Estimation of contour motion and deformation for nonrigid object tracking

Jie Shao,[1,*] Fatih Porikli,[2] and Rama Chellappa[1]

[1]*Center for Automation Research and Department of Electrical and Computer Engineering, University of Maryland, College Park, Maryland 20742, USA*
[2]*Mitsubishi Electric Research Laboratories, Cambridge, Massachusetts 02139, USA*
[*]*Corresponding author: shaojie@cfar.umd.edu*

Received September 8, 2006; revised January 3, 2007; accepted January 11, 2007; posted March 7, 2007 (Doc. ID 74806); published July 11, 2007

We present an algorithm for nonrigid contour tracking in heavily cluttered background scenes. Based on the properties of nonrigid contour movements, a sequential framework for estimating contour motion and deformation is proposed. We solve the nonrigid contour tracking problem by decomposing it into three subproblems: motion estimation, deformation estimation, and shape regulation. First, we employ a particle filter to estimate the global motion parameters of the affine transform between successive frames. Then we generate a probabilistic deformation map to deform the contour. To improve robustness, multiple cues are used for deformation probability estimation. Finally, we use a shape prior model to constrain the deformed contour. This enables us to retrieve the occluded parts of the contours and accurately track them while allowing shape changes specific to the given object types. Our experiments show that the proposed algorithm significantly improves the tracker performance. © 2007 Optical Society of America

OCIS codes: 330.7310, 330.4150.

1. INTRODUCTION

Visual tracking is an essential component of many applications from intelligent robotics to video surveillance. Basically, there are three groups of tracking methods: correspondence-based, transformation-based, and contour-based. The first group of methods is based on establishing correspondences between feature points. The second group performs tracking by estimating object motion, in which the objects are usually assumed to be made of planar shapes such as ellipses and rectangles. The last group achieves tracking by finding the object contour in successive frames. It applies to cases when not only the location but also the deformation of a target is estimated during tracking. Some useful applications of nonrigid tracking include surveillance tracking for recognition purposes and echocardiography tracking for computer-aided diagnosis. The tracking approach proposed in this paper belongs to the group of contour-based tracking methods.

A. Related Work

Several contour-based tracking methods have been proposed in the literature. As a milestone in contour-based tracking research, CONDENSATION, a parameterized B-spline contour tracking algorithm, was proposed by Isard and Blake [1]. It uses a particle filter as the basic framework to track global motion and deformation. The algorithm yields robust results when applied to rigid objects. However, it has no explicit criterion for extracting the exact boundary of a nonrigid object in its observation model. In [2], Li *et al.* presented a particle filter for nonrigid object contour tracking. But the algorithm lacks an appropriate model for discriminating a real boundary from all the detected edge points. The snake [3,4], or dynamic contour-based method, is another common approach that evolves an object boundary such that a weighted sum of external and internal energy terms is minimized. However, such methods are restricted to a relatively small range of situations, because they require that intensities inside objects be fairly uniform. Furthermore, their computational complexity makes them less suitable for real-time applications. The level-set approach is also a powerful method that deals with topological changes of the moving level-set function by using partial differential equations (PDEs) that describe the object motion, boundary, and region-based information [5–8]. But the PDE-based approach also prefers uniform intensity distributions inside objects.

Some other approaches closely related to nonrigid contour tracking were presented in [9–11]. The concepts of motion and deformation were defined in [11]. Motion is parameterized by a finite-dimensional group action, and deformation is the total deformation of the object contour (infinite-dimensional group) modulo the finite-dimensional motion group. By incorporating the prior information of the system dynamics into the deformation framework, Jackson *et al.* [9] proposed a nonlinear dynamical model for tracking a slowly deforming and moving contour, with the contour implicitly represented as the infinite-dimensional locus of zeros of a given function. The algorithm suffers from expensive computations due to the joint minimization of group action and deformation. The work in [10] extended the ideas in [9]. It used a particle filter to estimate the conditional probability distribution of motion and contour, which enables the incorporation of a prior system model along with an observation

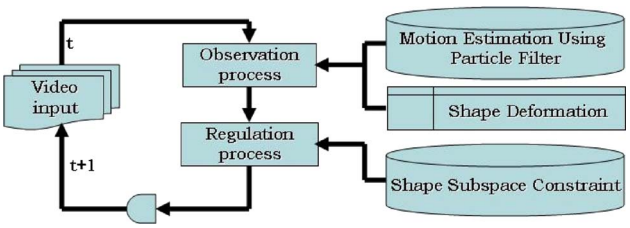

Fig. 1. (Color online) Illustration of the proposed tracking system.

model. However, this algorithm also has limitations. It counts on only appearance cues in the observation model and, because of sensor motion, lacks the ability to handle moving backgrounds.

