# Memory-based Particle Filter for Face Pose Tracking Robust under Complex Dynamics

Dan MIKAMI, Kazuhiro Otsuka, and Junji Yamato
NTT Communication Science Laboratories
3-1 Morinosato-Wakamiya, Atsugi, Kanagawa, 243-0198, Japan
mikami.dan@lab.ntt.co.jp, otsuka@eye.brl.ntt.co.jp, yamato@eye.brl.ntt.co.jp

## Abstract

*A novel particle filter, the Memory-based Particle Filter (M-PF), is proposed that can visually track moving objects that have complex dynamics. We aim to realize robustness against abrupt object movements and quick recovery from tracking failure caused by factors such as occlusions. To that end, we eliminate the Markov assumption from the previous particle filtering framework and predict the prior distribution of the target state from the long-term dynamics. More concretely, M-PF stores the past history of the estimated target states, and employs a random sampling from the history to generate prior distribution; it represents a novel PF formulation.Our method can handle nonlinear, time-variant, and non-Markov dynamics, which is not possible within existing PF frameworks. Accurate prior prediction based on proper dynamics model is especially effective for recovering lost tracks, because it can provide possible target states, which can drastically change since the track was lost. We target the face pose of seated humans in this paper. Quantitative evaluations with magnetic sensors confirm improved accuracy in face pose estimation and successful recovery from tracking loss. The proposed M-PF suggests a new paradigm for modeling systems with complex dynamics and so offers a various visual tracking applications.*

## 1. Introduction

Face pose tracking is one of the most demanded computer vision techniques and is necessary for a wide range of applications [12], such as human-computer interaction [18, 21], video surveillance, and meeting scene analysis [16]. Human head motion exhibits complex dynamics, not only smooth motion, but also abrupt changes in moving direction, such as the cases of head shaking and other gestures. Visual face tracking systems also suffer from occlusions, which may cause tracking loss. Occlusions can be caused by actions such as covering the mouth with hands and turning around. A face pose tracker should be robust against such complex human motions and able to regain tracking when challenged by occlusions.

Among the many face pose trackers that have been proposed [16, 5, 14, 21], Bayesian filter-based tracking is acknowledged as a promising approach. Bayesian filtering is a unified probabilistic framework for sequentially estimating the target state from an observed data stream [4]. At each time step, the Bayesian filter computes the posterior probability distribution of the target state by using observation likelihood and the prior distribution. The prior distribution is predicted from the posterior distribution estimated in the previous time step and the dynamics model of the target.

As typical implementations of Bayesian filter, Kalman filter and Particle Filters [2] (hereafter called PF) have been used for visual tracking [13, 15, 9]. The Kalman filter assumes linear dynamics with a Gaussian stochastic component; Gaussian prior and posterior distributions are used to represent the target state. Despite its simplicity, closed-form solutions can be derived, linear Gaussian dynamics is not sufficient to handle complex target motions. On the other hand, the particle filter can represent arbitrary probability distributions using sets of samples, called particles, and can potentially handle nonlinear non-Gaussian dynamics [2]. However, most PF-based visual trackers base their target motion model on linear Gaussian dynamics for simplicity. Such simple models can not match the complexity of human motions.

We aim to realize a robust tracker that can stably track the target's position and pose, both of which exhibit rapid motion and abrupt changes in moving direction, and can recover from tracking loss caused by occlusions. Specifically, we target human face pose in the seated situation. To that end, we focus on improving the prior distribution so as to more fully reflect human dynamics, which is nonlinear, time-variant, and non-Markov. An accurate prior prediction is important not only for accurate estimation of the target state, but also for recovering from the tracking loss caused by occlusions, because it allows possible target positions and poses to be determined at multiple time

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

step intervals since the track was lost. This paper proposes an advanced face pose tracker based on a novel particle filter called Memory-Based Particle Filter (M-PF). M-PF introduces a new paradigm; called memory-based prior prediction, it uses the past data of face poses to predict prior distributions. The memory-based prior prediction is done by weighted sampling of particles from the past history of face poses; we determine the sampling weight by calculating the temporal recurrent probability of face poses, which are obtained by a statistical analysis of actual head motion. We implemented an M-PF face pose tracker based on STC-Tracker (Sparse Template Condensation Tracker)[9].

This paper is organized as follows. Section 2 overviews related works. Section 3 proposes M-PF and Section 4 describes our face pose tracker based on M-PF. Section 5 introduces experimental results. Finally, Section 6 gives a discussion and draws our main conclusions.

## 2. Related works

### 2.1. Refinements of particle filtering

So far, several extension of particle filters have been proposed to deal with more complex dynamics than the typical simple models such as the random walk model and linear AR models. Such particle filters can be categorized into several classes, as follows.

First, some improved sampling strategies have been developed such as Sampling-Importance Resampling (SIR) [4] and Auxiliary Sampling Important Resampling (ASIR) [10]. Although, the sampling strategy can partly alleviate the negative effects of poor dynamics model, it is not the solution we seek.

Second, external cues have been employed for generating the prior distribution [19]. To generate the prior distribution, they utilize not only a dynamics model but also results from head pose detection. Though such approaches seem to suggest better tracking performance, they are effective only when a reliable detector is available.

