Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2008, Article ID 596989, 16 pages
doi:10.1155/2008/596989

Research Article

3D Shape-Encoded Particle Filter for Object Tracking and Its Application to Human Body Tracking

H. Moon[1] and R. Chellappa[2]

[1] *VideoMining Corporation, 403 South Allen Street, Suite 101, State College, PA 16801, USA*
[2] *Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742, USA*

Correspondence should be addressed to H. Moon, hmoon@videomining.com

Received 1 February 2007; Revised 14 July 2007; Accepted 25 November 2007

Recommended by Maja Pantic

We present a nonlinear state estimation approach using particle filters, for tracking objects whose approximate 3D shapes are known. The unnormalized conditional density for the solution to the nonlinear filtering problem leads to the Zakai equation, and is realized by the weights of the particles. The weight of a particle represents its geometric and temporal fit, which is computed bottom-up from the raw image using a shape-encoded filter. The main contribution of the paper is the design of smoothing filters for feature extraction combined with the adoption of unnormalized conditional density weights. The "shape filter" has the overall form of the predicted 2D projection of the 3D model, while the cross-section of the filter is designed to collect the gradient responses along the shape. The 3D-model-based representation is designed to emphasize the changes in 2D object shape due to motion, while de-emphasizing the variations due to lighting and other imaging conditions. We have found that the set of sparse measurements using a relatively small number of particles is able to approximate the high-dimensional state distribution very effectively. As a measure to stabilize the tracking, the amount of random diffusion is effectively adjusted using a Kalman updating of the covariance matrix. For a complex problem of human body tracking, we have successfully employed constraints derived from joint angles and walking motion.

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 such examples. A human body can also be decomposed into simple shapes. For tracking or recognition of human activities, appearance features are often too variable, and local features are noisy and not reliable for establishing temporal correspondences. Shape constraints also provide strong clues about object pose while the object is moving. "Shape" in this context refers to persistent geometric image signature, such as ellipsoidal human head boundary, parallel lines for the boundary of limbs, or facial features.

We model a human body using simple quadratic solids; the 2D projection of the solids constitutes the "shapes" to be tracked. The image gradient signature of a shape is modeled using the optimal shape operator that was introduced in [1]. The adoption of quadratic solids for modeling parts facilitates the computation of the shape operator. The responses of an image frame to a set of shape operators having certain ranges of pose and size parameters are used as observations in a nonlinear state space formulation, to guide object tracking and motion estimation. The magnitudes of the responses are accurate and robust to noise, and they enable reliable estimation 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 (EKFs) [2, 3]. A recursive version of Monte Carlo simulation (called sequential Monte Carlo or particle filtering) has become popular for tracking and motion computation problems. Mainly due to advances in computing power, applications to the state estimation problem [4, 5] have been proposed in the statistics community. Reference [6] introduced the *condensation* algorithm for tracking, and [7, 8] further refined the method by using layered sampling for accurate object localization and effective search for the state parameters. Reference [9] used the framework of sequential importance sampling [5] to solve the problem of simultaneous object tracking and verification. Reference [10] also employed particle filtering for the 3D tracking of a walking person.

In our approach, the functional relation between the geometric parameter space and the image space makes the observation process highly nonlinear. There is a generalization of the Kalman filter to the nonlinear framework, by Zakai [11]. They derived an equation that incorporates both dynamic and observation equations, and which, if solved, enables the temporal propagation of the probability of the states conditioned on the observations. Reference [12] introduced the Zakai equation to image analysis problems.

As derived in filtering theory, the unnormalized conditional density is a solution to the Zakai equation. The solution is in general not available in a closed form; we employ a branching particle method to solve the filtering problem. The system of particles that simulates the conditional density of states is found [13] to converge to the target distribution. The proposed measurement process—shape filter response—contributes to the accurate computation of the weights. We also have a unique way of computing the unnormalized conditional density used for computing the weights, that takes into account both *geometric fit* of the data and *temporal coherence* of the motion. The method of estimating the number of offsprings using randomized sampling is also designed to be optimal, while the total number of samples is fixed in resampling approaches. It has been shown in [14] that the particle method is superior to the resampling method in terms of large sample behavior.

After branching, the particles follow the system dynamics plus random perturbation. As we cannot assume any particular motion model in most applications, we employ an approximate second-order motion prediction. The prediction is modified by a random search to minimize the prediction error. The amount of random diffusion has to be determined, which we found to be 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 they are usually very small; nevertheless, this method of computing the diffusion shows noticeable improvements in tracking and pose estimation.

We first applied this method of shape tracking to the problem of human head tracking, and later to the full body tracking in a monocular video sequence. For head tracking, the head is modeled as a 3D 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 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. In addition, the three parameters representing the size of the ellipsoid are free, along with the distance from the ellipsoid to the camera. The proposed algorithm simultaneously estimates the size, pose, and location (up to scale) of the ellipsoid.

We also extended our application to full body tracking of a walking person when the person is walking approximately perpendicular to the camera axis. The body is modeled as being composed of simple geometric surfaces: ellipsoid for the head and truncated cones for the limbs. We also have added texture information of the parts in addition to the shape. We have found that the addition of texture cue helps the tracking in a meaningful way. The kinematic model of the body constrains the pose of the body within physically possible range, which also limits the search space for tracking. The full body tracking is a very hard problem due to complex motion, high dimensionality, and self-occlusion. While the proposed method cannot completely solve the problem, we have found that the constraint provided by the shape and texture cues, the employment of a smoothing filter to extract reliable features, and the adoption of weight function derived from filtering theory make the tracking of walking person more manageable.

We first introduce a representation based on quadratic surfaces to compute the shape operator (Section 2). In Section 3, the tracking of a human is formulated as a nonlinear filtering problem. The subsections cover the details of branching particle method. Section 4 presents the application of human head tracking. The tracking of human walking motion is detailed in Section 5.

2. SHAPE AND MEASUREMENTS

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

On the other hand, methods for appearance-based tracking using a linear subspace representation [15] or an object template [16] 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 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 addition of learned representation of object images will provide a powerful edge to object tracking, as shown in [17]. The proposed work, however, ventures to improve object tracking from the model-based front.

When we have a geometric model of the shape of a solid object, or an articulated kinematic model of a structured object, we can manipulate it to fit the motion of the model to a 2D object in video frames using any prediction method (e.g., a Kalman filter). The model and the scene are usually compared using edge features. Reference [18] deals with the problem of tracking objects with known 3D shapes. Reference [19] describes a comprehensive framework for tracking using 3D deformable model and optical flow computation using 3D shape constraints, and presents an application to face tracking. Reference [20] shows how the dynamical shape priors, represented using level sets, provide strong constraints for segmenting/tracking deformable shapes under severe noise and occlusions. Shape constraints provide tighter constraints on object configuration than point features do; the deformation of a shape due to changes in object pose or camera parameters (e.g., focal length) provides better clues about these parameters, while local point features (e.g., end-points, vertices, junctions), often cannot. We have observed that shape constraints, being global, effectively stabilize tracking when the tracking deviates from the correct course after a rapid motion.

We make use of 3D shape model, combined with the boundary gradient information extracted using this model, to track body motion. Given the predicted size, position, and pose of the body parts, the projection of the model is compared to the image using the set of shape filters. Using the optimal shape detection and localization technique derived in [1], the responses of the shape operators provide the tracker with an accurate geometrical fit of the model to the data, and a strong temporal correspondence between frames.

We now briefly introduce the image operator we use for measuring the model fit. We then introduce how the use of 3D solids facilitates the construction of shape filter for two kinds of shape cues: the body silhouette and facial features. The body tracking makes use of boundary shape, while head tracking is accomplished using the positions and shapes of facial features.

2.1. Shape filters to measure shape match

In [1], the optimal one-dimensional smoothing operator, designed to minimize the sum of noise response power and step edge response error, was shown to be $g_\sigma(s) = (1/\sigma)\exp(-|s|/\sigma)$. Then the shape operator for an arbitrary shape region D, whose boundary is the shape contour C, is defined by

$$G(\mathbf{x}) = g'_\sigma(l(\mathbf{x})), \tag{1}$$

where the level function l can be implemented by

$$l(\mathbf{x}) = \begin{cases} +\min\limits_{\mathbf{z}\in C}\|\mathbf{x}-\mathbf{z}\| & \text{for } \mathbf{x}\in D, \\ -\min\limits_{\mathbf{z}\in C}\|\mathbf{x}-\mathbf{z}\| & \text{for } \mathbf{x}\in D^c. \end{cases} \tag{2}$$

The level function l simply takes the role of supplying the distance function to the shape contour C. l can have a regular parametric form (e.g., quadratic), when the shape contour C is a parametric curve. Figure 4 shows a shape operator for a circular arc feature, matched to an eye outline or eyebrow in the head tracking problem. The operator is designed to achieve good shape detection and localization performance. The detection performance is equivalent to the accuracy of the filter response, while the localization performance is closely related to the recognition/discrimination of the shape.

2.2. 3D body and head model

The 3D model of the body consists of truncated cones (trunk and limbs) and an ellipsoid (head). The body contour shape is represented by the distance function around the contour; the equation is derived by combining the quadratic equation for the solids and the perspective projection equation.