Our approach shares some similarities with that of [10]. We claim that tracking nonrigid objects can be accomplished by estimating both translational (finite-dimensional, motion) and nontranslational movements (infinite-dimensional, deformation) of objects. However, instead of estimating both motion and deformation in one step, we propose a sequential approach. We first estimate the motion and then the deformation. The estimation of deformation fulfills the operation of discriminating the real boundary from all the edge points, the majority of which may be from the background. Robustness can be improved by constraining the deformation by using a prior shape model. Therefore we decompose the task of tracking nonrigid object contour into three components:

- 2D Motion estimation. It estimates the objectwise spatial rigid-body motion, including translation and rotation parameters. Since the motion parameters are finite dimensional, we use a particle filter to estimate them.
- 2D Shape deformation. It captures the pose changes of nonrigid objects. Each pixel on the boundary may have different but correlated deformations. We construct a deformation probability map based on a statistical analysis of different cues in each frame. Deformation is determined according to boundary pixels with higher deformation probabilities.
- Shape regulation. It uses a trained shape subspace to restrict shape deformations. Regulation also reconstructs the occluded parts of the contours. Our method adaptively integrates the off-line trained prior shape model with the object in the current video sequence.

Figure 1 presents a schematic illustration of the proposed system. The rest of the paper is organized as follows. In Section 2, we introduce some preliminary concepts of the active contour tracking algorithm. In the next three sections, we describe the three sequential stages in the nonrigid contour tracking algorithm: motion estimation, deformation estimation, and regulation. We present experimental results in Section 6, followed by conclusions and discussion in Section 7.

2. PRELIMINARIES

A. B-spline Parametric Curves

In our contour tracking system, the tracking target is represented by the parametric B-spline curve. The visual 2D curves outlining the objects are represented in terms of parametric B-spline curves $\mathbf{r}(s)=[x(s),y(s)]^T$ [12]. The coordinates $[x(s),y(s)]$ are both spline functions of the curve parameter s. Furthermore, we use a set of control points $\mathbf{Q}=\{q_1,q_2,\ldots,q_L\}$ to represent the B-spline curve, where each control point is defined as $q_l=(q_l^x,q_l^y)^T$ and L is the number of control points. One important reason for using the control-point representation is that the set of $\mathbf{Q}$ can uniquely determine a given B-spline curve. If we define the dynamic model that describes the contour motion as an affine transform, it is sufficient to apply the transform to the control points. Once the control points are transformed, the B-spline curve is transformed in the same manner. The property not only significantly improves the computational efficiency but also helps to discretize the infinite deformation parameters into finite deformation parameters; i.e., the deformation of the curve can be approximated by the deformations at the control points.

B. Particle Filter

The objective of motion tracking is to recursively estimate the state of the dynamic model given some noisy visual observations, which can be formulated using the Bayesian approach by computing

$$p(\theta_t|Y_{1:t}) \propto p(Y_t|\theta_t) \int p(\theta_t|\theta_{t-1})p(\theta_{t-1}|Y_{1:t-1})\mathrm{d}\theta_{t-1}, \quad (1)$$

where θ denotes the state vector, Y the observation, and $p(Y_t|\theta_t)$ the likelihood function at time instant t. Observation Y could be raw images or features in images. All inferences regarding the unknown state vector are based on the posterior probability in Eq. (1). The basic criterion is to find the vector with the maximum posterior probability. Many techniques (such as the Kalman filter [13] or the particle filter [14]) can be used to achieve the goal. The former is effective when the data are modeled as a linear Gaussian model. The latter, also known as the sequential Monte Carlo algorithm, is a set of simulation-based methods proposed to handle more complex data that may be non-Gaussian, and/or nonlinear. Many contour tracking algorithms use the particle filter algorithm because it is flexible, easy to implement, parallelizable [15], and applicable to general settings. We also use the particle filter to estimate the state vector of the dynamic model.