Third, the switching dynamic model approach is used in a number of target trackers; they are based on the assumption that the type of target motion can be discretized into few distinct motion models [6, 1]. For example, the linear motion case and maneuvering cases are separately modeled in the tracker of [1]. Switching model-based tracking can be applied only when there are several distinct motion models that can be acquired by techniques such as a priori or online learning [6].

Fourth, self-organized filters have been proposed to deal with more complex dynamics [8]; they sequentially estimate the parameters of the dynamics model. This filtering approach well tracks targets that move abruptly since it automatically changes the variance of Gaussian distribution that represents the stochastic components in the dynamics. However, the range of dynamics that can be modeled is limited due to its requirement of parametric model form. Also, it is necessary to carefully choose the values of the hyper-parameters since they determine how the parameters of the dynamics model change over time.

In contrast to the existing approaches mentioned above, we propose a totally different approach, memory-based prior prediction. It can exploit long-term dynamics and can potentially handle nonlinear, time-variant, and non-Markov dynamics of complex target motion, which is not possible with existing switching and self-organized trackers.

### 2.2. Memory-based approach for complex real world phenomena

To predict and estimate the complex phenomena seen in the real world, the memory-based approach has been the subject of a lot of research over many years [11, 17, 20, 7]. The basic idea is that the same phenomenon repeatedly exhibits similar states, and its development over time is also similar to past instances. Based on this understanding, the memory-based approach searches for the past target state that is similar to the present one, and predicts the future state by assessing the future part of the retrieved past data. Memory-based predictions have been applied in currency exchange rate system[11], radar echo pattern analysis [17], and so on. One advantage of memory-based prediction is that it does not require an explicit model of the target dynamics which are often extremely complex and unattainable. The disadvantage is that it require large amounts of past data to allow adequate coverage; the amount and quality of the dataset directly influences prediction accuracy.

The method proposed in this paper employs an idea similar to the memory-based prediction mentioned above, and applies it to predict prior distributions of the tracking target in a particle filter framework. To the best of our knowledge, M-PF is the first attempt to incorporate memory-based prediction into PF. Our proposed method alleviates the data acquisition problem by using an online data accumulation process.

## 3. Memory-based Particle Filter

We define the memory-based particle filter (M-PF), whose immediate purpose is the robust tracking of face pose, but it is a generic framework for state estimation. M-PF can provide prior distributions that can reflect complex dynamics, which could be nonlinear, time-variant, and non-Markov. Section 3.1 starts by re-formulating Bayesian filtering to define M-PF. It assumes the long-term statistics of target dynamics; that is, similar states reappear often over time, and its temporal development follows similar paths seen in the past. The main idea of M-PF is to predict the prior distribution of the target state in future time steps by weighted sampling of the past sequence of target states; the sampling weight is determined based on the long-term dynamics of the targeted system. In Section 3.2, from statistical analysis of actual head motion, we introduce two properties of long-term dynamics: *stationary property* and *trajectory similarity*. Finally, a sampling strategy that combines the two properties is described in Section 3.3.

### 3.1. Re-formulation of Bayesian filter

To define M-PF, we first re-formulate the Bayesian filter and the particle filter. The Bayesian filter is a probabilistic framework for sequentially estimating the target's state; here we denote it as vector $\mathrm{x}_t$ at discrete time step $t$. The Bayesian filter consists of two processes, *update* and *prediction*, of the target's state distributions, which are, respectively, written as in

$$p(\mathrm{x}_t|\mathrm{Z}_{1:t}) \;:=\; k_t \cdot p(\mathrm{z}_t|\mathrm{x}_t) \cdot p(\mathrm{x}_t|\mathrm{Z}_{1:t-1}) \qquad (1)$$

$$p(\mathrm{x}_{t+1}|\mathrm{Z}_{1:t}) \;:=\; \int p(\mathrm{x}_{t+1}|\mathrm{X}_{1:t}) \cdot p(\mathrm{x}_t|\mathrm{Z}_{1:t}) d\mathrm{x}_t, \quad (2)$$

where $k_t$ is a normalization term, $\mathrm{z}_t$ denotes the observation vector obtained at time $t$, and $\mathrm{Z}_{1:t}$ denotes the sequence of observations, $\mathrm{Z}_{1:t} = \{\mathrm{z}_1, \mathrm{z}_2, ..., \mathrm{z}_t\}$. Also, $\mathrm{X}_{1:t}$ denotes its history, $\mathrm{X}_{1:t} = \{\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_t\}$, from initial time step to $t$. Eq.(1) corresponds to the update process that computes the posterior probability distribution of the target state by using observation likelihood $p(\mathrm{z}_t|\mathrm{x}_t)$ and prior distribution $p(\mathrm{x}_t|\mathrm{Z}_{1:t-1})$ at time $t$. Eq.(2) corresponds to the prediction process, which calculates the prior distribution for the next time step by convoluting the posterior distribution estimated at time step $t$, and transition probability distribution $p(\mathrm{x}_{t+1}|\mathrm{X}_{1:t})$, which represents the dynamics of the target. Most particle filters assume Markov dynamics, i.e. $p(\mathrm{x}_{t+1}|\mathrm{X}_{1:t}) = p(\mathrm{x}_{t+1}|\mathrm{x}_t)$. At each time step, from the posterior distribution, the point estimates of target state, $\widehat{\mathrm{x}}_t$, are calculated; typical estimates are the mean value or maximum posterior estimates of the posterior distribution.