Let the 3D geometry of a body part be approximated by a quadratic surface parameterized by $(pose, size) = \xi$:

$$M_\xi(\mathbf{p}) = M_\xi(x, y, z) = 0, \tag{3}$$

where $\mathbf{p} = (x, y, z)$ is any point on the solid. Note that throughout the paper $M = M_\xi$ will denote both the quadratic equation that defines the surface and the surface itself. The image plane coordinates $\mathbf{P} = (X, Y)$ of the projection of $\mathbf{p}$ are computed using $X/f = x/z$ and $Y/f = y/z$, where f is the focal length. We construct the shape operator of the projection of M_ξ. Given a point $\mathbf{P} = (X, Y)$ in the image plane, let the corresponding point on M_ξ be (x, y, z). Then we have a quadratic equation with respect to depth z:

$$M_\xi\left(\frac{X}{f}z, \frac{Y}{f}z, z\right) \triangleq a_{X,Y,f}z^2 + b_{X,Y,f}z + c_{X,Y,f} = 0, \tag{4}$$

where $a = a_{X,Y,f}$, $b = b_{X,Y,f}$, $c = c_{X,Y,f}$ are constants that depend on X, Y, f. The distance from (X, Y) to the boundary contour of the projection of M_ξ is approximated by the determinant

$$d(\mathbf{P}) = d_f(x, y) = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \tag{5}$$

assuming that (X, Y) is close to the boundary contour. The shape operator for a given shape region D is then defined by

$$G(\mathbf{P}) = g'_\sigma(d(\mathbf{P})), \tag{6}$$

where g'_σ is as defined in Section 2.1.

2.3. Facial feature model

Head 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 (Figure 1); the eyes and eyebrows are modeled by combinations of circular arcs, which are assumed to be drawn on the ellipsoid (Figure 2). Using these simple models of the head and facial features, we are able to compute the expected feature signatures and corresponding shape operators.

2.3.1. Ellipsoidal head model

We provide a detailed description of the 3D representation of facial features, which will also serve as an example of the formulation laid out in the previous section. We model the head as an ellipsoid in xyz space, with z being the camera axis:

$$\begin{aligned} M_\xi(x, y, z) &= M_{R_x,R_y,R_z,C_x,C_y,C_z}(x, y, z) \\ &\triangleq \frac{(x-C_x)^2}{R_x^2} + \frac{(y-C_y)^2}{R_y^2} + \frac{(z-C_z)^2}{R_z^2} - 1. \end{aligned} \tag{7}$$

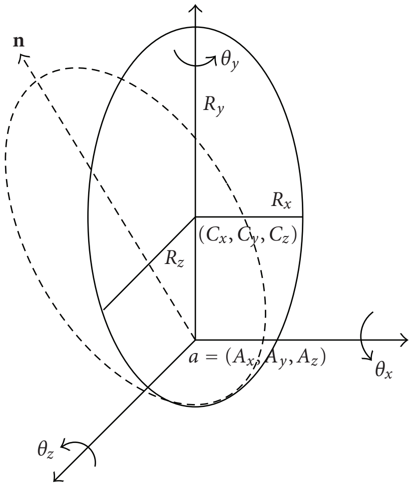

Figure 1: Rotational motion model of the head.

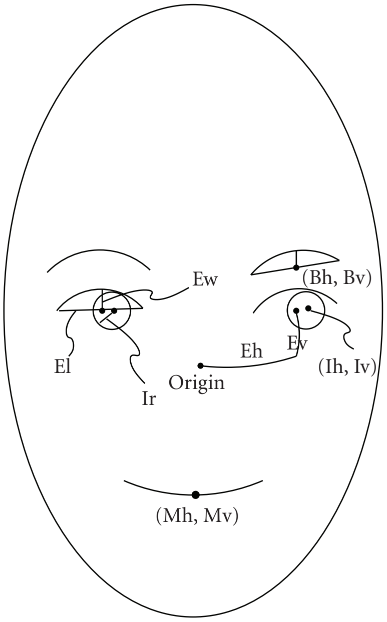

Figure 2: Ellipsoidal head model and the parameterization of facial features.

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 **n**, and the rotation of the head around **n** is denoted by $\theta_y (= \theta_{\mathrm{n}})$. The center of rotation is assumed to be near the bottom of the ellipsoid (corresponding to the rotation around the neck), 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 **n** and the rotation of the head 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 axes, respectively. Let $p = (x, y, z)$ be any point on the ellipsoid $M_{R_x,R_y,R_z,C_x,C_y,C_z}(x, y, z)$. p moves to $p' = (x', y', z')$ under rotation Q_y followed by rotations Q_x and Q_z:

$$p' = Q_z \, Q_x \, Q_y \, (p - b - a) + a + b. \tag{8}$$

Note that $b = b_{(C_x,C_y,C_z)} = (C_x, C_y, C_z)$ represents the position of the ellipsoid before the rotation.

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. Circles or circular arcs on the ellipsoid approximate these feature curves. We parameterize 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, which include 6 pose/position parameters.

2.3.2. 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. Figure 3 illustrates the scheme. 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

$$\frac{X}{f} = \frac{x'}{z'}, \qquad \frac{Y}{f} = \frac{y'}{z'}. \tag{9}$$

Given $\xi = (C_x, C_y, C_z, \theta_x, \theta_y, \theta_z, \nu)$, the geometric parameters of the head and features (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 $\|(x, y, z) - (e^\xi_x, e^\xi_y, e^\xi_z)\|^2 = R^{\xi 2}_e$ centered at $(e^\xi_x, e^\xi_y, e^\xi_z)$ (which is also on the surface). Let $P = (X, Y)$ be any point in the image. The inverse projection of P is the line defined by (9). The point (x', y', z') on the ellipsoid is computed by solving (9) along with the quadratic equation $M_{R_x,R_y,R_z,C_x,C_y,C_z}(x, y, z) = 0$. 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 $M_{0,0,0,C_x,C_y,C_z}(x, y, z) = 0$ is computed using the inverse operation of (7).

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

$$G^\xi(X, Y) = g'_\sigma\Big(\big\|(\rho(X, Y) - (e^\xi_x, e^\xi_y, e^\xi_z)\big\|^2 - R^{\xi 2}_e\Big). \tag{10}$$

Note that the expression inside g'_σ represents the displacement from (X, Y) to the feature contour; it defines the level function l of the circular (arc) feature contour (refer to Section 2.1).

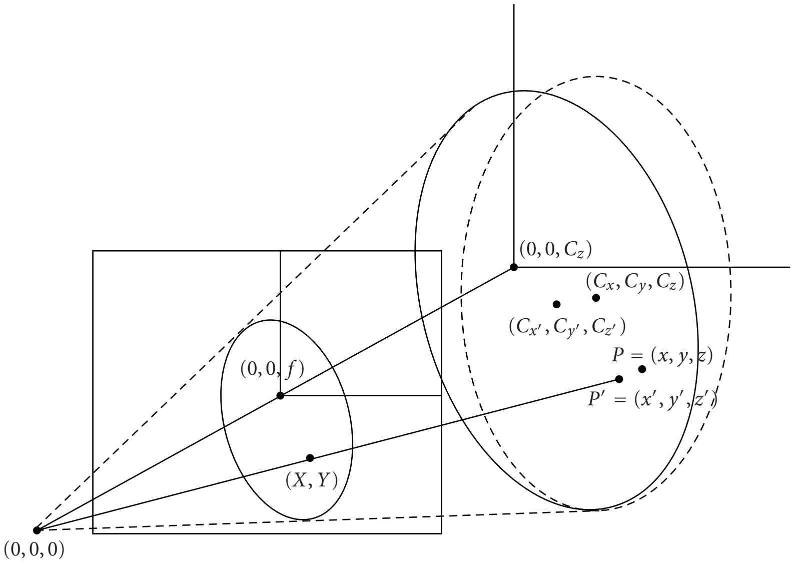

Figure 3: Perspective projection model of the camera.

2.4. The measurement equation

The response of the local image I to the shape operator G_α that represents an object having geometric configuration α is

$$r^\alpha = \int G_\alpha(\mathbf{u})I(\mathbf{u})d\mathbf{u}. \tag{11}$$

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

$$y^\alpha = \int G_\alpha(\mathbf{u})I(\mathbf{u})d\mathbf{u} + \int G_\alpha(\mathbf{u})n(\mathbf{u})d\mathbf{u} = r^\alpha + \tilde{n}, \tag{12}$$

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

$$Y_t = \int_0^t h(\alpha_s)ds + V_t, \tag{13}$$

where we have defined $h(\alpha_s) \triangleq r^{\alpha_s}$.

We assume that the observation noise is a standard Brownian motion V_t. The observation noise, though correlated in the spatial dimension, is independent in the temporal dimension. Since the noise structure of $\tilde{n}$ is homogeneous with respect to geometric parameters, we can assume 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 an edge detector, some of the boundary edge information may be lost due to thresholding. The total edge strength from thresholded contour pixels after edge detection should fluctuate much more than the response to convolution with a global operator. On the other hand, the support of the filter is thin around the shape contour (Figure 4); the filter is designed to emphasize the local changes of 2D object shape due to motion, while de-emphasizing variations due to lighting and other imaging conditions, thereby providing a compact and efficient representation of the shape of the object. Past work has made use of wavelet bases [21] 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 generic filters cannot provide a close approximation to object shape. It is also hard to achieve a global description of an object shape.