In the particle filter algorithm, the *prediction step* samples new particles based on the state transition probability $p(\theta_t|\theta_{t-1})$ and the previous posterior distribution $p(\theta_{t-1}|Y_{1:t-1})$, while the *update step* is controlled by particle weights characterized by the likelihood function $p(Y_t|\theta_t)$:

$$\omega_t^{(j)} \propto p(Y_t|\theta_t^{(j)}), \quad (2)$$

The algorithm approximates the current posterior distribution $p(\theta_t|Y_{1:t})$ by a set of weighted particles $S_t=\{\theta_t^{(j)},\omega_t^{(j)}\}_{j=1}^J$ with J representing the number of particles. To avoid the potential of the particles collapsing into a few particles with high weights, the sequential importance sampling [16,17] draws particles from a proposal distribution $g(\theta_t^{(j)}|\theta_{t-1}^{(j)},Y_{1:t})$ and eliminates particles with lower weights. The weights are assigned as

$$\omega_j^{(j)} \propto \frac{p(Y_t|\theta_t^{(j)})p(\theta_t^j|\theta_{t-1}^{(j)})}{g(\theta_j^{(j)}|\theta_{t-1}^{(j)},Y_{1:t})}. \tag{3}$$

The selection of proposal distribution can be tuned to applications under consideration.

3. MOTION ESTIMATION

A. Dynamic Motion Model

We use the parameters of a 2D affine transform to represent the finite-dimensional motion of the object, which allows rotation and translation and is independent of scale. The contour transform is given in terms of the homogeneous coordinates of the control points as

$$\begin{bmatrix} q_{l,t-1} \\ 1 \end{bmatrix} = T \cdot \begin{bmatrix} q_{l,t} \\ 1 \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} q_{l,t} \\ 1 \end{bmatrix}, \tag{4}$$

where $T \in \mathbb{R}^{3\times 3}$ represents an affine transform matrix with independent scale factors along both the x and the y axes. Accordingly, the state vector of the dynamic model is defined as

$$\theta_t = (T_{11} \quad T_{12} \quad T_{21} \quad T_{22} \quad T_{13} \quad T_{23})^T. \tag{5}$$

B. Estimate State Vector by Particle Filter

We use the particle filter to obtain the maximum *a posteriori* (MAP) estimate of the state vector θ.

1. Prediction Step

Rather than using a proposal distribution for prediction, we predict the configuration of particles based on the following state transition model:

$$\theta_t = \hat{\theta}_{t-1} + v_t + U_t \tag{6}$$

with $\hat{\theta}_{t-1}$ is the previous state estimate, v_t is the predicted adaptive velocity in the motion vector, and U_t is the driving noise, assumed to be zero-mean Gaussian noise. The computation of v_t depends on the configuration of the previous particles in the prediction step. Therefore, the diversity of particles is not compromised.

The prediction of v_t is based on the assumption of brightness invariance [18], which means that there exists a θ_t such that the warping patch is similar to the previous image patch. We denote $Z(\mathbf{Q}_t)$ as the intensities (colors) of the control point set $\mathbf{Q}_t$ (for robustness purposes we use the corresponding intensities of Gaussian smoothed image frames). Then there exists a θ_t that satisfies

$$Z_t(\mathbf{Q}_t) = Z_{t-1}\left(T(\theta_t) \cdot \begin{bmatrix} \mathbf{Q}_t \\ \mathbf{1} \end{bmatrix}\right) = Z_{t-1}(\hat{\mathbf{Q}}_{t-1}), \tag{7}$$

where $\hat{\mathbf{Q}}_{t-1}$ denotes the control-point set estimated at $t-1$. We simplify Eq. (7) as $\mathcal{T}\{Z_t,\theta_t\}=\hat{Z}_{t-1}$, where $\hat{Z}_{t-1}$ represents the corresponding intensities of $\hat{\mathbf{Q}}_{t-1}$. Generally, $\mathcal{T}$ symbolizes all kinds of transforms, parameterized by θ_t, on the control-point set $\mathbf{Q}_t$, whose intensities are represented by Z_t, from frame t to $t-1$. But since we consider only 2D rigid-body *motion* at this stage and it is assumed to satisfy an affine transform, $\mathcal{T}$ represents an affine transform from frame t to $t-1$. Approximating $\mathcal{T}\{Z_t;\theta_t\}$ via a first-order Taylor series expansion around $\hat{\theta}_{t-1}$ yields

