

3D Object Tracking Using Shape-Encoded Particle Propagation

H. Moon, R. Chellappa, A. Rosenfeld

Center for Automation Research and Dept. of Electrical and Computer Engineering
University of Maryland
College Park, MD 20742-3275 *

Abstract

We present a comprehensive treatment of 3D object tracking by posing it as a nonlinear state estimation problem. The measurements are derived using the outputs of shape-encoded filters. The nonlinear state estimation is performed by solving the Zakai equation, and we use the branching particle propagation method for computing the solution. The unnormalized conditional density for the solution to the Zakai equation is realized by the weight of the particle. We first sample a set of particles approximating the initial distribution of the state vector conditioned on the observations, where each particle encodes the set of geometric parameters of the object. The weight of the particle represents geometric and temporal fit, which is computed bottom-up from the raw image using a shape-encoded filter. The particles branch so that the mean number of offspring is proportional to the weight. Time update is handled by employing a second-order motion model, combined with local stochastic search to minimize the prediction error. The prediction adjustment suggested by system identification theory is empirically verified to contribute to global stability. The amount of diffusion is effectively adjusted using a Kalman updating of the covariance matrix. We have successfully applied this method to human head tracking, where we estimate head motion and compute structure using simple head and facial feature models.

1 Introduction

Using object shape information for tracking is useful when it is difficult to extract reliable features for tracking and motion computation. In many cases, an object in a video sequence constitutes a *perceptual unit* which can be approximated by a limited set of *shapes*. Many man-made objects provide examples. A human body can also be decomposed into approximate shapes such as an ellipsoid for

*Partially supported by the Office of Naval Research under the Grant N00014-011-0265.

the head and truncated cones for the limbs. For tracking or motion computation of human activities, local features are noisy and often not reliable for establishing temporal correspondences. Shape constraints also provide strong clues about object geometry while the object is moving. ‘Shape’ in this context refers to the image plane projection of the object, which can be approximated by simple figures.

We perform detection and tracking of shapes using an optimal shape operator which was introduced in [16]. The responses of an image frame to a set of shape filters having certain ranges of geometric parameters are used as observations in a nonlinear state space formulation, to guide object tracking and estimate the motion. The magnitudes of the responses are accurate and robust to noise, so that they achieve reliable estimates of geometric parameters (location, orientation, size), and provide a strong temporal correspondence for tracking the object in subsequent frames.

Many motion problems have been treated as posterior state estimation problems, and typically solved using Kalman or extended Kalman filters (EKF) [1][4]. Since, in our approach, the observation is a set of responses obtained from shape filters, the functional relation between the geometric parameter space and the image space makes the observation process highly nonlinear, or even nonanalytic. There is a generalization of the Kalman filter to the nonlinear case, by Duncan, Mortensen, and Zakai [18]. They derived an equation which incorporates both dynamic and observation equations, and which, if solved, gives the temporal propagation of the probability of the states conditioned on the observations.

Recently, the application of Monte Carlo simulation to tracking and motion computation problems has become popular. Mainly due to advances in computing power, applications to the state estimation problem [11] [14] have been proposed in the statistics community. [10] introduces the *Condensation* algorithm for tracking, and [7] and [17] further refine the method by using a layered sampling for accurate object localization and effective search for the state parameters. [12] uses the framework of Sequential Importance Sampling [14] to solve the problem of simultaneous object

tracking and verification. In the proposed method, shape filtering, viewed as a measurement process is elegantly incorporated into the nonlinear filtering framework, which contributes to the accurate computation of the weight. The solution using the branching particle method has a strong analytical foundation based on the Zakai equation, from which the expression for computing the weights follows directly. The expression of the unnormalized conditional density for computing weights involves both *geometric fit* of the data and *temporal coherence* of the motion, and the shape filter is designed to achieve accuracy with respect to both criteria. The method of estimating the number of offspring using randomized sampling is also designed to be optimal, while the total number of samples is fixed in resampling approaches. The often neglected problem of determining the time update is handled nicely in the proposed method using the Kalman filter equation.

A recent paper [9] has introduced the Zakai equation to image analysis problems. Our approach resembles this work; shape filters are used in our work, just as wavelet filters are used in [9]. They utilized a mixture of analytical/numerical methods to compute the solution. We employ a branching particle method; the system of particles which mimics the conditional density of states is found [6] to converge to the target distribution.

After branching, the particles should follow the system dynamics and random perturbation. As we cannot assume any particular motion model in most applications, we employ crude second-order motion prediction. The prediction is modified by a random search to minimize the prediction error. This step is suggested by system identification theory [15], and the benefit is empirically verified. The amount of random diffusion — formally, the state error — has to be determined, which we found to be very crucial for stable tracking. The state error covariance matrix is computed by subtracting the prior covariance matrix from the posterior covariance matrix, according to the Kalman filter time update equations. We found that the computed covariances adapt to the motion, and usually are very small; nevertheless, this method of computing the diffusion shows noticeable improvements in tracking and pose estimation.