The M-PF eliminates the Markov property in computing the prior distribution and predicts the prior distribution based on the total past state history. The M-PF replaces the prior distribution in Eq.(2) with memory-based prior distribution, $\pi$, as written in

$$p(\mathrm{x}_{T+\Delta t}|Z_{1:T}) := \pi(\mathrm{x}_{T+\Delta t}|\widehat{X}_{1:T}, \Delta t), \qquad (3)$$

where $\widehat{X}_{1:T}$ denotes the sequence of point estimates of the target state, $\widehat{\mathrm{X}}_{1:T} = \{\widehat{\mathrm{x}}_1, \cdots, \widehat{\mathrm{x}}_T\}$, which are obtained up to current time step $T$ from $t = 1$. In Eq.(3), $\pi(\mathrm{x}_{T+\Delta t}|\widehat{\mathrm{X}}_{1:T}, \Delta t)$ represents memory-based prior distribution, i.e. prediction at $\Delta t$ time steps later. The memory-based prior distribution is defined from the past estimates of the target state. To achieve this, we introduce temporal recurrent probability, it indicates the tendency of the past state to reappear in the future. Specifically, we define it as a mixture distribution of past state estimates with a certain kernel distribution, as written in

$$\pi(\mathrm{x}_{T+\Delta t}|\widehat{X}_{1:T}, \Delta t) := \sum_{t=1}^{T} \Phi(t|\widehat{X}_{1:T}, \Delta t) \cdot K(\mathrm{x}_{T+\Delta t}|\widehat{\mathrm{x}}_t), \qquad (4)$$

where $\Phi(t|\widehat{X}_{1:T}, \Delta t)$ denotes the temporal recurrent probability, which is defined as the probability of past state $\widehat{\mathrm{x}}_t$ at time $t(\leq T)$ reappearing at future time step $T + \Delta t$, where $\sum_{t=1}^{T} \Phi(t|\cdot) = 1$. In Eq.(4), $K(\mathrm{x}_{T+\Delta t}|\widehat{\mathrm{x}}_t)$ denotes the kernel distribution that represents two aspects of the uncertainty in the prior distribution estimation. One is uncertainty in state estimates $\widehat{\mathrm{x}}_t$, which is once obtained as its posterior distribution $p(\mathrm{x}_t|Z_{1:t})$; the kernel distribution is used to approximately simulate this posterior distribution. The other is the uncertainty in predictions based on past data. That is, in a continuous state space, strictly speaking, it is thought that exactly the same state will not reappear, but something similar to a past state may occur. The kernel distribution is used to reflect the difference. As one example of the kernel distributions possible, a multivariate Gaussian distribution, $N(\mathrm{x}|\mu, \Sigma)$, is used in this paper, as written in $K(\mathrm{x}|\widehat{\mathrm{x}}_t) := N(\mathrm{x}|\widehat{\mathrm{x}}_t, \Sigma)$, where $\mu$ and $\Sigma$ denote the mean vector and covariance matrix of the Gaussian distribution, respectively.

M-PF generates multiple samples from the memory-based prior distribution. The sampling process consists of two stages; i) sampling past time steps using the temporal recurrent probability, and ii) sampling from the Kernel distribution, as written in

$$\begin{array}{ll} \text{stage 1:} & t^* \sim \Phi(t|\widehat{X}_{1:T}, \Delta t) \\ \text{stage 2:} & \mathrm{x}^* \sim K(\mathrm{x}|\widehat{\mathrm{x}}_{t^*}). \end{array} \qquad (5)$$

This sampling process in Eqs. (5) is repeated for $N$ times, where $N$ is the number of particles. The resulting sampling set, $\{\mathrm{x}^{*(1)}, \mathrm{x}^{*(2)}, \cdots, \mathrm{x}^{*(N)}\}$, represents the prior distribution at time $T + \Delta t$.

In traditional PF, the set of particles, which represents the prior distribution, is generated from the set of particles that represent the posterior distribution. The M-PF replaces this sampling-from-present-posterior with sampling-from-past-posterior estimates. In other words, M-PF solves the problem of creating a multi-dimensional spatial distribution (i.e. prior) by creating a 1-dimensional temporal distribution. Here, this temporal distribution is called the temporal recurrent probability distribution, it indicates the likelihood of past states reappearing; it is discussed in the following subsections.

### 3.2. Long-term dynamics of target motion

As described in Section 3.1, the temporal recurrent probability is introduced to represent such long-term dynamics; how often and how similar past states reappear in the future. The appropriate target is thought to be physically constrained targets with bounded state space. The head pose of seated humans is one typical target of such a system,

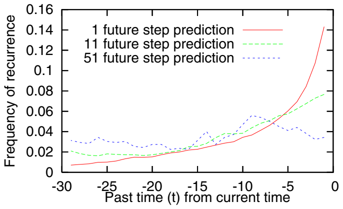


Figure 1. Histogram of past time steps whose state is the best prediction; Origin is current time. Time axis is relative to the current time.