$$\mathcal{T}\{Z_t;\theta_t\} \simeq \mathcal{T}\{Z_t;\hat{\theta}_{t-1}\} + \mathcal{C}_t(\theta_t - \hat{\theta}_{t-1}) = \mathcal{T}\{Z_t;\hat{\theta}_{t-1}\} + \mathcal{C}_t v_t, \tag{8}$$

where $\mathcal{C}_t = \partial\mathcal{T}/\partial\theta$ is the Jacobian matrix. Substituting $\hat{Z}_{t-1}$ into Eq. (8), we obtain

$$\hat{Z}_{t-1} \simeq \mathcal{T}\{Z_t;\hat{\theta}_{t-1}\} + \mathcal{C}_t v_t, \tag{9}$$

$$v_t \simeq -\mathcal{B}_t(\mathcal{T}\{Z_t;\hat{\theta}_{t-1}\} - \hat{Z}_{t-1}), \tag{10}$$

where $\mathcal{B}_t$ is the pseudo-inverse of $\mathcal{C}_t$. Using the differences in motion vectors and the observation matrix as inputs, we obtain a least-squares solution to $\mathcal{B}_t$ as

$$\Theta_{t-1}^{\delta} = [\theta_{t-1}^{(1)} - \hat{\theta}_{t-1}, \ldots, \theta_{t-1}^{(J)} - \hat{\theta}_{t-1}], \tag{11}$$

$$\mathcal{Z}_{t-1}^{\delta} = [Z_{t-1}^{(1)} - \hat{Z}_{t-1}, \ldots, Z_{t-1}^{(J)} - \hat{Z}_{t-1}], \tag{12}$$

$$\mathcal{B}_t = (\Theta_{t-1}^{\delta}\mathcal{Z}_{t-1}^{\delta T})(\mathcal{Z}_{t-1}^{\delta}\mathcal{Z}_{t-1}^{\delta T})^{-1}, \tag{13}$$

where Θ_{t-1}^{δ} is the set of differences between all particle samples of $\{\theta_{t-1}^{(t)}\}_{j=1}^{J}$ and the optimal estimate $\hat{\theta}_{t-1}$, $Z_{t-1}^{(j)}$ is the jth patch sample with state vector sample $\theta_{t-1}^{(j)}$, and $\mathcal{Z}_{t-1}^{\delta}$ is the set of all intensity differences between samples of $\{Z_{t-1}^{(j)}\}_{j=1}^{J}$ and $\hat{Z}_{t-1}$. Obviously, the particle configuration at $t-1$ is incorporated here for prediction.

2. Updating Step

The weight is updated based on the likelihood function $p(Y_t|\theta_t)$. We follow the definition of the observation model in [1]. We search along the normal line $\mathbf{n}_l$ at each control point q_l, which is determined by the corresponding θ, and detect feature points $\{z_j^{(l)}\}_{j=1}^{N_l}$, where N_l is the number of features detected along the normal. The existence of multiple feature points is due to background clutter. Assuming that $\{z_j^{(l)}\}$ can be modeled as a spatial Poisson distribution along the normal lines and the true control point is a Gaussian distribution, the 1D measurement density along $\mathbf{n}_l$ can be determined by the distances between the feature points to the corresponding control point, characterized as

$$p_l(z|q_l) \propto 1 + \frac{1}{\sqrt{2\pi}\sigma\psi\lambda}\sum_{j=1}^{N_l}\exp\left[-\frac{(z_j^{(l)} - q_l)^2}{2\sigma^2}\right], \tag{14}$$

where ψ is the probability of nondetection, λ is the density of clutter in the Poisson distribution, and σ is the standard deviation of the Gaussian distribution. Figure 2 is an intuitive illustration of evaluating the likelihood function for one control point. With the assumption that feature outputs on distinct control points are statistically independent, the overall likelihood becomes