We have applied this method of shape tracking to the problem of human head tracking in a monocular video sequence. The head is modeled as an ellipsoid, and the motion of the head as rotation combined with translation, having a total of six degrees of freedom. Facial features are approximated as simple geometric curves; we can compute the operators for tracking the features given the hypothetical pose of the head and the positions and sizes of the features, by using the inverse camera projection. Experiments show that the particles are able to track and estimate the head motion accurately. The algorithm also estimates the size, pose, and location (up to scale) of the ellipsoid simultaneously.

2 Shape and measurements

In the general context of object recognition or tracking, the outline of an object gives a compact representation of the appearance of the object, whereas color or texture information is usually highly variable with different object configurations or imaging environments. The boundary contour of an object gives clues for detection/recognition which are almost invariant to imaging conditions except for the camera parameters.

Methods for appearance-based tracking using a linear subspace representation [3] or an object template [13] have been considered. While these methods use holistic representations of object intensity structure, which can be effectively used to recognize or classify objects in video, they have a very limited ability to represent and compute changes in object pose. Nevertheless, the use of a global object representation has the advantage that it helps to maintain the temporal correspondence of features. The point set based approach is flexible, simple, and elegant in terms of algebraic manipulation, but it is hard to reliably extract and correspond point features in real-world videos.

When we have a geometric model of the shape of a solid object, or a kinematic and shape model of a structured object (e.g., the human body), we can manipulate it to fit the motion of the model to an object in the video using any prediction method (e.g., a Kalman filter). The model and the scene are usually compared using edge features. [8] deals with the problem of tracking objects with known 3D shapes. Shape constraints provide more information about the object configuration or the imaging conditions than point features; the deformation of shapes under changes in object pose or camera parameters (e.g., focal length) provides better clues about these parameters, while points (e.g., endpoints, vertices, junctions), being subsets of shapes, often cannot. We have observed that shape constraints effectively stabilize tracking when the pose parameters deviate from their real values after a rapid motion.

We make use of an approximate shape model of an object, and of boundary gradient information extracted using this model, for tracking and motion estimation. Given the predicted object size, position, and pose, the projection of the model object is compared to the image using the set of shape filters. Using the optimal shape detection and localization technique derived in [16], the accurate responses of the shape operators provide the tracker with an accurate geometrical fit of the model to data, and a strong temporal correspondence between frames. The detection performance is equivalent to the accuracy of the filter response, while the localization performance is closely related to the recognition/discrimination of shapes. In [16], the optimal one-dimensional smoothing operator, designed to minimize the sum of noise response power and the step edge response

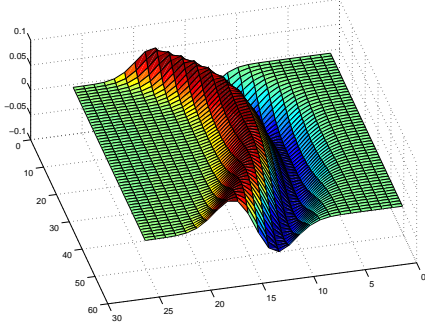


Figure 1. Shape filter: The shape is matched to a circular arc to detect the eye outline, and the cross-section is designed to detect the intensity change along the boundary.

error, was shown to be $g_\sigma(t) = \frac{1}{\sigma} \exp(-|t|/\sigma)$. Then the shape operator for a given shape region D with boundary contour C is defined by

$$h(\mathbf{x}) = g'_\sigma(l(\mathbf{x}))$$

where l is the distance function from C : $l(\mathbf{x}) = \pm \min_{\mathbf{z} \in C} \|\mathbf{x} - \mathbf{z}\|$, where the sign differs between the inside and the outside of C . Figure 1 shows a shape operator for a circular arc feature, matched to an eye outline or eyebrow in the head tracking problem.

The response of the local image s of an object to the operator h_ξ having geometric configuration ξ is

$$r^\xi = \int h_\xi(\mathbf{u})s(\mathbf{u})d\mathbf{u}$$

If we assume that the image is corrupted by noise $n(t)$, then the observation y^ξ is given by

$$y^\xi = \int h_\xi(\mathbf{u})s(\mathbf{u})d\mathbf{u} + \int h_\xi(\mathbf{u})n(\mathbf{u})d\mathbf{u} = r^\xi + \tilde{n}$$

where \tilde{n} is the noise response. Since we sample the observations y^ξ over the course of time, we denote the observation process by

$$Y_t = y_t^\xi = \int_0^t h(X_s)ds + V_t$$

We can assume without loss of generality that the observation noise is a standard Brownian motion V_t .