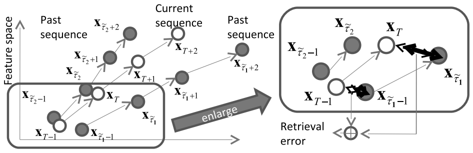

Feature space
Past sequence
Current sequence
Past sequence
$\mathbf{x}_{\tilde{\tau}_2+2}$
$\mathbf{x}_{\tilde{\tau}_2+1}$
$\mathbf{x}_{\tilde{\tau}_2}$
$\mathbf{x}_{\tilde{\tau}_2-1}$
$\mathbf{x}_{T+2}$
$\mathbf{x}_{T+1}$
$\mathbf{x}_{T}$
$\mathbf{x}_{T-1}$
$\mathbf{x}_{\tilde{\tau}_1+2}$
$\mathbf{x}_{\tilde{\tau}_1+1}$
$\mathbf{x}_{\tilde{\tau}_1}$
$\mathbf{x}_{\tilde{\tau}_1-1}$
enlarge
Retrieval error

Figure 2. Similar trajectories are retrieved; The more similar trajectories tend to yield to more similar face poses in future ( especially in small $\Delta t$ situations). Even for similar trajectories, the greater the prediction lead time is, the larger the estimation error becomes.

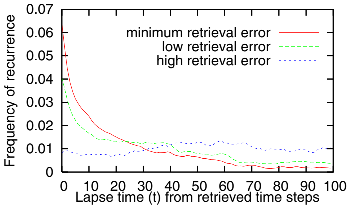


Figure 3. Histogram of past time steps whose state is the best prediction; conditioned on different levels of retrieval error. Time axis is relative to the past time point retrieved. Prediction lead time is 1.

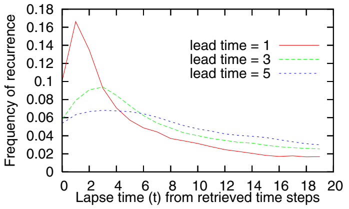


Figure 4. Histogram of past time steps whose state is the best prediction; conditioned on different lead times ($\Delta t = 1, 3, 5$). Time axis is relative to retrieved past time point.

because bone structure and its articulation, including position and rotational angles of face, constrain poses possible. From a statistical analysis of actual human head pose, we have discovered two properties of recurrence, called stationary property and trajectory-similarity property, and use them to define the temporal recurrent probability for M-PF.

#### 3.2.1 Stationary property

The stationary property has two aspects. One is that the target state in the near-term future tends to be similar to the state in the recent past (Property 1-i). The other is that target state in the far future tends not to be found in the recent past (Property 1-ii). Fig. 1 shows an example of the statistical analysis of the stationary property, based on data of the head pose sequences of a person in seated condition[1]. Fig. 1 shows a histogram of the relative temporal position of the past state that was the most similar to the future state at 1, 11, and 51 time steps ahead, as a function of relative time $\tau(< 0)$ against the present time step $T$. Fig. 1 confirms that recent past has a strong peak with small lead-time, and as the prediction is cast further into the future, the histgram broadens. The former holds Property 1-i, and the latter holds Property 1-ii. The form of the histogram in Fig. 1 leads to the stationary-property-based temporal recurrent probability, $\phi^S(\tau|\Delta t), \tau \leq 0$, which is defined over the temporal position relative to the present time $T$. The recurrent probability on absolute time can be defined as $\Phi^S(t|\Delta t) := \phi^S(t - T|\Delta t)$, where $\sum_{t=1}^{T} \Phi^S(t|\Delta t) = 1$. The temporal recurrent probability can be modeled by using parametric distributions such as Gamma distributions and truncated uniform distributions.

#### 3.2.2 Trajectory similarity

The trajectory similarity is based on the observation that when the current state and a past state are similar, the temporal development of the current state tends to follow the path that the past state went through. We introduce two properties, as illustrated in Fig. 2. One is that the more similar the current and past states are, the more its future path will continue to be similar, and vice versa (Property 2-i). The other is that the further into the future the prediction is cast, the lower the prediction accuracy becomes, and vice versa (Property 2-ii).

To verify these properties in actual data and to define the trajectory-similarity-based recurrent probability, we need to evaluate the (dis)similarity between the current sequence and past sub-sequences, and to find (retrieve) the set of past subsequences that is most useful for prior prediction. This paper defines the (dis)similarity measure, $\epsilon$, (the retrieval error) using Euclid distance, as written in

$$\epsilon(T, t) = |\mathrm{x}_T - \mathrm{x}_t| + |\mathrm{x}_{T-1} - \mathrm{x}_{t-1}|, \tag{6}$$

where $|\cdot|$ denotes L2 norm. Eq.(6) evaluates (dis)similarity in terms of not only the distance between current state $\mathrm{x}_T$ and past state $\mathrm{x}_t, (t < T)$, but also the temporal development sequence. Using measure $\epsilon$, the set of time steps, whose state is similar to the current state, is obtained as $\mathcal{T} = \{\tilde{\tau}_1, \tilde{\tau}_2, \cdots, \tilde{\tau}_M\}$, where $M$ is the number of retrieved past time steps.

Fig. 3 shows the histogram of relative temporal position that is most similar to 1-step future state $\mathrm{x}_{T+1}$, as a function of time relative to the retrieved past time $\tilde{\tau} \in \mathcal{T}$.