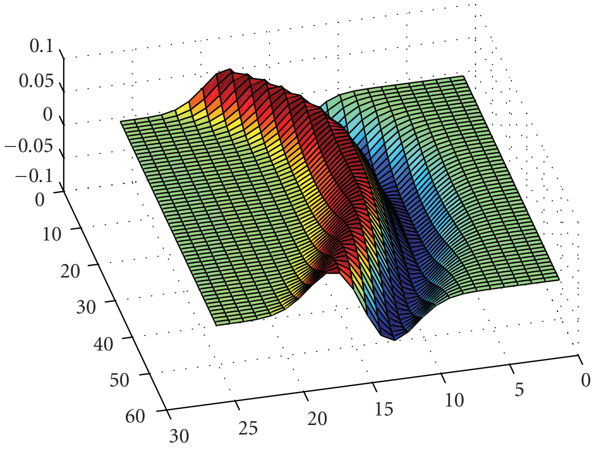

Figure 4: Shape filter: the shape is matched to a circular arc to detect the eye outline, and the cross-section is designed to detect the intensity change along the boundary.

The shape filter can be constructed for arbitrary contours, so that more accurate fitting can be carried out.

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 $X_t \in \Omega$ representing the geometric parameters of an object is governed by the equation

$$dX_t = f(X_t)dt + \sigma(X_t)dW_t. \tag{14}$$

Here W_t is a Brownian motion, and $\sigma = \sigma(X_t)$ models the state noise structure in a standard (probability) measure space $(\Omega, \mathcal{F}, \widetilde{P})$. Since we will not be using any linearization in the computation, the transfer function f can have a very general form. The state vector should be of the form $X_t = (\alpha_t, \beta_t)$, where α_t is the vector representing the geometry (position, pose, etc.) of the object and β_t is the motion parameter vector.

The tracking problem is solved if we can compute the state updates, given information from the observations in (10). We are interested in estimating some statistic ϕ of the states, of the form

$$\pi_t(\phi) \overset{\Delta}{=} E[\phi(X_t) \mid \mathcal{Y}_t] \tag{15}$$

given the observation history $\mathcal{Y}_t$ up to t. Zakai [11] has shown that the unnormalized conditional density $p_t(\phi)$ satisfies a partial differential equation, usually called the Zakai equation:

$$dp_t(\phi) = p_t(A\phi)dt + p_t(h^*\phi)dY_t. \tag{16}$$

Here A is a differential operator involving the state dynamics f and the state noise structures $\sigma(X_t)$ and dW_t. Note that the equation is equivalent to the pair of state equation (14) and observation equation (10).

3.2. The branching particle algorithm

It is known in nonlinear filtering theory [22] that the *unnormalized optimal filter* $p_t(\phi)$, which is a solution to (16), is given by

$$\widetilde{E}\left[\phi(X_t)\exp\left(\int_0^t h^*(X_s)dY_x - \frac{1}{2}\int_0^t h^*(X_s)h(X_s)ds\right)\middle|\, \mathcal{Y}_t\right], \tag{17}$$

where the expectation is taken with respect to the measure $\widetilde{P}$ which makes Y_t a Brownian motion (cf. [22]). This equation is merely a formal expression, because one needs to evaluate the integration $\widetilde{E}[\cdot \mid \mathcal{Y}_t]$ with respect to the measure $\widetilde{P}$. However, this equation provides a recursive relation to derive a numerical solution; we will construct a sequence of branching particle systems U_n as in [13] which can be proved to approach the solution p_t, that is, $\lim_{n\to\infty} U_n(t) = p_t$.

Let $\{U_n(t), \mathcal{F}_t;\ 0 \le t \le 1\}$ be a sequence of branching particle systems on $(\Omega, \mathcal{F}, \widetilde{P})$.

Initial condition

(0) $U_n(t)$ is the empirical measure of n particles of mass $1/n$, that is, $U_n(t) = (1/n)\sum_{i=1}^n \delta_{x_i^n}$, where $x_i^n \in E$, for every $i, n \in \mathbf{N}$, and $\delta_{x_i^n}(x)$ is a delta function centered at x_i^n.

Evolution in the interval $[i/n, (i+1)/n]$, $i = 0, 1, \dots, n-1$

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

(2) During the interval, the particles move independently with the same law as in the system dynamics equation (14). Let $Z(s), s \in [i/n, (i+1)/n)$, be the trajectory of a generic particle during this interval.

(3) At $t = (i+1)/n$, each particle branches into ξ_n^i particles with a mechanism depending on its trajectory in the interval. The mean number of offsprings for a particle is

$$\mu_n^i = E(\xi_n^i) = \exp\left(\int h^*(Z(t))dY_t - \frac{1}{2}\int h^*h(Z(t))dt\right) \tag{18}$$

so that the variance $\nu_n^i(V)$ is minimal, where the variance occurs due to the off-rounding of $\nu_n^i(V)$ to compute the integer value ξ_n^i. The symbol $*$ represents complex conjugate (transpose for the real-valued case) here and throughout the paper. More specifically, we determine the number ξ_n^i of offsprings by

$$\xi_n^i = \begin{cases} [\mu_n^i] & \text{with probability } \mu_n^i - [\mu_n^i], \\ [\mu_n^i] + 1 & \text{with probability } 1 - \mu_n^i + [\mu_n^i], \end{cases} \tag{19}$$

where [] is the rounding operator.

Note that the integrals in (18) are along the path of the particles $Z(t)$. In the proposed visual tracking application, we only apply the branching mechanism only once per observation interval (between image frames). We take advantage of the branching particle method in two aspects: the recursive unnormalized conditional density filter (its implementation is described in Section 3.4) and the minimum variance branching scheme.

3.3. Time update of the state

Another feature of the proposed method is the use of effective prediction and diffusion strategies. Step 2 of the algorithm is based on an unrealistic assumption that we have a particular state transition function and an error covariance matrix. We only assume a second-order motion model, and recursively estimate the motion and diffusion parameters. We represent the dynamical equation as a discrete-time process: $X_{k+1} = X_k + d_k + \Sigma_k w_k$, where 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 reduces to one of recursively estimating the motion parameters using a system identification technique. In fact, [23] 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: $d_k = \arg\max_d \int h(\hat{x}_k + d)ds$.

The random search is performed by first generating a number of particles around the predicted state, according to a Gaussian distribution. The spread of the Gaussian distribution is empirically determined. Then the shape fitness (the response to the corresponding shape operator) of each particle is computed. The particle having the maximum fitness is chosen as the adjusted predicted state. This scheme is different from the original particle process, in that the particles for random search are used once in the given cycle and discarded. The particle fitness is simply the shape filter response, not the filtering weight (the unnormalized conditional density).

The original weight equation is supposed to adjust the weights of the sampled particles (diffused around the predicted state) based on the observation. However, if the prediction is off by too much (e.g., when the prediction falls at the tail of the true distribution), it introduces significant bias. The original branching particle framework suggests applying the branching mechanism multiple times within the observation interval, though it would be too costly to implement. The prediction adjustment can also be seen as a cheaper alternative to achieve the same goal. This seemingly simple 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:

$$\begin{aligned} \hat{x}_{k+1}^- &= \hat{x}_k + d_k, \\ \hat{P}_{k+1}^- &= \hat{P}_k + \Sigma_k. \end{aligned} \tag{20}$$

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 branching particle algorithm. The *a priori* and *a posteriori* error covariance matrices are formally defined as

$$\begin{aligned} \hat{P}_k^- &= E[(\hat{x}_k^- - x_k)(\hat{x}_k^- - x_k)^T], \\ \hat{P}_k &= E[(\hat{x}_k - x_k)(\hat{x}_k - x_k)^T]. \end{aligned} \tag{21}$$

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 (20):

$$\hat{\Sigma}_k = \Sigma_{k-1} = \hat{P}_k^- - \hat{P}_{k-1} \tag{22}$$

since we can only compute (21) after the diffusion and the measurement update. The subtraction of the prior covariance matrix ensures that the perturbation due to the diffusion is measured. If the particles are perturbed according to $\hat{P}_k$, they are bound to divergence because of the addition of unnecessary uncertainties at each step. $\hat{\Sigma}_k$ is positive semidefinite since $\hat{x}_k = E[x_k]$.

We have observed 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 $(\hat{P})$ becomes large, 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. Many of the geometric parameters of the object model are initially chosen by crude guesses, and they are adjusted as more information comes in. Since the amount of perturbation is tuned according to the goodness of fit, the parameter value eventually settles down. If a stabilized value turns out to be inaccurate as the pose changes, more perturbation due to the mismatch causes the parameter to escape from a local maximum and wander around looking for a better value. This stabilizing characteristic is observed in experiments, and will be explained in a later section.