While the proposed method belongs to the family of feature-based motion computation methods, in that it relies on boundary gradient information, we do not use *detected* features. The gradient information is computed bottom-up from the raw intensity map using the shape filters. The boundary gradient information is retained for computing the

fit to the model shape. If we try to extract gradient features using edge detection, some of the boundary edge information may be missed due to thresholding. Some work makes use of wavelet bases [5] or blobs. While the set of basis filters used to approximate the intensity signatures of the features can give more flexibility in algebraic manipulation, a small number of basis filters cannot provide a close approximation to object shape. It is also hard to achieve a global description of an object shape.

3 The Zakai equation and the branching particle method

3.1 The Zakai equation

We start the formulation in a more general context to introduce the Zakai equation and the branching particle method. The state vector X_t representing the geometric parameters of an object is governed by the equation

$$dX_t = f(X_t)dt + \sigma(X_t)dW_t$$

Here W_t is a Brownian motion, and $\sigma = \sigma(X_t)$ models the state noise structure.

The tracking problem is solved if we can compute the state updates, given information from the observations. We are interested in estimating some statistic ϕ of the states, of the form

$$\pi_t(\phi) \triangleq E[\phi(X_t)|\mathcal{Y}_t]$$

given the observation history \mathcal{Y}_t up to t . Zakai et al. have shown that the unnormalized conditional density $p_t(\phi)$ satisfies a partial differential equation, usually called the Zakai equation:

$$dp_t(\phi) = p_t(A\phi)dt + p_t(h^*\phi)dY_t$$

Here A is a differential operator involving the state dynamics f and the state noise structure $\sigma(X_t)$ and dW_t .

3.2 The branching particle algorithm

It is known in nonlinear filtering theory [2] that the *unnormalized optimal filter* $p_t(\phi)$ is given by

$$\tilde{E} \left[\phi(X_t) \exp \left(\int_0^t h^*(X_s)dY_s - \frac{1}{2} \int_0^t h^*(X_s)h(X_s)ds \right) \middle| \mathcal{Y}_t \right]$$

where the expectation is taken with respect to the measure \tilde{P} which makes Y_t a Brownian motion (cf. [2]). Keeping this in mind, we will construct a sequence of branching particle systems U_n as in [6] which can be proved to approach the solution p_t : $\lim_{n \rightarrow \infty} U_n(t) = p_t$.

Let $\{U_n(t), \mathcal{F}_t; 0 \leq t \leq 1\}$ be a sequence of branching particle systems on $(\Omega, \mathcal{F}, \bar{P})$, the standard measure space on the state space.

Initial condition

0. $U_n(t)$ is the empirical measure of n particles of mass $\frac{1}{n}$, i.e., $U_n(t) = \frac{1}{n} \sum_{i=1}^n \delta_{x_i^n}$, where $x_i^n \in E$, for every i , $n \in \mathbf{N}$.

Evolution in the interval $[\frac{i}{n}, \frac{i+1}{n}]$, $i = 0, 1, \dots, n-1$

1. At time $\frac{i}{n}$, the process consists of the occupation measure of $m_n(\frac{i}{n})$ particles of mass $\frac{1}{n}$ ($m_n(t)$ denotes the number of particles alive at time t).

2. During the interval, the particles move independently with the same law as the signal X . Let $V(s)$, $s \in [\frac{i}{n}, \frac{i+1}{n}]$ be the trajectory of a generic particle during this interval.

3. At $t = \frac{i+1}{n}$, each particle branches into ξ_n^i particles with a mechanism depending on its trajectory in the interval. The mean number of offspring for a particle given the σ -field $\mathcal{F}_{\frac{i+1}{n}-} = \sigma(\mathcal{F}_s, s < \frac{i+1}{n})$ of events up to time $\frac{i+1}{n}$ is

$$E(\xi_n^i) = \exp\left(\int h^*(V(t))dY_t - \frac{1}{2} \int h^*h(V(t))dt\right) \quad (1)$$

so that the variance $\nu_n^i(V)$ is minimal, consistent with the number of offspring being an integer. The integrations are on the interval $[\frac{i}{n}, \frac{i+1}{n}]$.

The ‘observation likelihood’ term inside the exponential in (1) can be rearranged as

$$-\frac{1}{2} \int_0^t (h^* - dY_s^*)(h - dY_s) + \frac{1}{2} \int_0^t dY_s^* dY_s$$

The first term measures the disparity between the predicted and measured responses, which forces temporal invariance of the shape signature between the current and the previous frame. The second term is the response strength, representing how close the data is to the model shape in the current frame. We can compute the weights accurately without any loss of edge information, as explained in Section 2.