$$p(Y|\theta) = \prod_{l=1}^{L} p_l(z|q_l). \tag{15}$$

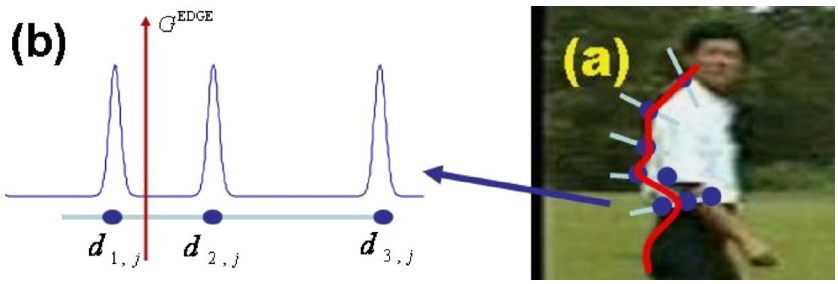

Fig. 2. (Color online) Illustration of estimating the measurement model. (a) The red (upper) line is the contour determined by the control points. Light lines are the normal lines on the control points. The solid dots are feature points detected by the normal lines; (b) 1D measurement density along the line normal on one control point q_j; $d_{1,j}$, $d_{2,j}$ and $d_{3,j}$ are the three distances between the feature points and the corresponding control point. The vertical axis G^{EDGE} represents the gradient magnitude along the normal line $\mathbf{n}_j$.

4. DEFORMATION ESTIMATION

After the MAP estimator $\hat{\theta}$ is obtained and the transformed control point $\hat{\mathbf{Q}}$ is acquired based on $\hat{\theta}$, the transformed curve $\tilde{\mathbf{r}}$ is determined. The next thing is to enforce local deformation, i.e., find the real boundary points [1]. We used a simple strategy that the exact contour is determined by selecting feature points with maximum gradient magnitudes detected on corresponding normal lines. In other words, the contour estimation counts on only the gradient magnitudes. Unfortunately, this strategy does not always work, especially when the background is heavily cluttered or the object undergoes shape deformations between frames.

In our algorithm, we identify the correct feature points by detecting the deformation along the normal lines on transformed control points $\hat{\mathbf{Q}}$. Two elements are involved. One relates to the assumption that the real boundary points of an object are detected along the orthogonal directions of the contour. It implies that the scanning range of normal lines influences the probability of the real boundary being detected. The other is that the gradient magnitudes are not sufficiently robust for exact contour delineation, especially when contaminated by background clutter and object textures. Accordingly, our strategy for deformation estimation contains two new features: (1) set normal lines adaptive and (2) integrate several statistical cues into a deformation confidence map.

A. Set Normal Lines Adaptive

The scanning range of normal lines is determined by searching lengths and centers. Earlier algorithms [1,2,8,19,20] set the search lengths and centers of normal lines identical and fixed, which may result in false detections due to inadequate modeling of shape variations. We allow for adaptability of the normal lines to reduce the number of false detections.

1. Lengths of Normal Lines

Intuitively, a short normal line may miss the true boundary pixel while a long one may intersect with edge points from background clutter. To reduce the possibility of a normal line intersecting with background clutter without sacrificing the chance of finding the actual boundary pixel, the lengths of normal lines are altered according to the pose variations of the corresponding contour control points in training sequences. For example, in sequences of walking humans, the relative positions of the head and trunk change slightly from frame to frame; on the other hand, the sides, especially the legs and arms change their relative positions more. Therefore, we should set the normal lines of large pose variations longer than those of small pose variations. The pose variations of pixels can be learned offline. For example, walking pedestrian samples are acquired from the University of South Florida dataset [21], which consists of one thousand 120×80 binary images with aligned pedestrian silhouettes. Accordingly, the length of normal line, denoted by $u(l)$, is adaptively set as

$$\xi(l) = \mathbf{E}\|q_l^k - \mathbf{E}(q_l^k)\|^2, \tag{16}$$

$$u(l) \propto L_{\min}\log\frac{\xi(l)}{\min(\xi(l))}, \tag{17}$$

where L_{min} is a constant representing the minimum length and k denotes the index of training samples.

2. Centers of Normal Lines

Earlier algorithms set the centers of scanning normal lines as the control points on the estimated contour, which may cause the intertwining of normal lines, because the same point may be selected twice along two normal lines and the output contour may end up looped. Making the line centers adaptive by applying a distance transform (DT) [22] significantly reduces the probability of normal line intersections inside the object. The detailed steps are listed in Table 1. The given algorithm concerns only close contours. For open contours, we will force them to be closed by simply linking the first point and the last point. Figure 3 demonstrates the procedure of sketching adaptive normal lines along the estimated contour, with which we are able to search for the real contour pixels with more flexibility.