An alternative way of handling the state prediction is to include the velocity parameters into the state vector and propagate them with model and pose parameters. We found that estimating the dynamic parameters using the prior estimates of the states gives much better performance, in the applications studied here. The increased dimensionality is one of the possible causes, and one can also suspect that this is due to the extra degree of randomness caused by perturbing the velocity parameter.

3.4. Measurement update

The "observation likelihood" term inside the exponential in (18) can be rearranged as

$$-\frac{1}{2}\int_0^t (h^* - dY_s^*)(h - dY_s) + \frac{1}{2}\int_0^t dY_s^* dY_s. \tag{23}$$

The first term measures the disparity between the predicted and measured responses, which forces temporal invariance of the shape signature between the current and previous frames. 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.

The observation function h is not usually available in visual tracking problems since the functional relation between the state x and the measurement Y is not well defined due to the scene variations—the gap between the model and the real object image—and other environmental features such as background clutter and illumination. While these factors are hard to model, we only assume that they are constant between frames. We bootstrap the measured values from the previous frame to obtain the expected measurements for the current frame. That is, if we use the discrete-time notation H

for h and R for Y, we compute the unnormalized conditional density (18) by

$$\exp\left(H_{x_k}R_{x_k}-\frac{1}{2}H_{x_k}^2\right)=\exp\left(R_{x_{k-1}}R_{x_k}-\frac{1}{2}R_{x_{k-1}}^2\right) \quad (24)$$

by replacing H_{x_k} with $R_{x_{k-1}}$. We have found that this unique way of computing the unnormalized conditional density is essential for propagating the posterior density. We experimented with other ad hoc expressions for computing the weights by trying many combinations of terms in the above equation; they were all unsuccessful.

Figure 5 illustrates how the particles are processed at each stage of the branching particle algorithm. The sizes of the dots represent the weights, and the dominant particles are marked with white dots, which yield more offsprings after branching than the other "weaker" particles. The values of the state vectors are preserved until the last stage where the state vectors go through a uniform displacement and a random perturbation.

4. HEAD TRACKING

We have first applied the proposed method to the problem of 3D head tracking. There have been successful appearance-based tracking algorithms [24, 25], using texture mapping on cylindrical head model. We use feature shape information—global arrangement and local shapes of facial features to guide tracking. The set of shape filters constructed from the 3D head and facial feature model (Section 2) is used to extract image features. The problem is relatively manageable because the head pose change is almost rigid; one only needs to take into account the local deformation due to facial expression.

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 6. The head is "shaking" while moving back and forth. The plots in Figure 7 compare the estimated translation and rotation parameters with real values.

We have tested many human head motion sequences, and the algorithm achieved reliable tracking. Figure 8 shows an example, where the person repeatedly moves his head left and right, and the rotation of the head is naturally coupled with translation. The principal motions are x-translation and y-rotation; 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 9(b). The global motion (C_x, T_y, C_y, T_z) shows coherent periodicity.

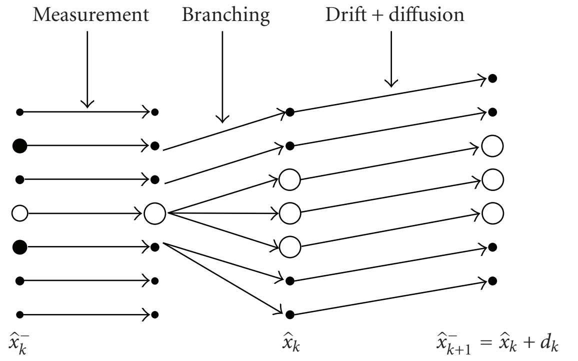

FIGURE 5: Schematic diagram of branching particle method.

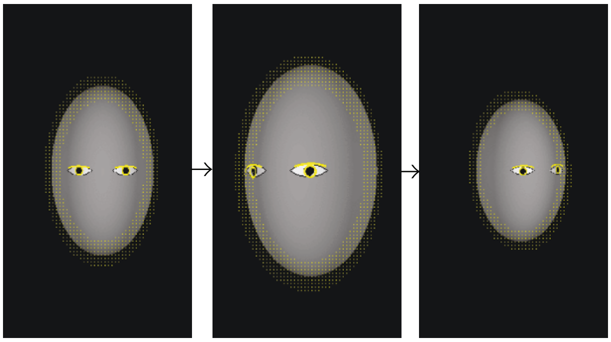

FIGURE 6: Sampled frames from a synthetic sequence. The head is moving back and forth (translation) while "shaking" (rotation). The estimated head pose and location and the facial features are marked.

The contributions of the maximum observation likelihood prediction adjustment and the adaptive perturbation are verified as well. In Figure 9(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 parameters estimated using prediction adjustment only. The gain in stability is readily noticeable, as some of the instances in the first experiment resulted in unsuccessful tracking. The bottom plot demonstrates the effect of adaptive diffusion; 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 9(b), in which more parameters are 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 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, and 5 in Figure 8 capture this effect. The adaptive scheme is

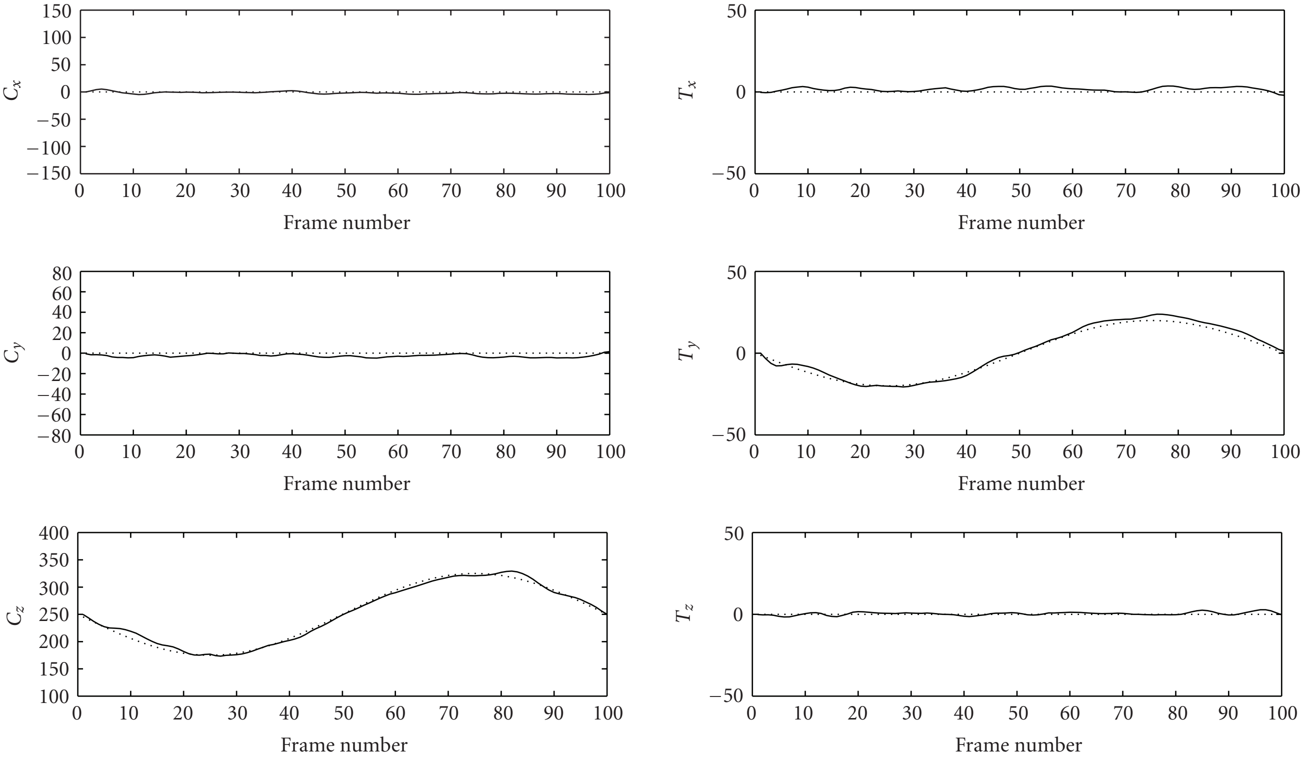

FIGURE 7: Estimated parameters for synthetic data (left column: translational motion; right column: rotational motion). The dotted lines are the real parameters used to generate the motion.

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 estimates of model parameters are also shown 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 human 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 modeling errors cause inaccurate prediction, and the resulting increase in perturbation makes the parameter escape from a local maximum. This process of searching for an optimal value of a model parameter can be thought of as stochastic hill-climbing; a more involved analysis would be desirable.

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 9(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 multimodal concentrations. We observed that the concentration is dispersed when the motion is rapid, and it shrinks when the motion is close to one of the two "extreme" points. The parameters eventually settle into a dominant configuration ($t = 72$, $t = 210$).

We have tested the algorithm on an image sequence where the face is momentarily occluded by a waving hand. Figure 11 shows both successful and failed results. In the second column, only the facial feature filters were used for computing the response. The tracker deviates from the correct facial position due to the strong gradient response from the fingers boundary, and it fails to recover despite the shape constraints matched to the facial features. In the first column, we have employed the head boundary shape filter. The tracker initially deviates from the correct position (the third frame), but recovers after a few frames. The extra ellipsoidal filter matched to the head boundary adds to the computation, but greatly helps to achieve robustness to partial occlusion. We have observed that the head shape filer did not improve nonoccluding sequences.