4 Time update of the state

Another contribution of the proposed method is the use of effective prediction and diffusion strategies. Step 2 of the algorithm requires that we have a particular dynamic function and error covariance matrix, which is not a realistic assumption. We only assume a second-order motion model, and recursively estimate the motion and diffusion parameters. We represent the dynamic equation as a discrete-time process: $X_{k+1} = X_k + d_k + \Sigma_k w_k$. w_k is a standard Gaussian random vector, and d_k is the displacement vector containing the velocity and acceleration parameters estimated using the preceding state estimates. d_k is further

refined by a random search step. The problem of updating states while estimating the motion parameters can be regarded as a system identification problem where the parameters are estimated recursively. In fact, [15] achieves better global stability of the EKF by adding an extra term in the Kalman gain computation. This term forces the state to be updated so that the prediction error with respect to these parameters is minimized. The proposed random search is closely analogous to this scheme in that it adjusts the displacement to ensure the maximum observation likelihood: $d_k = \arg \max_d \int h(\hat{x}_k + d)ds$. This seemingly trivial addition of a prediction adjustment is found to significantly increase stability.

Borrowing notation from the Kalman filter literature, the time update step yields the prior estimate of the state and the covariance matrix:

$$\begin{aligned} \hat{x}_{k+1}^- &= \hat{x}_k + d_k \\ \hat{P}_{k+1}^- &= \hat{P}_k + \Sigma_k \end{aligned} \quad (2)$$

Here \hat{x}_k and \hat{P}_k denote the posterior estimates after the measurement update (the application of the Kalman gain), which is equivalent to the observation and branching steps in the proposed algorithm. The *a priori* and *a posteriori* error covariance matrices are formally defined as

$$\begin{aligned} \hat{P}_k^- &= E[(\hat{x}_k^- - x_k)(\hat{x}_k^- - x_k)^T] \\ \hat{P}_k &= E[(\hat{x}_k - x_k)(\hat{x}_k - x_k)^T] \end{aligned} \quad (3)$$

These matrices are estimated by bootstrapping the particles x_k and the prior/posterior state estimates (\hat{x}_k^-, \hat{x}_k) into the above expressions. We use the error covariance estimated from the particles at time $k-1$ for the diffusion at time k by (2):

$$\hat{\Sigma}_k = \Sigma_{k-1} = \hat{P}_k^- - \hat{P}_{k-1}$$

since we can only compute (3) after the diffusion and the measurement update. The subtraction of the prior covariance matrix enforces only that the perturbation due to the diffusion be measured. If the particles are perturbed according to \hat{P}_k , they are bound to diverge because of the addition of unnecessary uncertainties at each step. $\hat{\Sigma}_k$ is positive semi-definite since $\hat{x}_k = E[x_k]$.

We observe that the diffusion matrix adapts to the motion. If the state vector moves fast in a certain direction, the prediction based on the previous estimates moves away from the correct value. The difference between the predicted distribution (\hat{P}^-) and the measured distribution (\hat{P}) then becomes larger, so that more diffusion is assigned to that direction. This characteristic of the diffusion method translates into an efficient search for the motion parameters. This property also helps the static(model) parameter values to stabilize. This stabilizing characteristic is observed in experiments, and will be explained in a later section.

5 Head tracking

5.1 Head model

We apply the algorithm presented above to the problem of head tracking. Tracking is guided by the intensity signatures of distinctive features of the face, such as eyes, eyebrows, and mouth. The head surface is approximated by an ellipsoid; the eyes and eyebrows are modeled by combinations of circular arcs, which are assumed to be ‘drawn’ on the face. Using these simple models of the head and face features, we are able to compute the expected feature signatures and corresponding shape operators.

We model the head as an ellipsoid in xyz space, with z being the camera axis:

$$\begin{aligned} E(x, y, z) &= E_{R_x, R_y, R_z, C_x, C_y, C_z}(x, y, z) \\ &\triangleq \frac{(x - C_x)^2}{R_x^2} + \frac{(y - C_y)^2}{R_y^2} + \frac{(z - C_z)^2}{R_z^2} = 1 \end{aligned}$$