[1] Data was captured with magnetic sensor. The person behaved naturally for 3 min in front of a desk.

Fig. 3 is based on the same data as in used in Fig. 1. Fig. 3 shows three curves corresponding to three different levels of retrieval error. The line with small retrieval error shows a tendency that the data similar to $\mathrm{x}_{T+1}$ appears in the near future of past retrieved data. It confirms property 2-i (lower retrieved error yields more accurate predictions). Fig. 4 shows the same histogram as in Fig. 3, but it shows two cases with different lead time, $\Delta t = 1$, $\Delta t = 3$ and $\Delta t = 5$. An analysis of data including Fig. 4 indicates that the peak position of the distribution moves forward, but its shape broadens as the lead time increases. This result indicates that the temporal development of current sequence and the past retrieved sequence are similar to each other, especially if the leadtime is small. Due to the fact that there is variance in the speed of temporal development, the distribution's shape broadens.

We build the trajectory-similarity-based temporal recurrent probability distribution, $\Phi^T(t)$ by combining those of each retrieved past subsequence, as written in

$$\Phi^T(t) = \sum_{i=1}^{M} w_i \cdot \psi(t - \widetilde{\tau}_i|\Delta t), \tag{7}$$

where $\psi(\tau|\Delta t)$ denotes the temporal recurrent probability distribution of a retrieved subsequence; it is defined over time relative to the retrieved time step, $\widetilde{\tau}_i$. We can model $\psi(\tau|\Delta t)$ by using Gamma distributions; its shape changes according to the prediction lead time $\Delta t$; it is peaky for short lead times and broad for long lead times. Moreover, the peak position moves from $\tau = 0$ to $\tau > 0$, as lead time $\Delta t$ increases. In Eq.(7), $w_i$ denotes the mixture weights, which are set according to the retrieval error; larger weights are given to past sequences with small retrieval error.

### 3.3. Sampling strategy

We define the temporal recurrent probability $\Phi(t)$ as a mixture of two distributions, $\Phi^S(t)$ and $\Phi^T(t)$ corresponding to each property in 3.2.1 (stationary property) and 3.2.2 (trajectory similarity), respectively, in order to cover a wide range of target dynamics. The stationary-property-based model is effective when the target tends to remain in the same state for some time and abruptly changes direction. Also, it is useful when predicting the possible distribution of a long lost track, i.e. the distribution converged into the a priori distribution, which is static and time-invariant. The trajectory-similarity-based model is more effective when the target with forward motion includes rapid motion and when there is past data similar to the current state.

As shown in Fig. 5, recurrent distribution $\Phi(t|\Delta t)$ is defined as a mixture of the two model distributions, as written in

$$\Phi(t|\Delta t) = w^S \cdot \Phi^S(t|\Delta t) + w^T \cdot \Phi^T(t|\Delta t), \tag{8}$$

where $w^S$ and $w^T$ denote the weights, and $w^S + w^T = 1$. The memory-based prior distribution is defined in Eq.(4), and sampling process is described by Eq.(5). The numbers of samples from the two distributions are determined by weights $w^S$ and $w^T$. These are determined based on minimum retrieval error; when retrieval error is large, fewer samples are generated from the trajectory-similarity-based recurrent distribution, and relatively more samples are generated from the stationary-property-based distribution.

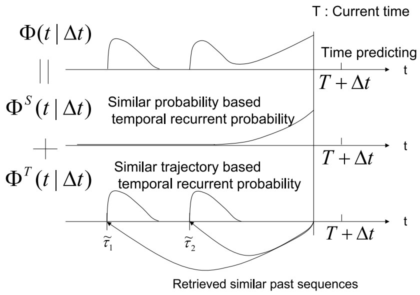
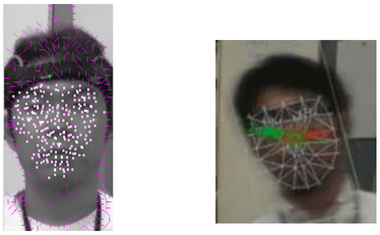


Figure 5. Temporal recurrent probability distributions



(a) Feature points indicated by white dots; A person attaches two magnetic sensor on each temple.

(b) White mesh indicates face pose; red and green dots denote stationary-property based and trajectory-similarity based prior respectively.

Figure 6. Tracked person and detected features.

## 4. Memory-based Face Pose Tracker

This section describes a face pose tracker based on the memory-based particle filtering (M-PF) proposed in Section 3. We incorporate memory-based prior prediction into the face pose tracker STCTracker (Sparse Template Condensation Tracker) [9].

### 4.1. STCTracker: Sparse Template Condensation Tracker

The basic idea of STCTracker [9] is combining template matching with particle filtering. In contrast to traditional template matching, which assesses all pixels in a rectangular region, sparse template matching focuses on a sparse set of feature points within a template region. Fig. 6 shows an example of the face template; Fig. 6(a) shows the feature points and the white mesh in Fig. 6(b) indicates the shape model of the face template. The state of a template, which represents the position and pose of the face, is defined as a 7 dimensional vector consisting of 2-DOF translation on the image plane, 3-DOF rotation, scale (we assume weak-perspective projection), and an illumination coefficient. The weight of each particle, which has a face pose candidate, is calculated based on matching error between input images and the template whose state is assigned by each particle; higher weight is given to particles with smaller matching error. STCTracker employs GPU(Graphics Processing Unit) processing to accelerate the weight computation of particles; it was shown to be 10 times faster than the CPU-only version.