5. TRACKING OF WALKING

The task of tracking and estimating human body motion has many useful applications including human-computer interaction, image-based rendering, surveillance, and video annotation. There are many hurdles in achieving reliable estimation of human motion. Some of the most challenging

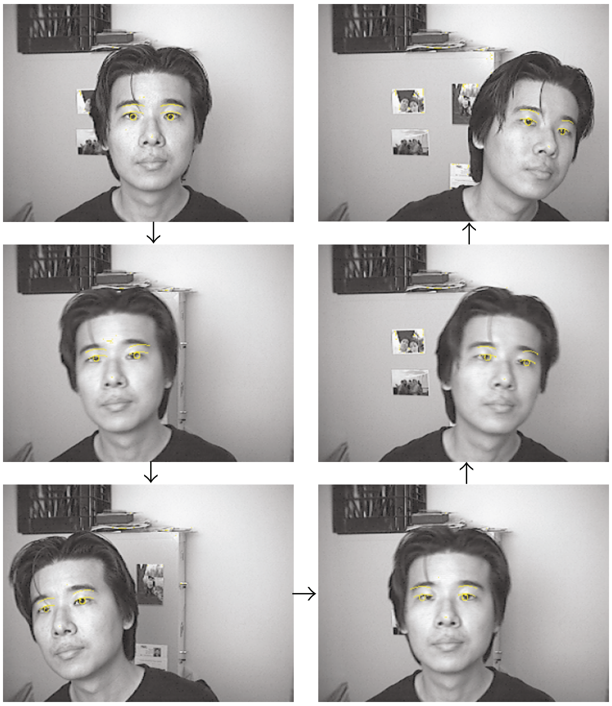

FIGURE 8: Sampled frames from a real human head movement sequence. While tracking shows some delays when the motion is fast, the tracked features yield correct head position and pose estimates.

ones are the complexity and variability of the appearance of the human body, the high dimensionality of articulated body pose space, and the pose ambiguity from a monocular view.

References [7, 10] employed articulated 3D models to constrain the bodily appearance and the kinematic prior. More recent trend is to use learned representation of body pose to constrain the pose space. Conditional prior between the configurations of body parts is learned to constrain the tracking in [26]. Reference [27] performed regression among learned instances of sampled pose appearance. Reference [28] made use of learned appearance-based low-dimensional representation of body postures to complement the weakness of model-based tracker. Another notable approach is to pose the tracking problem as a Bayesian graphical model inference problem. In [29], temporal consistency of body appearance is utilized to find and cluster body parts, and the tracking problem is carried out by finding the configuration of these parts represented by a Bayesian network. Reference [26] also belongs to this category.

We tackle the first problem (enforcing the invariance of appearance) by using the shape constraints provided by 3D models of body parts. The body pose is realized by the rotations of the limbs at the joints. The body model has a tree structure originating from the torso so that the motion of each part always follows the motion of its parent part. This global 3D representation provides the ability to represent most instances of articulate body pose efficiently. We assume that the initial pose can be provided by a more elaborate pose search method, such as that in [30].

The surface geometry as well as the silhouette information of the 3D model is utilized to compute the model fit to the data. For a given body pose, the image projection of each 3D part is computed and used to generate shape operators as in Section 2 to compute gradient response to the body image. For the whole body movement, local features are poorly defined, noisy, and often unreliable for establishing temporal correspondences. Boundary information is not always reliable either; body parts often occlude each other and the boundary of one part is easily confused with the boundary of the other.

The color (intensity) signature inside the part changes very little between frames when the motion is small; hence it provides a useful cue for discriminating one body part from another. Since it is not realistic to model the surface of the body and clothing, we simply assume that the apparent color signature is “drawn” on the 3D surface. We predict the appearance of the body from the current image frame to the next frame using the model surface.

The matches between the hypothetical and observed body poses are computed by combining the two aforementioned quantities and are fed into the nonlinear state estimation problem as measurements. Since we have not defined any dynamic equations for human activities, we make use of the motion information estimated from the previous frames to extrapolate the next positions of the state values, as in head tracking.

The measurements—silhouette and color appearance—from a monocular video do not usually give sufficient information to resolve the 3D body pose and self-occlusions of the limbs, especially for a side-view walking video. On the other hand, characteristics of human walking, or general human activities, can be exploited to provide useful constraints for tracking. We incorporated three kinds of constraints: the motion constraints at the joints, the symmetry of limbs in walking, and the periodicity of limb movement. The first two constraints are imposed at the measurement step, while the periodicity constraint is utilized at the prediction step. We found that this constraint on human walking provides very informative motion cues when the measurements are not available or not perfect due to occlusion.

5.1. Kinematic model of the body and shape constraints

As shown in Figure 12(a), we decompose the human body into truncated cones and ellipsoids. The body parts are organized as a tree with an ordered chain structure to provide a kinematic model of the limbs (Figure 12(b)). The cross-section of each cone is elliptical so that it can closely approximate torso and limb shapes. The computation of shape operators from each of these solids is described in Section 2. The motions of limbs are the rotations at the joints, and are represented using the relative rotations between local coordinate systems (Figure 12(c)). The local coordinate system is fixed at the joint that the part shares with its parent part. Each axis is determined so that the y axis is along the length direction (to the next joint) and the z axis is in the direction which the body is facing. For example, the joint which is the reference

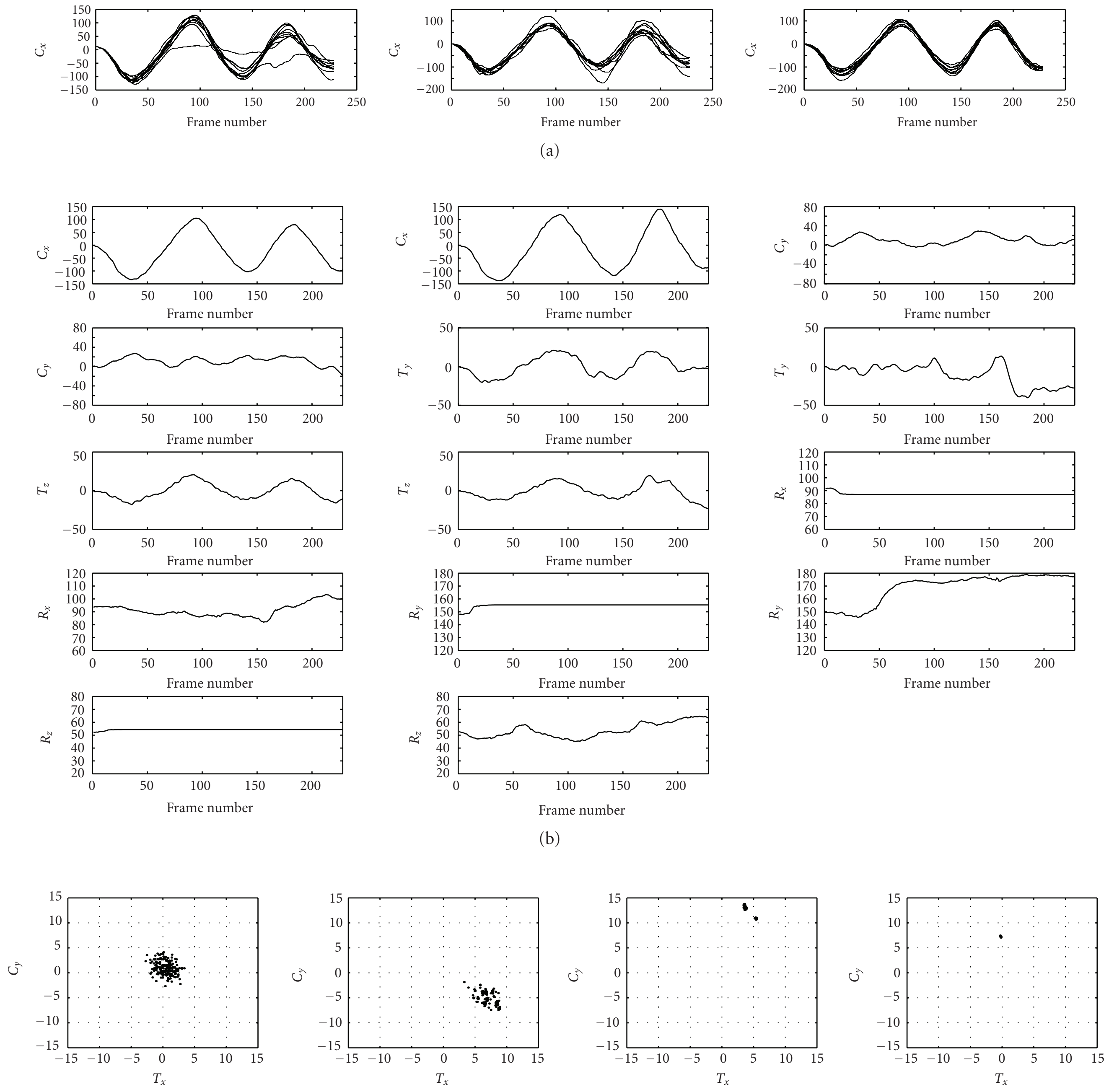

Figure 9: (a) Comparison of time update schemes. Top: no prediction adjustment, fixed diffusion. Middle: prediction adjustment only. Bottom: prediction adjustment and adaptive diffusion. (b) Comparison of diffusion schemes. Estimated location, pose, and motion parameters using adaptive (left column) and fixed (right column) diffusions. (c) The spread of the particles shows the ambiguity of the translation and rotation parameters. As the algorithm receives more data, the uncertainty decreases and is finally resolved.

point $v0_1$ of the second part in Figure 12(d) has the local coordinates $v0_1 = (0, \text{len}_1, 0)$ when the body is in an upright standing pose. The (global) coordinate of the tip of the second part after the rotations $R_1 = R(\theta_1)$ and $R_2 = R(\theta_2)$ is given by

$$v_2 = v_0 + R_1 \cdot (v0_1 + R_2 \cdot v0_2). \tag{25}$$

The rotation $R = R_z R_x R_y$ is the combination of the three rotations $R_x = R_x(\theta_x), R_y = R_y(\theta_y), R_z = R_z(\theta_z)$ around each axis, with rotation angles $(\theta_x, \theta_y, \theta_z)$.