We represent the pose of the head by three rotation angles $(\theta_x, \theta_y, \theta_z)$: θ_x and θ_z measure the rotation of the head axis \mathbf{n} , and the rotation of the head around \mathbf{n} is denoted simply by $\theta_y (= \theta_n)$. The center of rotation is assumed to be near the bottom of the ellipsoid, denoted by $a = (a_x, a_y, a_z)$, which is measured from (C_x, C_y, C_z) for convenience. Since the rotation of \mathbf{n} and the rotation around it are commutative, we can think of any change of head pose as rotation around the y axis, followed by ‘tilting’ of the axis. Let Q_x , Q_y , and Q_z be rotation matrices around x , y , and z , respectively. Let $p = (x, y, z)$ be any point on the ellipsoid $E_{R_x, R_y, R_z, C_x, C_y, C_z}(x, y, z)$. p moves to $p' = (x', y', z')$ under rotation Q_y followed by rotations Q_x and Q_z :

$$p' = Q_z Q_x Q_y (p - t - a) + a + t \quad (4)$$

The eyes are undoubtedly the most prominent features of a human face. The round curves made by the upper eyelid and the circular iris give unique signatures which are preserved under changes in illumination and facial expression. Features such as the eyebrows and mouth can also be utilized. The feature curves are approximated by circles or circular arcs on the ellipsoid. We parametrize the positions of these features by using the spherical coordinate system (*azimuth, altitude*) on the ellipsoid. A circle on the ellipsoid is given by the intersection of a sphere centered at a point on the ellipsoid with the ellipsoid itself. We typically used 22 parameters including 6 pose/location parameters. Figure 2 shows images with different head poses, generated using the 3D model of the head and the 2D facial feature models. Correctly tracked features are marked with the corresponding model shapes.

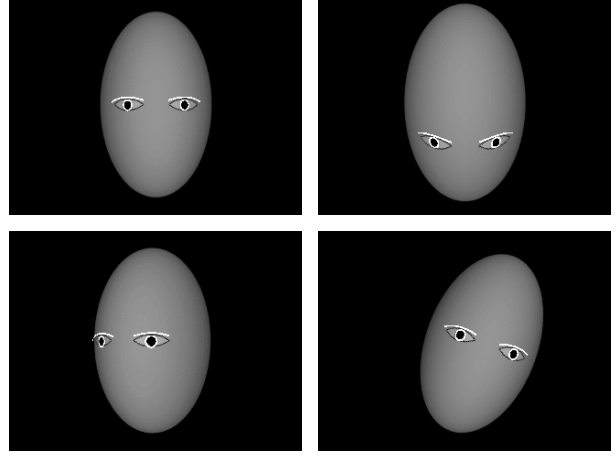


Figure 2. Three different head poses and tracked features. Upper right: rotation around x -axis, Lower left: rotation around y -axis, Lower right: rotation around z -axis.

5.2 Camera model and filter construction

We combine the head model and the camera model to compute the depth of each point on the face, so that we can compute the inverse projection and construct the corresponding operator. The center of perspective projection is $(0, 0, 0)$ and the image plane is $z = f$. Let $P = (X, Y)$ be the projection of $p' = (x', y', z')$ on the ellipsoid. These two points are related by

$$X/f = x'/z' \quad \text{and} \quad Y/f = y'/z' \quad (5)$$

Given $\xi = (C_x, C_y, C_z, \theta_x, \theta_y, \theta_z, \nu)$, the hypothetical geometric parameters of the head and feature (simply denoted by ν), we need to compute the inverse projection on the ellipsoid to construct the shape operator. Suppose the feature curve on the ellipsoid is the intersection (with the ellipsoid) of the circle $\| (x, y, z) - (e_x^\xi, e_y^\xi, e_z^\xi) \|^2 = R_e^{\xi^2}$ centered at $(e_x^\xi, e_y^\xi, e_z^\xi)$ which is on the ellipsoid. Let $P = (X, Y)$ be any point in the image. The inverse projection of P is the line defined by (5). The point (x', y', z') on the ellipsoid is computed by solving (5) combined with the quadratic equation $E_{R_x, R_y, R_z, C_x, C_y, C_z}(x, y, z) = 1$. This solution exists and is unique, since we seek the solution on the visible side of the ellipsoid. The point (x, y, z) on the reference ellipsoid $E_{0,0,0, C_x, C_y, C_z}(x, y, z) = 1$ is computed using the inverse operation of (4).

If we define the mapping from (X, Y) to (x, y, z) by $\rho(X, Y) \triangleq (x, y, z) \triangleq (\rho_x(X, Y), \rho_y(X, Y), \rho_z(X, Y))$ we can construct the shape filter as

$$h^\xi(X, Y) = h_\sigma(\| (\rho(X, Y) - (e_x^\xi, e_y^\xi, e_z^\xi) \|^2 - R_e^{\xi^2})$$

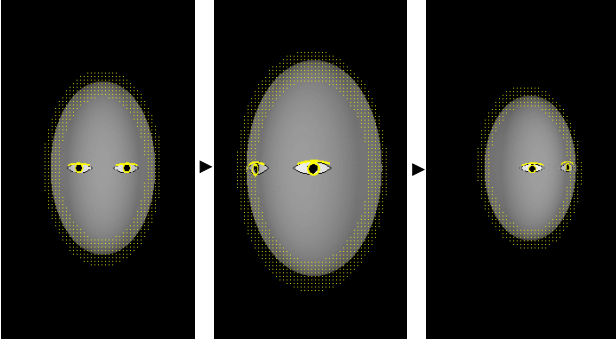


Figure 3. Sampled frames from a synthetic sequence. The head is moving back and forth (translation) while ‘shaking’ (rotation).