### 4.2. Flow of memory-based face pose tracking

Figure 7 shows the flow of our face pose tracker based on M-PF, here we call it the memory-based face pose tracker. The key extension from the previous STCTracker is the prior prediction part, which generates particles based on our memory-based method. In addition, we define two tracking modes depending on tracking stability, one is "stable" and the other is "lost". This tracking mode can be decided by simple thresholding of the minimum matching error of particles. If the minimum matching error exceeds a certain value, we consider that the track has been lost and try to rediscover the face. Otherwise, one-step prior prediction is conducted as usual. For rediscovering the lost face, the prior distribution at the next time step is obtained in multiple-step prediction from the time at which the track was lost. Furthermore, we altered the basic M-PF defined in Section 3 by incorporating a data clustering procedure to suppress the memory needed for storing past data and the time spent searching for past similar sequences, which monotonically increase as time passes. The past face poses are represented as a set of clusters. Each cluster contains at least one face pose. When the point estimate at current time $t$ is obtained, the new estimate is added to one of the existing clusters or a new cluster is created for it; the decision is based on a distance measure. We simply replace the past sequences with the set of clusters.

The proposed memory-based face tracker can start without any past memory and accumulate the face pose estimates at each time step to build the database. Without past memory, the behavior of our tracker is equivalent to that of traditional tracking with random walk dynamics, since we set the temporal recurrent probability to become 1 at the current time step. Also, when initializing the tracker, it can load a face pose sequence that was obtained at previous tracking sessions; the preloaded sequences could be obtained from the same person or others.

## 5. Experiments and results

We conducted several experiments to verify the effectiveness of our method. This section first explains the experimental environment, and then presents its performance in terms of tracking accuracy and abrupt motion robustness. Its ability to recover from tracking failure is also shown.

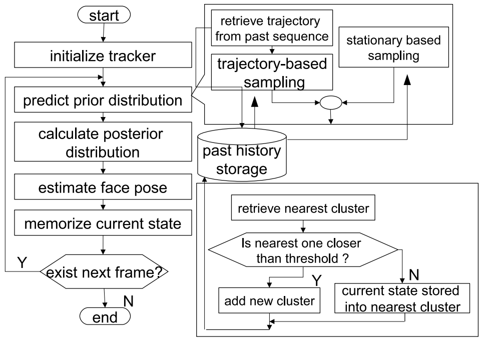

start
initialize tracker
predict prior distribution
calculate posterior distribution
estimate face pose
memorize current state
exist next frame?
Y
N
end
retrieve trajectory from past sequence
trajectory-based sampling
stationary based sampling
past history storage
retrieve nearest cluster
Is nearest one closer than threshold ?
Y
N
add new cluster
current state stored into nearest cluster

Figure 7. Flowchart of our method face pose tracker.

### 5.1. Experimental Environment

We used PointGreyResearch's FLEA, a digital color camera, to capture 1024 × 768 pixel-size images at 30 frames per second. Note that the tracking processes use only grayscale images converted from the color images. The CPU of the PC used was an Intel Core2Extreme 3.0GHz (Quad Core) and the GPU was NVIDIA GeForce GTX280. A magnetic-based sensor, Polhemus FASTRAK was used to obtain quantitative ground truth data. The camera and the receiver were set to share the same coordinate system, as in [3]. As shown in Fig. 6, two sensors were attached to both temples of the person. The Z axis of the magnetic sensor coordinate system was set to the lens axis, and the horizontal and vertical axis of the sensor coordinate system were set horizontal and vertical, respectively. The rotation angles, pitch, roll, and yaw were defined as rotation around the Y, X, Z, axes, respectively; they correspond to shaking, nodding, and tilting actions, respectively. We employed Gamma distributions as the models of temporal recurrent probabilities of the stationary property and trajectory similarity. We confirmed that the proposed face pose tracker ran at 30.0 frames per second, using realtime input from the camera. The number of particles was set to 2000.

Fig. 6(b) displays a typical output screen of our face tracker. In Fig. 6(b), the white mesh indicates the face shape model; its position and pose are the outputs of the tracker. The cloud of red points near the face's center indicate the particles representing the prior distribution; red points indicate the stationary-property-based samples and green points indicate the trajectory-similarity-based samples.