5.2. Appearance constraints

As pointed out earlier, the fitting of the boundary feature is often confused with self-occlusion. Our 3D model provides not only the silhouette information, but also surface

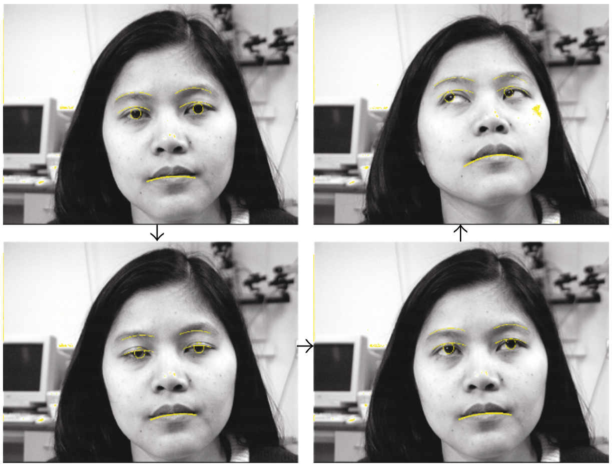

FIGURE 10: Tracking of independently moving local features. Squinting and iris movement are captured and tracked, as well as head movement.

geometry for predicting approximate appearance. While the 3D model does not make a noticeable difference when the motion is close to perpendicular to the camera axis or the color appearance is uniform, there are instances where the 3D surface model gives a better approximation than a planar model. Since it is not feasible to have a prior model of the color appearance of the human body or clothing, we compare only consecutive frames.

For a given image pixel $\mathbf{P}_t = (X, Y)$, we compare the intensity or the color value $I_t(X, Y)$ at $\mathbf{P}_t$ with the value $I_{t-1}(X', Y')$ at the corresponding pixel $\mathbf{P}_{t-1}$ in the previous frame. We can compute the 3D point $\mathbf{p}_t$ on the body part M_{ξ_t} by solving the quadratic equation (4) to get $z = z(X, Y)$ and

$$(x, y, z) = \left(\frac{X}{f}z, \frac{Y}{f}z, z\right). \tag{26}$$

Suppose we predict that the motion of a body part is determined by the following transformation: $\mathbf{p}_{t-1}$ on $M_{\xi_{t-1}}$ moves to $\mathbf{p}_t$ on M_{ξ_t} given by

$$\mathbf{p}_t = R\mathbf{p}_{t-1} + k, \tag{27}$$

where R is a rotation matrix and k is a translation vector.

Since we can compute $\mathbf{p}_{t-1}$ by the inverse transformation

$$\mathbf{p}_{t-1} = R^{-1}(\mathbf{p}_t - k), \tag{28}$$

the image plane projection of $\mathbf{p}_{t-1} = (x', y', z')$ is

$$(X', Y') = \left(\frac{x'}{z'}f, \frac{y'}{z'}f\right). \tag{29}$$

We can now compute the intensity (color) difference measure:

$$\sum_{(X,Y)\in\text{projection}(M_{\xi_t})} \left[I_t(X, Y) - I_{t-1}(X', Y')\right]. \tag{30}$$

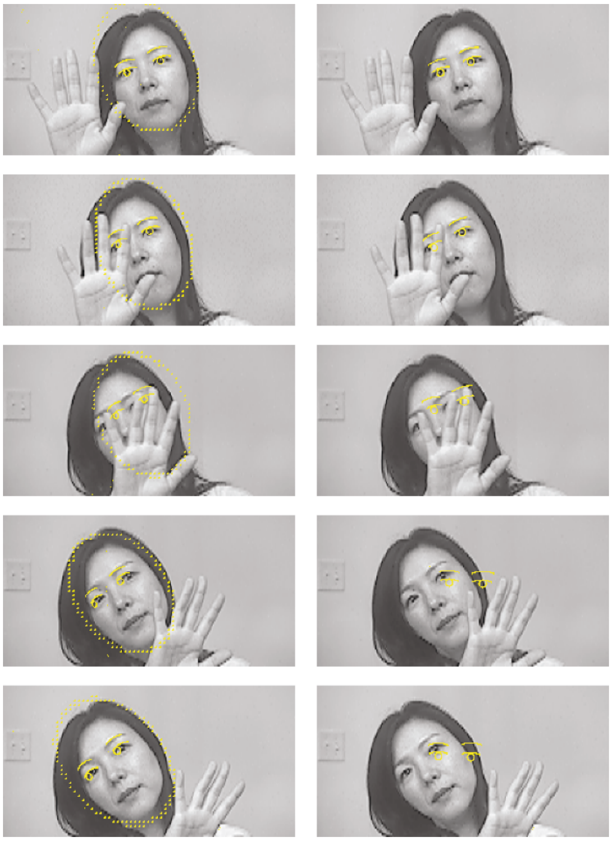

FIGURE 11: Tracking of occluded face. The first column: by using the ellipsoidal head filter, the tracking recovers after the occlusion. The second column: the tracking deviates from the correct track, due to the strong gradient response from the boundaries of fingers.

Because the morphing scheme only works on the current frame to predict the next one, there is a danger that slight prediction error can grow to larger error by the "snowballing effect." We have observed such occurrences, but found that the tracker recovers if the shape response is strong enough. The color/texture matching gives spatially slowly changing response profile, while the shape gradient filter response is very sensitive to spatial alignment and sometimes noisy. We have empirically verified that the appearance information positively contributed to stable tracking.

5.3. *Motion constraints*

For successful tracking, it is essential to explore the parameter space in an efficient way so that a viable set of random hypotheses is generated. The distribution of particles is updated by adjusting their weights using the measurements. Nevertheless, a prediction that is outside a tolerable range can lead to a biased estimate of the state and unstable tracking.

We also reduce the search space by incorporating proper prior constraints on human motion. The first constraint is the range of physically possible joint angles. Since we cannot rotate our limbs beyond certain limits, it is reasonable to limit the possible joint angles in the kinematic model. The constraint is enforced after making these measurements, by

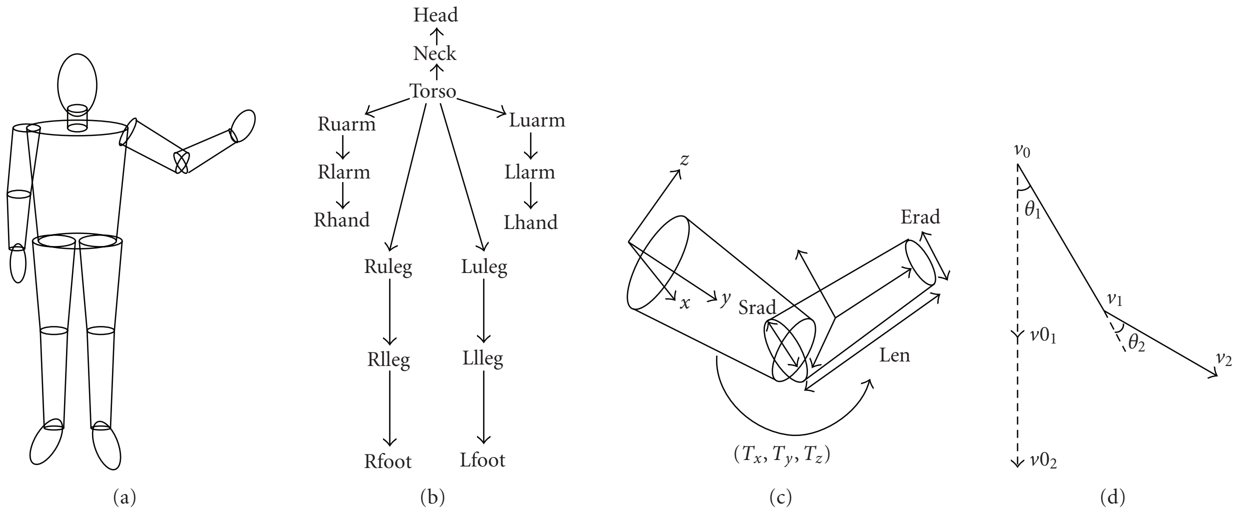

FIGURE 12: Shape and kinematic model of a human body. The body is decomposed into truncated cones and ellipsoids, and the joint motion is represented using rotation of the local coordinate system.

reweighting the fitness of each hypothesis. This has also been suggested in [31].