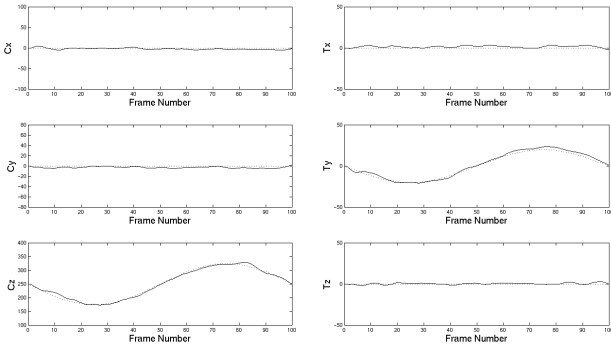


Figure 4. Estimated parameters for synthetic data. (Left column: translational motion, Right column: rotational motion.) The dotted lines are the real parameters used to generate the motion.

5.3 Experiments

The initial distribution is realized by uniformly sampling parameter vectors from a suitably chosen 22-dimensional cubic region in parameter space, and by thresholding them by shape filter responses. We used about 200 particles in most experiments, and observed that further increasing the number of particles did not make a noticeable difference in performance.

Experiments on synthetic data show good tracking of facial features and accurate head pose estimates, as shown in Figure 3. The head is ‘shaking’ while moving back and forth. The plots in Figure 4 compare the estimated translation and rotation parameters with the real values. We also tested many real human head motion sequences, and the algorithm achieved reliable tracking. Figure 5 shows an example, where the person repeatedly moves his head left and right, and the rotation of the head is naturally coupled with



Figure 5. Sampled frames with tracked features.

the translation. The principal motions are x -translation and y -rotation; a small y -translation and z -rotation are added, since the head motion is caused by the ‘swing’ of the upper body while sitting on a chair. Tracking and motion estimation would be easier if we only allowed rotation in which the axis of rotation is fixed around the bottom of the upper body. However, allowing all degrees of freedom yielded good performance. The plots of the estimated parameters are given in the left column of Figure 6(b). The principal motions (C_x, T_y, C_y, T_z) show coherent periodicity.

The contribution of the maximum observation likelihood prediction adjustment and the adaptive perturbation is verified as well. In Figure 6(a), ten instances of tracking results using different random number seeds are plotted. The first plot is the estimate of C_x obtained by applying fixed, empirically chosen diffusion parameters, and no prediction adjustment. The middle plot shows the same parameter estimated using the prediction adjustment only. The gain in stability is readily noticeable, as some instances in the first experiment showed unsuccessful tracking. The bottom plot demonstrates the effect of adaptive diffusion, in which the estimates show less variability than in the second experiment. Notice the consistency of the estimates at the end of the sequence. The contribution of adaptive diffusion is further illustrated in Figure 6(b), in which more parameters are