### 5.2. Accuracy of pose estimates: moderate and abrupt-motion cases

To verify the accuracy of pose estimates, we used two types of videos, Type I, and Type II. Type I video includes only moderate motion, and both proposed and former trackers could follow the face without loss. Type II video includes abrupt movement (spatial displacement between adjacent frames was up to 20 ∼ 50 pixels). These two types of

Table 1. Average and variance of estimated error in pose angles [deg] (Type I videos (moderate case) and Type II videos (abrupt-motion case).

| | Type I (moderate) | | | Type II (abrupt) | | |
|---|---|---|---|---|---|---|
| | Pitch | Roll | Yaw | Pitch | Roll | Yaw |
| Ave. (Proposed) | 2.73 | 4.49 | 0.59 | 8.54 | 3.67 | 2.39 |
| Var. (Proposed) | 4.73 | 2.61 | 0.21 | 51.99 | 13.15 | 10.99 |
| Ave. (STCTracker) | 3.32 | 3.66 | 0.65 | 17.35 | 5.00 | 8.85 |
| Var. (STCTracker) | 5.68 | 4.36 | 0.30 | 316.48 | 28.63 | 315.78 |
| Ave. (S-Model) | 3.57 | 3.46 | 0.67 | 13.1 | 4.36 | 2.61 |
| Var. (S-Model) | 5.99 | 4.56 | 0.31 | 134.3 | 13.65 | 12.73 |

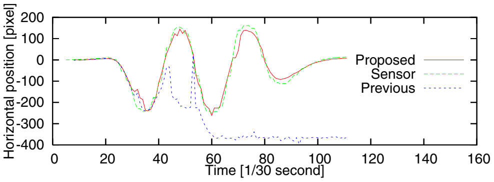
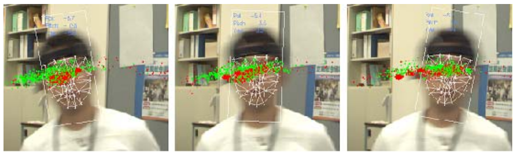


Figure 8. Comparisons of magnetic sensor tracker, previous, and proposed tracker ( Time series of horizontal position).

videos were captured from two people, so 4 videos in total were used in this evaluation. Two previous methods were employed for evaluations; one is the STCTracker with random walk dynamics model as described in [9], the other is a STCTracker with switching dynamics model (S-Model) between random walk, first order (uniform motion), and second order (uniform accelerated motion) dynamics models. Table 1 shows the average and variance of the estimation errors (average absolute error) in rotation angles; the error values were averaged over the two participants, separately for Type I and Type II. Table 1 compares the results from the proposed method and previous methods (the STCTracker and the S-Model). Fig. 8 shows the temporal sequences of the position from Type II video (abrupt case), using the proposed tracker and the STCTracker. Fig. 9 shows sequential snapshots of tracking results output by the proposed method. Although this scene exhibits large significant motion and resulting image blur, our tracker successfully followed the target face.

For Type I, all methods basically offered the same tracking accuracy, because the face motion was small enough that the previous method could follow it through its use of the random-walk dynamic model. For Type II, the results confirm that the proposed method was significantly better than the conventional trackers. This is because our tracker successfully predicted the prior distributions for abrupt motion. Note that the proposed tracker started with no past data, and accumulated data online. Despite the fact that the accumulated data was quite short (up to 600 frames at the end of video), the memory-based prediction was surprisingly effective.



(a) T (b) T+1 (c) T+2

Figure 9. Tracking of quick motion by our method.

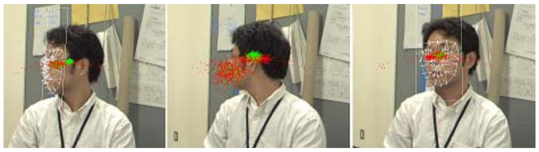


Figure 10. Successful recovery from tracking failure (12 frame interval between above images).

Table 2. Time spent for each compornents of previous PF and M-PF in each frame [msec].

| | Prior | Likelihood | Post Process | Total |
|---|---|---|---|---|
| STCTracker | 0.77 | 6.0 | 0.08 | 6.85 |
| Proposed | 0.98 | 6.0 | 0.14 | 7.12 |

### 5.3. Recovery from tracking failure

To investigate recovery from tracking failure, we used another class of video clips, called Type III, which consisted of a series of actions that caused visual occlusion and resulting track loss. Typical action was turning-back motion, i.e. face repeatedly leaves the camera's view and then returns. As the measure of track "recoverability", we calculated the ratio of number of successful recoveries to the number of all tracking failures, for the proposed method and the STCTracker with random walk dynamics model and with switching model. The proposed method achieved the successful recovery rate of 100 % with no human intervention. The STCTrackers often lost the track permanently and had to be manually reset; its recovery rates were 55% and 47%. Fig. 10 shows the behavior of our tracker when recovering from self-occlusion. After the occlusion triggered tracking failure, the particles spread over wider ranges as the elapsed time increased as anticipated from the long-term dynamics assumed. The excellent recoverability of the proposed tracker will be the key to successful fully-automated applications like consumer products.

### 5.4. Memory usage and processing time

Under the conditions of experiments, immediately after tracking started, the number of clusters grew rapidly, and then saturated within a few minutes at about 700. This confirms the effectiveness of the memory usage suppression.

The processing times of STCTracker and our memory-based face pose tracker are shown here. The main components of the PF/M-PF framework include "prior prediction", "calculation of likelihood", and "post processing (calculating the state statistics)". M-PF's overheads reside in, unlike PF, "prior prediction" and "post processing". The former includes sampling prior from the past, and the latter conducts clustering for history storage. The approximate elapsed times are given Table2. Here, we measured the elapsed times under the condition that the number of clusters had saturated about 1500. This confirms two facts; one is that most of the processing time was used for "calculation of likelihood" which is commonly used by both previous PF and M-PF. The other is the overheads in "prior prediction" and "post processing" which require additional processing for M-PF are, surprisingly not large.