Another set of constraints can be incorporated that is more restrictive than the physical constraints of the joint angles: the symmetry of the limb angles about the axis of symmetry. There are correlations between the left and right joint angles and the angles between the arms and legs when a person walks in a usual way. The constraints are expressed as (Figure 13)

$$\begin{aligned}
C_{\text{LarmRarm}} &= (\text{luarm} - \text{armsymm})(\text{ruarm} - \text{armsymm}) \leq \epsilon,\\
C_{\text{LlegRleg}} &= (\text{luleg} - \text{legsymm})(\text{ruleg} - \text{legsymm}) \leq \epsilon,\\
C_{\text{LarmRleg}} &= (\text{luarm} - \text{armsymm})(\text{ruleg} - \text{legsymm}) \leq \epsilon,\\
C_{\text{RarmLleg}} &= (\text{ruarm} - \text{armsymm})(\text{luleg} - \text{legsymm}) \leq \epsilon,
\end{aligned} \tag{31}$$

where the limb pose parameters (luarm, ruarm, etc.) represent joint angles of the corresponding limbs, and the symmetry parameters (armsymm and legsymm) represent the angles of the axis of symmetry (which is determined from the torso pose). These quantities do not intend to pose hard constraints to the tracking, but to serve to limit the search space of the tracker so that the tracking never wanders off too much. The ϵ parameter controls the range of possible "deviation" from the strict symmetry (when $\epsilon = 0$). The symmetry constraints are implemented as a "reweighting" of the particles after making the image measurements. The reweighting factor is

$$\begin{aligned}
&Fac_{\text{Reweighting}}\\
&\quad = \exp\left(-K \cdot \left(C_{\text{LarmRarm}} + C_{\text{LlegRleg}} + C_{\text{LarmRleg}} + C_{\text{RarmLleg}}\right)\right).
\end{aligned} \tag{32}$$

The constant K adjusts the degree of contribution of the symmetry constraint. We impose these constraints only on the upper limbs, as the correlations between the lower limbs are more complicated. The physical constraints are enforced in the same manner.

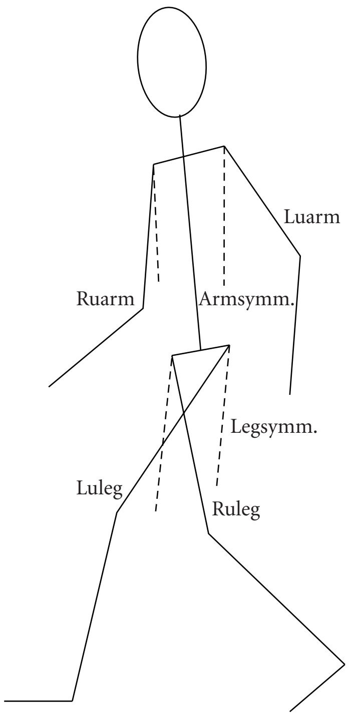

FIGURE 13: Joint angle constraints for human walking motion.

Another motion constraint exploits the periodicity of human walking. While walking, the left and right limbs move in very similar ways, and they lag behind each other's motions by one half of the walking cycle. For a side-view walking sequence, the gradients and appearance signatures of the front (visible) limbs are more prominent than those of the other limbs. We can alleviate the occlusion problem by estimating the phase difference between the left and right limbs and

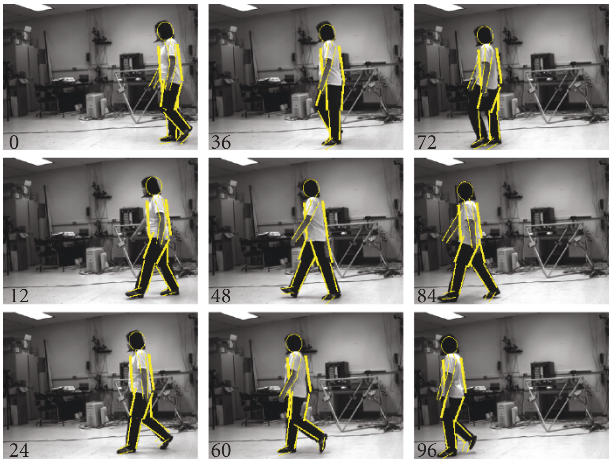

Figure 14: Side-view human walking sequence and tracked limb motion.

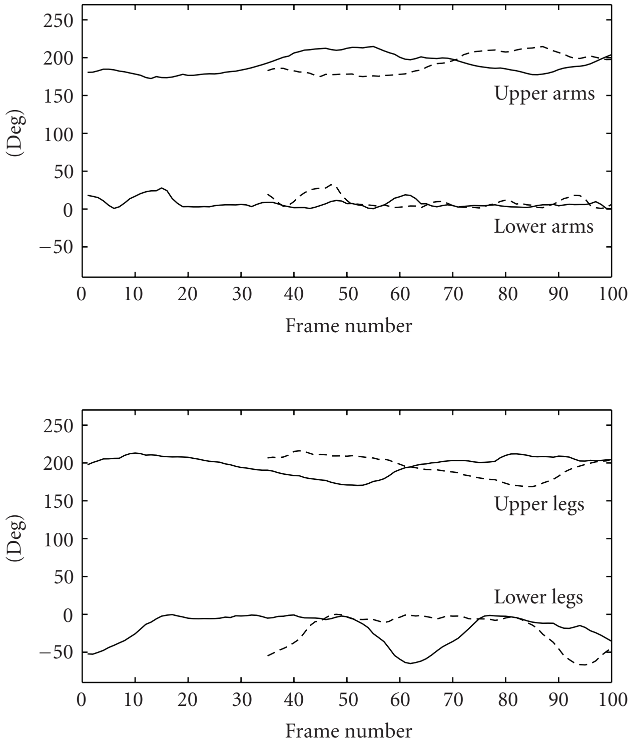

Figure 15: Estimated motion parameters. The top and bottom boxes show the plots of arm and leg pose parameters, respectively (solid lines: right limbs; dashed lines: left limbs).

incorporating the information into the prediction stage. That is, we predict the pose of an occluded limb using the pose parameters of its visible counterpart a half-period prior to the current frame. The initial prediction is adjusted by the image measurement. This constraint is far less restrictive than the motion priors employed in [32] or the learned motion model in [10], but we found that it contributes significantly to tracking performance.

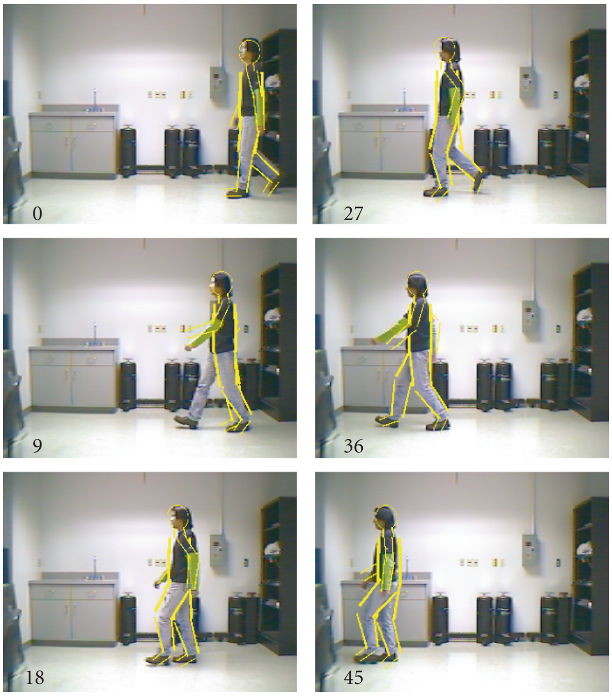

Figure 16: Tracked walking sequence with very low frame rate.

5.4. Experiments

We have applied the proposed method to many side-view walking sequences; one tracked sequence is shown in Figure 14. We used only about 600 particles for each frame; nevertheless, the result shows very good fitting of the body parts. Figure 15 shows the plots of the estimated pose parameters of the arms and legs. The plots for the left limbs (dashed lines) show only the portion after the half-period prediction is engaged. The plots for the upper limbs generally exhibit more apparent periodicity.

We tested our algorithm on a low frame rate color video; the result is shown in Figure 16. The frame rate is about 15 frames per second, and about 730 particles were used for tracking. We found that tracking is less stable for this sequence than for the first sequence (about 60 frames per second), although the latter has color information.

Figure 17 shows an outdoor walking sequence in which the frame rate is slightly lower than in the previous sequence. We have applied the joint angle symmetry constraints explained in the previous section. While these constraints are restrictive in that they are only applicable to standard walking motion, we found that they effectively limit the parameter space, making the tracking much more stable.