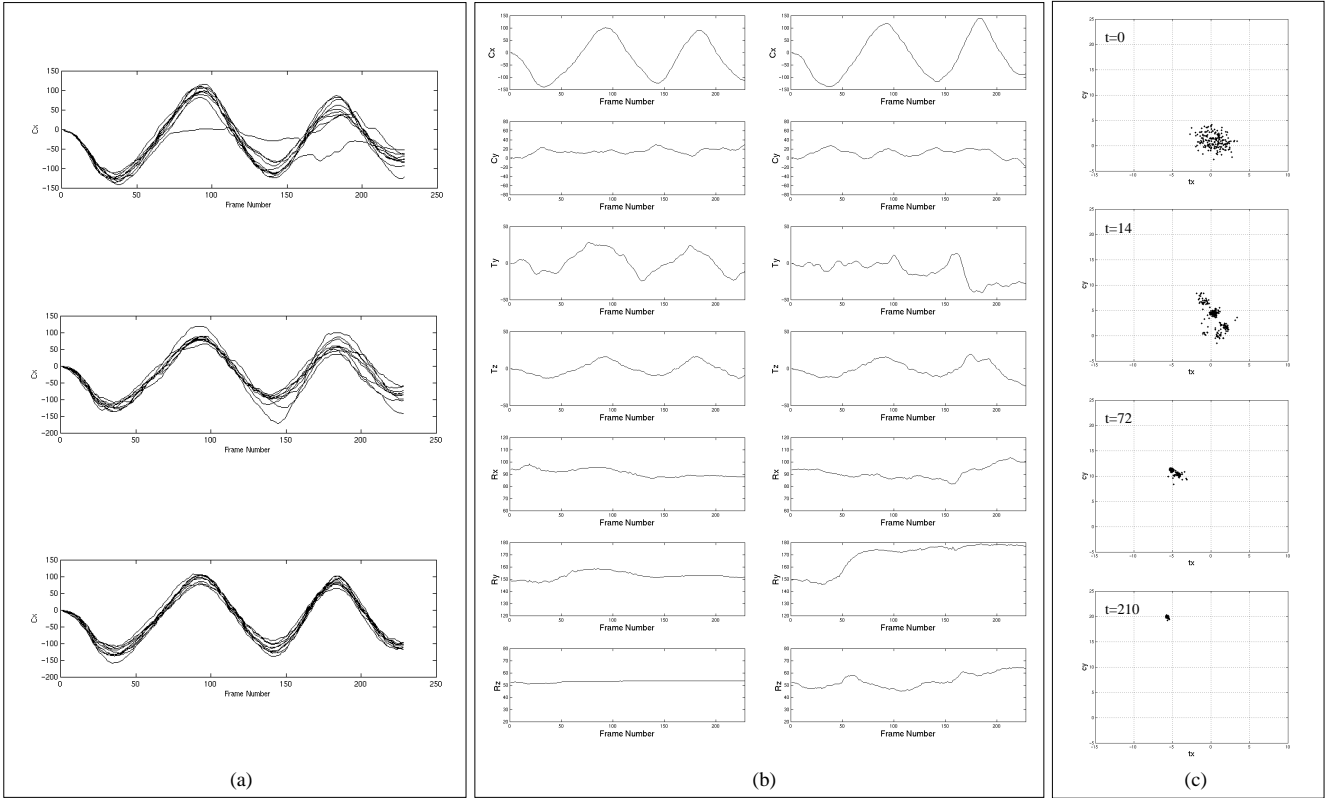


Figure 6. (a) Time update schemes. Top: no prediction adjustment, fixed diffusion, Middle: prediction adjustment only, Bottom: prediction adjustment and adaptive diffusion. (b) Diffusion schemes. Estimated location, pose and motion parameters using adaptive(left column) and fixed(right column) diffusions. (c) The spread of the particles shows the ambiguity of the translation and motion parameters. As the algorithm receives more data, the uncertainty decreases and is finally resolved.

compared. The estimates using fixed diffusion parameters are plotted in the right column. We can easily see that the estimates of the rotation parameters (T_y, T_z) are inferior. We also observed that the tracking is very sensitive to the diffusion parameter. Larger diffusion of the motion parameters helps in tracking fast motions, but unnecessary dispersion of inertial motion parameters often leads to divergence. Since the adaptive scheme determines the covariance matrix from the previous motion, we notice ‘delays’ when the head moves fast. Frames 2,4,5 in Figure 5 capture this effect. The adaptive scheme is more ‘cautious’ in exploring the parameter space, while the fixed diffusion method ‘ventures’ into parameter space using larger steps. The amount of diffusion in the case of the adaptive method is much smaller than in the case of a (working) fixed method.

The determination of model parameters is also observed in this figure. In the left column, the ellipsoid dimension parameters (R_x, R_y, R_z) eventually settle into stable values, while in the right column they remain highly variable. These model parameters are bound to be biased in the case of real data, since an ellipsoid cannot perfectly fit the hu-

man face. However, we suspect that stabilizing these values after enough information is provided would cause the other dynamic parameters to be assessed more reliably. When a temporally stabilized value cannot fit new data, the model gap causes an inaccurate prediction, and the consequently increased perturbation makes the parameter escape from a local maximum.

Since rotation and translation are being treated at the same time, there can be ambiguities between the two kinds of motion. For example, a small translation of the head in the vertical direction can be confused with a ‘nodding’ motion. Figure 6(c) depicts the ambiguity present in the same sequence by plotting the projections of particles onto the $T_x - C_y$ plane. At $t = 0$, the initial distribution shows the correlation between C_y and T_x . As more information is provided ($t = 14$), the particles show multi-modal concentrations. We observed that the concentration is dispersed when the motion is rapid, and shrinks when the head motion is close to one of the two ‘extreme’ points. The parameters eventually settle into a dominant configuration ($t = 72$ and $t = 210$).



