Gaussian tracking problem, we may still want to test whether a single Gaussian is a valid posterior assumption for a certain frame. For a single object tracking problem, the current frame computing result can be considered favorable if the tracking density can be well approximated by a single Gaussian with tight variance. Furthermore, well tracked frames can be used as exemplars to build an adaptive multiview object model [15] for many purposes.

We address the problem as verifying whether a Gaussian mixture model (GMM) can achieve a statistically superior approximation for tracking posterior density, compared with a single Gaussian model. Since we are only concerned with whether the distribution is unimodal or multi-modal, the testing on 1 or 2-component GMM is adequate. This model selection problem can be solved by evaluating the conditional mixture density (log-)likelihoods of 1 or 2-component GMM via Expectation Maximization [9][8].

Alternatively, the modality of the tracking density may be observed directly by detecting whether the particle of maximum a posterior converging to the estimated posterior expectation, as an empirical measure. It is clearly shown from our videos that this convergence (unimodal) is mostly kept in our real experiments, except for the poorly tracked frames.

4 Experiments on 3-D face tracking

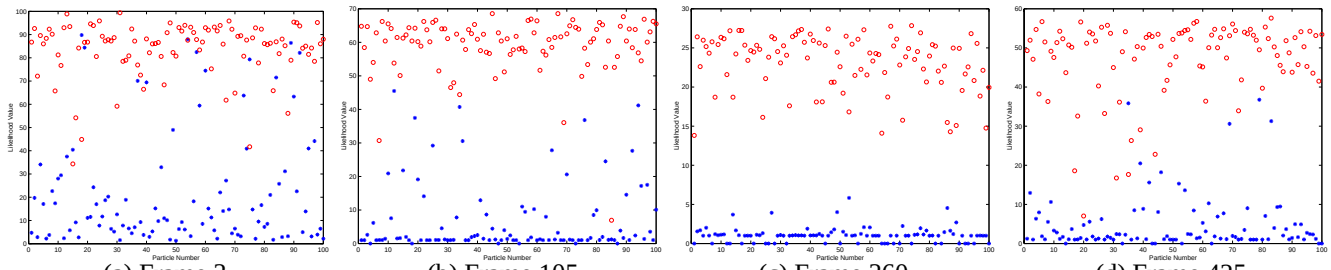
The diagram of our face tracking system is shown in Figure 6 (a). We use a generic triangle based face model, which is highly parameterized and can be easily manipulated with geometric modeling software. Different from [22], the approximate 3D face models are sufficient to achieve the reasonable good tracking results in our experiments, thanks to the stochastic property of RANSAC-PF.

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 [4, 23] 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).

4.1 Feature Detection and Random Projection

In our face tracking application, we first detect Harris-like [8] 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

[8] The difference is that particles are data samples with different weights, making non-equal contributions for the mixture density modelling.



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

Figure 5: *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).)*

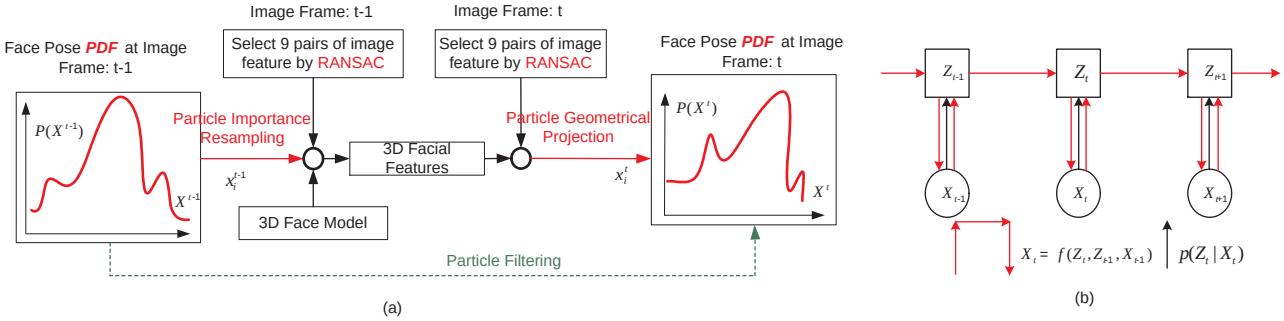
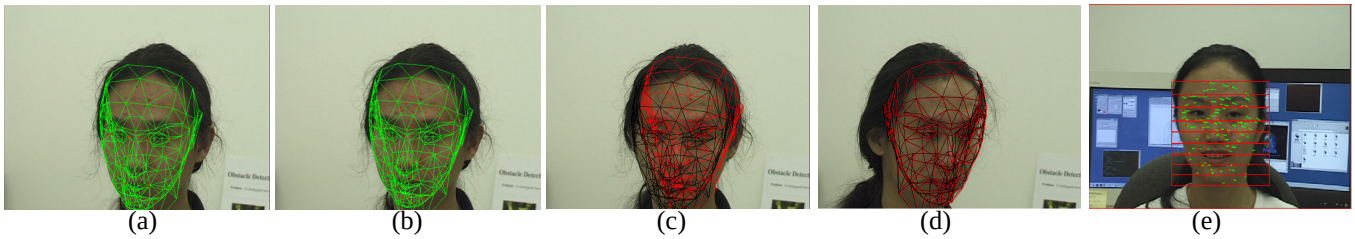


Figure 6: *(a) Diagram of RANSAC-PF as applied to 3D face pose tracking. (b) The graphical representation of RANSAC-PF where the new state X_t is a function of new observation Z_t, former observation Z_{t-1} and state X_{t-1}.*



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

Figure 7: *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. (e) Image features are selected spatially and uniformly via RNASAC.*

frames by RANSAC, illustrated in Figure 7 (c). We now obtain a set of rigid 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 (probably 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 (factor sampling) 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 the following equation

$$\mathcal{A}\mathcal{P}\hat{\mathbf{T}}_{\mathrm{i}}\tilde{\mathbf{Z}}_j = \lambda\tilde{\mathbf{m}}_j^t \qquad (6)$$

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[9] described in [4]. 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)*.