B. Multicue Deformation Probability Map

To extract the real contour, we define a posterior deformation probability map as $P_t(Y|\hat{\mathbf{Q}})$ based on the transformed control point set obtained by global motion estimation, with Y representing related features. In the map, a high probability implies that the corresponding pixel is more likely to be on the real contour and a low value implies a lower likelihood. Instead of using edge magnitudes as the only feature, we integrate several cues, including edge

magnitudes, edge orientations, shape templates, and foreground estimation, to evaluate the deformation probability.

1. Multicue Fusion

The process of fusing different cues can be interpreted as using multiple measurement sources. Several real-time tracking algorithms reported in the literature fuse motion and appearance cues. Viola *et al.* [23] proposed a pedestrian detection system that integrates intensity and motion information. The detector is trained (using *AdaBoost*) to take advantage of both motion and appearance information. There are approaches that select the "optimal" cue for the entire sequence, with other less reliable cues as supporting cues. A good example of cue selection is found in the layered hierarchy extraction algorithm proposed by Toyama and Hager [24]. CONDENSATION, as an extension of CONDENSATION with importance sampling, proposed by Isard and Blake [1] also employed a multicue scheme by complementing the original intensity gradient cue with a supportive color cue. Sidenbladh *et al.* [25] incorporated motion measurements in the particle filter framework. Odobez and Gatica-Perez [26] improved the particle filter by using transition prior as the proposal distribution. Some approaches assume that each cue has the same reliability in all frames. For example, Birchfield [27] used intensity and color distribution of the target with equal confidences for robust head tracking. Democratic integration, proposed by Triesch and Van der Malsburg [28] introduced an adaptive multicue integration so that the contribution of each cue varies according to its reliability in each frame. Such a strategy improves the robustness of the system in different visual situations. Vermaak *et al.* extended the idea for integration of multiple cues in the particle filter framework [29].

Compared with these approaches, our approach uses some special feature cues more natural to contour tracking. The contribution of each cue to the entire system is reflected by a probability. Assume that we have M cues: the observation can be represented by $Y=(Y_1,\ldots,Y_M)$. We further assume that the observations are conditionally independent [30]. The deformation probability is therefore factorized as

$$P_t(Y|\hat{\mathbf{Q}}) = \prod_{i=1}^{M} P_t(Y_i|\hat{\mathbf{Q}}). \tag{21}$$

Scanning for pixels with maximum probability values (maximum-likelihood estimator) on adaptive normal lines, we obtain the refined contour pixels, denoted $\hat{\mathbf{r}}$. As an example, we show some probability maps from different cues on a processed frame in Fig. 4, together with the fusion map $P_t(Y|\hat{\mathbf{Q}})$. The fused result shows a reduction of noise from the gradient magnitude cues inside the contour caused by object textures.

We introduce the computation of probability maps from different cues in the rest of this section. It is worth noting that since the scanning range for each pixel is 1D, the 2D deformation probability map P can be further simplified to several 1D probability vectors associated with each control point. Therefore the calculation of the deformation probabilities is limited to normal lines. This helps to improve the computational efficiency.

2. Gradient Magnitude Cues

As shown in Fig. 4(a), the gradient magnitude is an important feature for representing an object boundary. However, when the object itself is not homogenous in color or intensity, many edges are generated inside the object. We want to minimize the effect of inside edges. Anisotropic diffusion is one possible approach to make the entire image to be more uniform in color or texture while still preserving the object boundaries [31]. Therefore it is highly probable that points with high gradient magnitudes after diffusion belong to the boundary of the target.

After the diffused feature map $\mathcal{EI}$ is extracted from the original image, a motion mask $\Delta\mathcal{I}$ indicating the possible area where motion could occur is applied to $\mathcal{EI}$ to further

Table 1. Algorithm 1: Set Center of the Normal Line Adaptive

1. Based on the transformed control points set $\hat{\mathbf{Q}}_t$ (the result from global motion estimation), construct a binary image $\mathcal{BI}$, set the region Ω circled by the contour $\tilde{\mathbf{r}}_t$ to 1;

2. Apply DT to $\mathcal{BI}$ to obtain a distance map $\mathcal{DI}$, which is defined as

$$\mathcal{DI}(\mathrm{x}) = \begin{cases} \min_{\mathbf{y}\in\tilde{\mathbf{r}}} \mathrm{dist}(\mathbf{x},\mathbf{y}), & \mathbf{x}\in\Omega \\ 0 & \text{otherwise} \end{cases}. \tag{18}$$