Figure 7. Tracking of independently moving local features. Squinting and iris movement are captured and tracked, as well as the head movement.

6 Summary and future work

We have presented a method for tracking and estimating object motion using particle propagation and a 3D model of the object. The measurement update is done by particle branching according to the weights computed by shape-encoded filtering, and the shape constraint provides an ability to estimate the motion and model parameters. Time update is handled by minimizing the prediction error and adaptive diffusion, which contribute to global stability of the tracking and effective perturbation of the parameters. More complete analysis and possible improvements would be desirable to ensure the global optimization of model or ‘inertial’ parameters. Experiments show sensitivity of the tracker to the initial arrangement of particles, and the problem of determining the initial distribution needs to be addressed. As shown in Section 5.2, simple parametrization of the object surface and feature curves facilitates the construction of the shape operator, which helps to reduce computation. While we would achieve better fitting of the tracked features by using more realistic models, we obtained satisfactory results using simple models. Nevertheless, a more sophisticated (but efficient) parametrization would be desirable to achieve more accurate pose estimation and object shape computation. Figure 7 shows another example in which local feature motion is tracked in addition to global object motion; the motions of the irises and upper eyelids are more carefully tracked, so that squinting and gaze are recognized. The recognition of facial expression is a possible application of the proposed method. In our current work, only objects are allowed to move, and the camera is fixed. Extension of this work to the problem of simultane-

ous estimation of camera and object movement is another challenge to be addressed in future work.

References

- [1] A. Azarbayejani and A. P. Pentland, “Recursive Estimation of Motion, Structure, and Focal Length,” *PAMI*, Vol. 17, pp. 562-575, 1995.
- [2] A. Bensoussan, “Stochastic Control of Partially Observable Systems,” Cambridge University Press, 1992.
- [3] M.J. Black and A.D. Jepson, “EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation,” *IJCV*, Vol. 26, pp. 63-84, 1998.
- [4] T.J. Broida, S. Chandrashekar, and R. Chellappa, “Recursive 3-D Motion Estimation from a Monocular Image Sequence,” *AeroSys*, Vol. 26, pp. 639-656, 1990.
- [5] O. Chomat and J.L. Crowley, “Probabilistic Recognition of Activity Using Local Appearance,” *CVPR*, pp. 104-109, 1999.
- [6] D. Crisan, J. Gaines, and T. Lyons, “Convergence of a Branching Particle Method to the Solution of the Zakai Equation,” *SIAM J. Appl. Math.*, Vol. 58, pp. 1568-1590, 1998.
- [7] J. Deutscher, A. Blake, and I. Reid, “Articulated Body Motion Capture by Annealed Particle Filtering,” *CVPR*, (II) pp. 126-133, 2000.
- [8] D.B. Gennery, “Visual Tracking of Known Three-Dimensional Objects,” *IJCV*, Vol. 7, pp. 243-270, 1992.
- [9] Z.S. Haddad and S.R. Simanca, “Filtering Image Records Using Wavelets and the Zakai Equation,” *PAMI*, Vol. 17, pp. 1069-1078, 1995.
- [10] M. Isard and A. Blake “CONDENSATION – Conditional Density Propagation for Visual Tracking,” *IJCV*, Vol. 29, pp. 5-28, 1998.
- [11] G. Kitagawa, “Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models,” *J. Comp. and Graph. Stat.*, Vol. 5, pp. 1-25, 1996.
- [12] B. Li and R. Chellappa, “Simultaneous Tracking and Verification via Sequential Monte Carlo Method,” *CVPR*, (II) pp. 110-117, 2000.
- [13] A.J. Lipton, H. Fujiyoshi, and R.S. Patil, “Moving Target Classification and Tracking from Real Time Video,” *DARPA Image Understanding Workshop*, pp. 129-136, 1998.
- [14] J. Liu and R. Chen, “Sequential Monte Carlo Methods for Dynamic Systems,” *J. Amer. Statist. Assoc.*, Vol 93, pp. 1032-1044, 1998.
- [15] L. Ljung, “Asymptotic Behaviour of the Extended Kalman Filter as a Parameter Estimator for Linear Systems,” *Automatic Control*, Vol. 24, pp. 36-50, 1979.
- [16] H. Moon, R. Chellappa, and A. Rosenfeld, “Optimal Shape Detection,” *ICIP*, 2000.
- [17] J. Sullivan, A. Blake, M. Isard, and J. MacCormick, “Object Localization by Bayesian Correlation,” *ICCV*, pp. 1068-1075, 1999.
- [18] M. Zakai, “On the Optimal Filtering of Diffusion Processes,” *Z. Wahrscheinlichkeitstheorie verw. Gebiete*, Vol. 11, pp. 230-243, 1969.