In both sequences (Figures 16 and 17), some parts of the body are not being tracked correctly at times (e.g., frame 9 in Figure 16 and frame 24 in Figure 17). These are some of the shortcomings of the tracker, mainly due to the left-right limb shape ambiguity and the high dimensionality of the problem. However, the tracker also shows robustness, in that it correctly tracks the body in later frames.

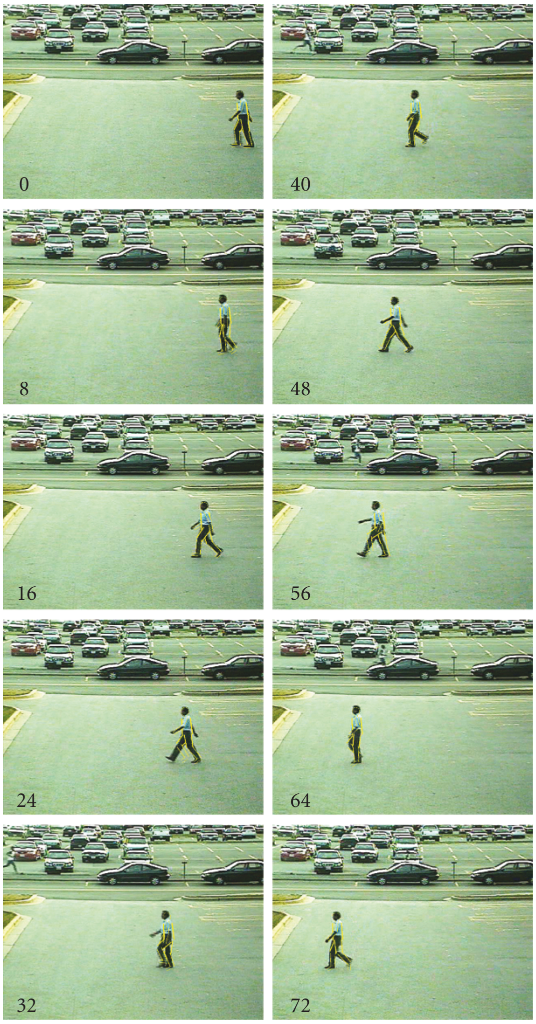

Figure 17: Outdoor walking sequence and tracked motion.

In all of the experiments, the number of particles was around 800 or less. Meanwhile [10, 31] reported using 4000 and 5000 particles, respectively, for successful tracking of similar type of walking sequences. We have observed that increasing the number of particles did not have much effect when we used more than 800 particles. This verifies the efficiency of the approach against other particle-based tracking methods.

6. SUMMARY

We have presented a method of tracking and estimating object motion using particle propagation and the 3D model of the object. The measurement update is carried out by particle branching according to 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 and effectiveness of tracking. More complete analysis and possible improvements would be desirable to ensure global optimization of model or "inertial" parameters. We used very simple models of the head and facial features to generate the shape operators for tracking. Since we need to compute the inverse camera projection for every pixel in the range of the shape operator, constructing the shape operator is highly time-consuming. As shown in Section 2, simple parameterization of the object surface and feature curves facilitates the construction of the shape operator. The measure helps to reduce computation, and we have obtained satisfactory results. Nevertheless, a more sophisticated parameterization would be desirable to achieve better pose and shape estimation. Figure 10 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. We have also applied the proposed method to the human body tracking problem. The human body model consists of head, torso, and limbs approximated by ellipsoids and truncated cones, and body pose is parameterized by joint angles. Other than boundary gradient information, between-frame appearance is computed by using the 3D surface model and provides another image measurement. We dealt with unobservability due to occlusions of limbs by exploiting the joint motion and symmetry constraints, and found that these natural dynamic constraints contribute to reliable tracking of human walking. We have verified that the method is able to efficiently track walking human in real-life video, using significantly fewer particles than other state-of-the-art approaches.

REFERENCES

[1] H. Moon, R. Chellappa, and A. Rosenfeld, "Optimal edge-based shape detection," *IEEE Transactions on Image Processing*, vol. 11, no. 11, pp. 1209–1227, 2002.

[2] A. Azarbayejani and A. P. Pentland, "Recursive estimation of motion, structure, and focal length," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 562–575, 1995.

[3] T. J. Broida, S. Chandrashekhar, and R. Chellappa, "Recursive 3-D motion estimation from a monocular image sequence," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 4, pp. 639–656, 1990.

[4] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.

[5] J. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032–1044, 1998.

[6] M. Isard and A. Blake, "CONDENSATION—conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[7] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '00)*, vol. 2, pp. 126–133, Hilton Head Island, SC, USA, June 2000.

[8] J. Sullivan, A. Blake, M. Isard, and J. MacCormick, "Object localization by Bayesian correlation," in *Proceedings of the 7th*

IEEE International Conference on Computer Vision (ICCV '99), vol. 2, pp. 1068–1075, Kerkyra, Greece, September 1999.

[9] B. Li and R. Chellappa, "Simultaneous tracking and verification via sequential Monte Carlo method," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '00)*, vol. 2, pp. 110–117, Hilton Head Island, SC, USA, June 2000.

[10] H. Sidenbladh, M. J. Black, and D. J. Fleet, "Stochastic tracking of 3D human figures using 2D image motion," in *Proceedings of the 6th European Conference on Computer Vision (ECCV '00)*, Dublin, Ireland, June-July 2000.

[11] M. Zakai, "On the optimal filtering of diffusion processes," *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, vol. 11, no. 3, pp. 230–243, 1969.

[12] Z. S. Haddad and S. R. Simanca, "Filtering image records using wavelets and the Zakai equation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 11, pp. 1069–1078, 1995.

[13] D. Crisan, J. Gaines, and T. Lyons, "Convergence of a branching particle method to the solution of the Zakai equation," *SIAM Journal on Applied Mathematics*, vol. 58, no. 5, pp. 1568–1590, 1998.

[14] D. Crisan and M. Grunwald, "Large deviation comparison of branching algorithm versus resampling algorithms: application to discrete time stochastic filtering," *Tech. Rep. 9*, Cambridge University Statistical Laboratory, Cambridge, England, 1999.

[15] M. J. Black and A. D. Jepson, "Eigen tracking: robust matching and tracking of articulated objects using a view-based representation," *International Journal of Computer Vision*, vol. 26, no. 1, pp. 63–84, 1998.

[16] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real time video," in *Proceedings of the 4th IEEE Workshop on Applications of Computer Vision (WACV '98)*, pp. 8–14, Princeton, NJ, USA, October 1998.

[17] K. Toyama and A. Blake, "Probabilistic tracking in a metric space," in *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV '01)*, Vancouver, BC, Canada, July 2001.

[18] D. B. Gennery, "Visual tracking of known three-dimensional objects," *International Journal of Computer Vision*, vol. 7, no. 3, pp. 243–270, 1992.

[19] D. DeCarlo and D. Metaxas, "Optical flow constraints on deformable models with applications to face tracking," *International Journal of Computer Vision*, vol. 38, no. 2, pp. 99–127, 2000.

[20] D. Cremers, "Dynamical statistical shape priors for level set-based tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1262–1273, 2006.

[21] O. Chomat and J. L. Crowley, "Probabilistic recognition of activity using local appearance," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, vol. 2, pp. 104–109, Fort Collins, Colo, USA, June 1999.

[22] A. Bensoussan, *Stochastic Control of Partially Observable Systems*, Cambridge University Press, Cambridge, UK, 1992.

[23] L. Ljung, "Asymptotic behaviour of the extended Kalman filter as a parameter estimator for linear systems," *IEEE Transactions on Automatic Control*, vol. 24, no. 1, pp. 36–50, 1979.

[24] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3D models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, pp. 322–336, 2000.

[25] J. Xiao, T. Moriyama, T. Kanade, and J. Cohn, "Robust full-motion recovery of head by dynamic templates and re-registration techniques," *International Journal of Imaging Systems and Technology*, vol. 13, no. 1, pp. 85–94, 2003.

[26] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard, "Tracking loose-limbed people," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 1, pp. 421–428, Washington, DC, USA, June-July 2004.

[27] A. Agarwal and B. Triggs, "Recovering 3D human pose from monocular images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 44–58, 2006.

[28] C. Curio and M. A. Giese, "Combining view-based and model-based tracking of articulated human movements," in *Proceedings of IEEE Workshop on Motion and Video Computing (WACV/MOTIONS '05)*, vol. 2, pp. 261–268, Breckenridge, Colo, USA, January 2005.

[29] D. Ramanan, D. A. Forsyth, and A. Zisserman, "Tracking people by learning their appearance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 65–81, 2007.

[30] M. W. Lee and I. Cohen, "Proposal maps driven MCMC for estimating human body pose in static images," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. 334–341, Washington, DC, USA, June-July 2004.

[31] J. Deutscher, B. North, B. Bascle, and A. Blake, "Tracking through singularities and discontinuities by random sampling," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, vol. 2, pp. 1144–1149, Kerkyra, Greece, September 1999.

[32] J.-C. Cheng and J. M. F. Moura, "Capture and representation of human walking in live video sequences," *IEEE Transactions on Multimedia*, vol. 1, no. 2, pp. 144–156, 1999.