From these facts, we consider that the overheads of M-PF are relatively trivial, and real-time applications are likely.

## 6. Conclusions and discussions

We proposed a novel particle filter, called M-PF, for the visual tracking of human face pose, which has complex dynamics. M-PF can predict accurate prior distributions by sampling past pose estimates, according to temporal recurrent probability, which is derived from statistical analysis; similar target states reappear often, and the temporal changes repeat those seen in the past. M-PF can handle nonlinear, time-variant, and non-Markov dynamics without explicit modeling, and can effectively predict not only prior distributions for the immediate next step, but also for multiple steps, which allows for recovery from track loss. Quantitative evaluations with magnetic sensors confirmed that the proposed face tracker can accurately estimate face poses, whose motions are rapid and abrupt, and can successfully recover from tracking failures.

Future works include the following. First, more comprehensive evaluations are required, such as tracking accuracy vs. stored data amount and person-specific data vs. others' data. Second, data assimilation is needed in some cases, e.g. normalization based on one's sitting height. Third, it is desirable to extend our tracker to multiple targets. Such application might require a personal identification function when recovering lost faces over wider areas that might hold multiple faces. Also, there is the possibility of applying M-PF to articulated motion, such as human body poses. Finally, the authors believe that the proposed M-PF offers a new paradigm for modeling systems with complex dynamics and so will lead to a wide range of applications.

## References

[1] T. Bando, T. Shibata, K. Doya, and S. Ishii. Switching particle filters for efficient visual tracking. *Robotics and Autonomous Systems*, (54):873–884, 2006.

[2] B.Ristic. *Beyond the Kalman Filter: Particle Filters for Tracking Aplications*. Arthech House, 2004.

[3] M. L. Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models. *IEEE Trans. PAMI*, 22(4), 2000.

[4] N. Gordon, D.Salmond, and A.F.M.Smith. Novel approach to non-linear and non-Gaussian Bayesian state estimeation. *IEE proceedings F:Communications Rader and Signal Processing*, 140(2):107–113, 1993.

[5] J. Huang and M. Trivedi. Face pose discrimination using support vector machines (SVM). In *Proc. ICPR*, pages 154–156, 1998.

[6] K. Ishiguro, T. Yamada, and N. Ueda. Simultaneous clustering and tracking unknown number of objects. In *Proc. IEEE CVPR*, pages 1–8, 2008.

[7] J.D.Farmer and J.J.Sidorowich. Predicting chaotic time series. *Physical Letter Review*, 59(8):845–848, 1987.

[8] G. Kitagawa. Self-organizing state space model. *J. Amer. Statist. Assoc.*, 93(443):1203–1212, 1998.

[9] O. M. Lozano and K. Otsuka. Real-time visual tracker by stream processing. *Journal of VLSI Signal Processing Systems*, 2008. DOI 10.1007/s11265-008-0250-2.

[10] M.Pitt and N.Shephard. Filtering via simulation: Auxiliary particle filters. *J. Amer. Statist. Assoc.*, 94(446):590–599, 1999.

[11] F. J. Mulhern and R. J. Caprara. A nearest neighbor model for forecasting market response. In *Int'l J. Forecasting*, 1994.

[12] E. Murphy-Chutorian and M. M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Trans. PAMI, Digital Library*, 2008. DOI 10.1109/TPAMI.2008.106.

[13] E. Murphy-Chutorian and M. M. Trivedi. Hybrid head orientation and position estimation (HyHOPE): A system and evaluation for driver support. In *Proc. IEEE Intelligent Vehicles Symposium*, 2008.

[14] S. Niyogi and W. Freeman. Example-based head tracking. In *Proc. IEEE Int'l conf. FG*, pages 374–378, 1996.

[15] K. Oka, Y. Sato, Y.Nakanishi, and H.Koike. Head pose estimation system based on particle filtering with adaptive diffusion control. In *Proc. IAPR MVA*, pages 586–589, 2005.

[16] K. Otsuka, S. Araki, K. Ishizuka, M. Fujimoto, M. Heinrich, and J. Yamato. A realtime multimodal system for analyzing group meeting by combining face pose tracking and speaker diarization. In *Proc. ACM ICMI*, pages 257–264, 2008.

[17] K. Otsuka, T. Horikoshi, S. Suzuki, and H. Kojima. Memory-based forecasting for weather image patterns. In *Proc. of 17th AAAI*, pages 330–336, July 2000.

[18] G. R.Bradski. Computer vision face tracking for use in a perceptual user interface. In *Proc. IEEE Workshop Applications of Computer Vision*, pages 214–219, 1998.

[19] S.Ba and J.-M. Odobez. Probabilistic head pose tracking evaluation in single and multiple camera. *Multimodal Technologies for Perception of Humans*, 4625/2008:276–286, 2008.

[20] T.Ikeguchi and K.Aihara. Prediction of chaotic time series with noise. *IEICE Trans. on Fundamentals*, E78-A(10):1291–1298, 1995.

[21] J. Tua, H. Taob, and T. Huang. Face as mouse through visual face tracking. *CVIU*, 108(1–2):35–40, 2007.