4.2 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 $(t-1)$

[9] 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.

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 $(\tilde{u}_m^{(t)}, \tilde{v}_m^{(t)})$.

$$d_m^2 = (u_m^{(t)} - \tilde{u}_m^{(t)})^2 + (v_m^{(t)} - \tilde{v}_m^{(t)})^2 \quad (7)$$

Then the conditional probability for likelihood is

$$p(z|x) \propto \frac{1}{\sqrt{2\pi}\sigma} \sum_m e^{-\frac{d_m^2}{2\sigma^2}} \quad (8)$$

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

4.3 Results of Our Algorithm

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

In Figure 7, we show a short face tracking video *(Comparison.avi)* 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[10] is achieved, though 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 6). The testing results on the above short sequence *(Comparison.avi)* is shown in Figure 8. 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 8 (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 8 (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 8 (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 8) does not apparently influence tracking quality.

We also test our algorithms on tracking people faces from different races. Reasonably good tracking performance is achieved. 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 9. 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.

In our experiments, the relative motion between successive frames is not required to be very smooth. 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 frames.

5 Summary and Future Work

In this paper, we have presented a stochastic method for full 3D face tracking with a small number of particles and no learned dynamics. RANSAC-based image feature selection is integrated within the Monte Carlo sampling framework. The convergence issue and tracking quality evaluation problem are also discussed.

Our RANSAC-PF algorithm does not depend on the existence of a specific, fine-tuned dynamics for the diverse object moving sequences. Furthermore, our algorithm can help the labelling problem for the new tracking data, which can be valuable for dynamics learning and motion recognition.

We also intend to 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. Data association problem will be much easier.

Finally, finding suitable methods to compute importance proposal probabilities for Monte Carlo-style algorithms is our emphasis on future work.

References

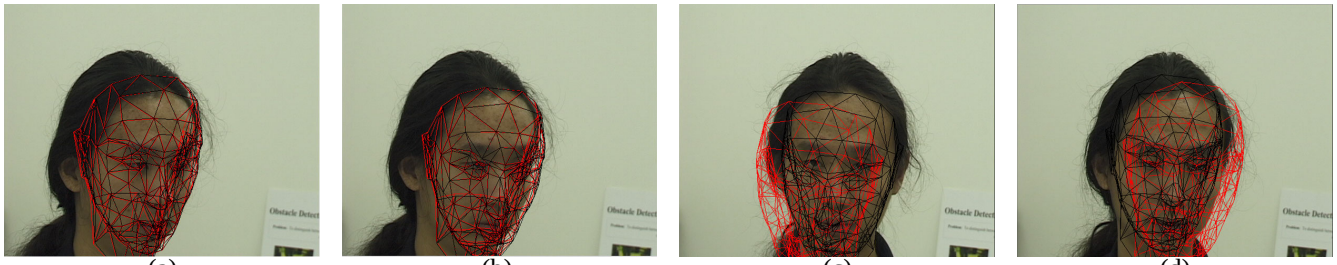
[1] D. Crisan and A. Doucet, A Survey of Convergence Results on Particle Filtering for Practitioners, *IEEE Trans. Signal Processing*, vol. 50, no. 3, pp. 736-746, 2002

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

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

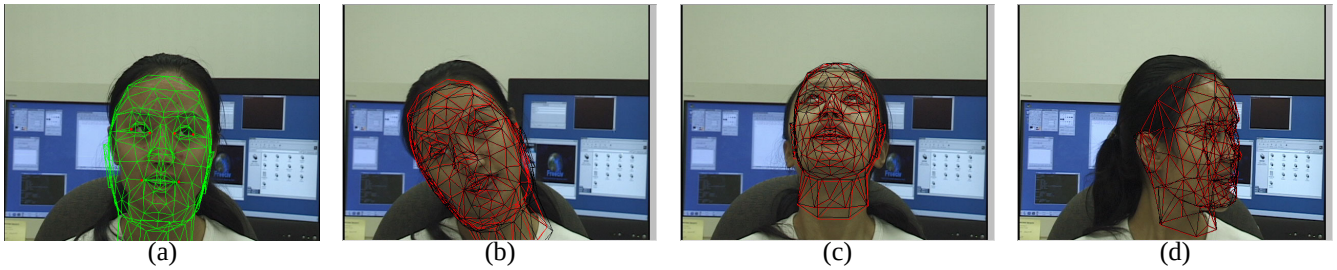
[4] O. Faugeras, *Three-Dimensional Computer Vision: a Geometric Viewpoint*, MIT Press, 1993.

[10] 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.



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

Figure 8: *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*



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

Figure 9: *Robustness testing for a misaligned video sequence. (a) The initial frame with the visible alignment error (b) Frame 195 (c) Frame 628 (d) Frame 948*

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

[6] 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.

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

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

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

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

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

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

[13] 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.

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

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

[16] 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.

[17] J. Sullivan and J. Rittscher, Guiding Random Particles by Deterministic Search. ICCV'01, Vol.I, pp. 323-330.

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

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

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

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

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

[23] 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.