3. On each control point $\hat{q}_l$, draw a normal line $\mathbf{n}_l$, find the distance value satisfying $\mathcal{DI}_l=\max_{\mathbf{x}\in\mathbf{n}_l}\mathcal{DI}(\mathbf{x})$.
The lengths of the normal line on two sides of the control point are set as follows:

$$u(l)_{\mathrm{in}} = \min(u(l)/2, \mathcal{DI}_l - d_0), \tag{19}$$

$$u(l)_{\mathrm{out}} = \max(u(l)/2, u(l) - [\mathcal{DI}_l - d_0]), \tag{20}$$

where d_0 represents a minimum safe distance to avoid contour loops. Here $d_0=2$.

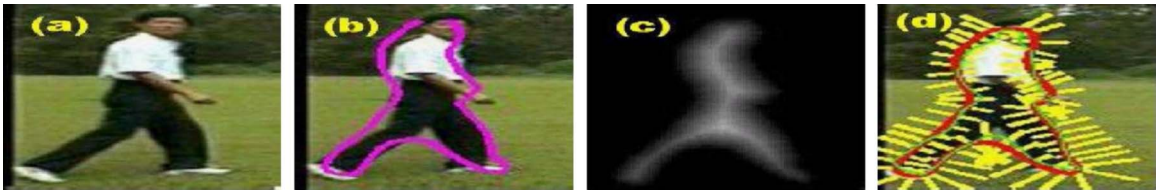

Fig. 3. (Color online) Example of how to make a normal line scanning adaptive. (a) Cropped original object, (b) estimated contour, (c) the distance transformed object, (d) normal lines on control points.

suppress the background clutter. The masked map is then filtered by $\mathbf{G}$, a smoothing Gaussian filter. Finally, we convert the filtered map to a magnitude probability map P by

$$P_t(Y_m|\hat{\mathbf{Q}}) = \alpha_m(\mathcal{EI} \cdot \Delta\mathcal{I}) * \mathbf{G}, \tag{22}$$

where α_m is a normalizing coefficient and * denotes convolution. Figure 5 demonstrates an example of performance improvement when we utilize the diffusion edge map convolved with the motion mask. Since both the background and the tracked object are moving, a background stabilizing step [32] is applied to estimate the motion mask $\Delta\mathcal{I}$. The stabilization is based on the idea that the background movement can be modeled as a planar affine transform.

3. Gradient Orientation Cues

A gradient orientation map $\mathcal{OI}$ provides the orientations of edges. Gradient orientation is a useful feature to discriminate the real object boundaries from all the detected edges, especially when background clutter is present. If we denote the normal orientation of the true boundary as ϕ, it is expected that the local normal orientation should present a Gaussian distribution with mean equal to ϕ [33], While the normal orientation distributions of pixels not on the boundary tend to be a uniform distribution between $[0, 2\pi)$. Figure 4(b) illustrates the different distributions generated by the pixels on the contour and those not on the contour. This leads to the definition of the orientation probability map as

$$P_t(Y_o|\hat{\mathbf{Q}}) \propto \exp\left\{-\frac{[\mathcal{OI}_t(\mathbf{x}) - \hat{\phi}_{t-1}(l)]^2}{\sigma_o^2}\right\} \quad \forall\, \mathbf{x} \in \mathcal{R}(\mathbf{n}_l), \tag{23}$$

where $\mathcal{R}(\mathbf{n}_l)$ defines a proximity region to $\mathbf{n}_l$:

$$\mathcal{R}(\mathbf{n}_l) \triangleq \{\mathbf{y} \in \mathcal{R}(\mathbf{n}_l) : \text{dist}(\mathbf{y}, \mathbf{n}_l) \leq \text{dist}(\mathbf{y}, \mathbf{n}_k), k \neq l\}. \tag{24}$$

4. Shape Template Cues

The shape of a tracked object always has certain pattern. Therefore the shape template could be used as one cue, indicating the probability that each image pixel belongs to the real-object contour. Our shape template is different from the static shape energy proposed by Cremers *et al.* [34], which is pretrained and remains the same during tracking. We use an online model that incorporates a dynamic part that varies according to the observations (transformed contour by global motion estimation). The

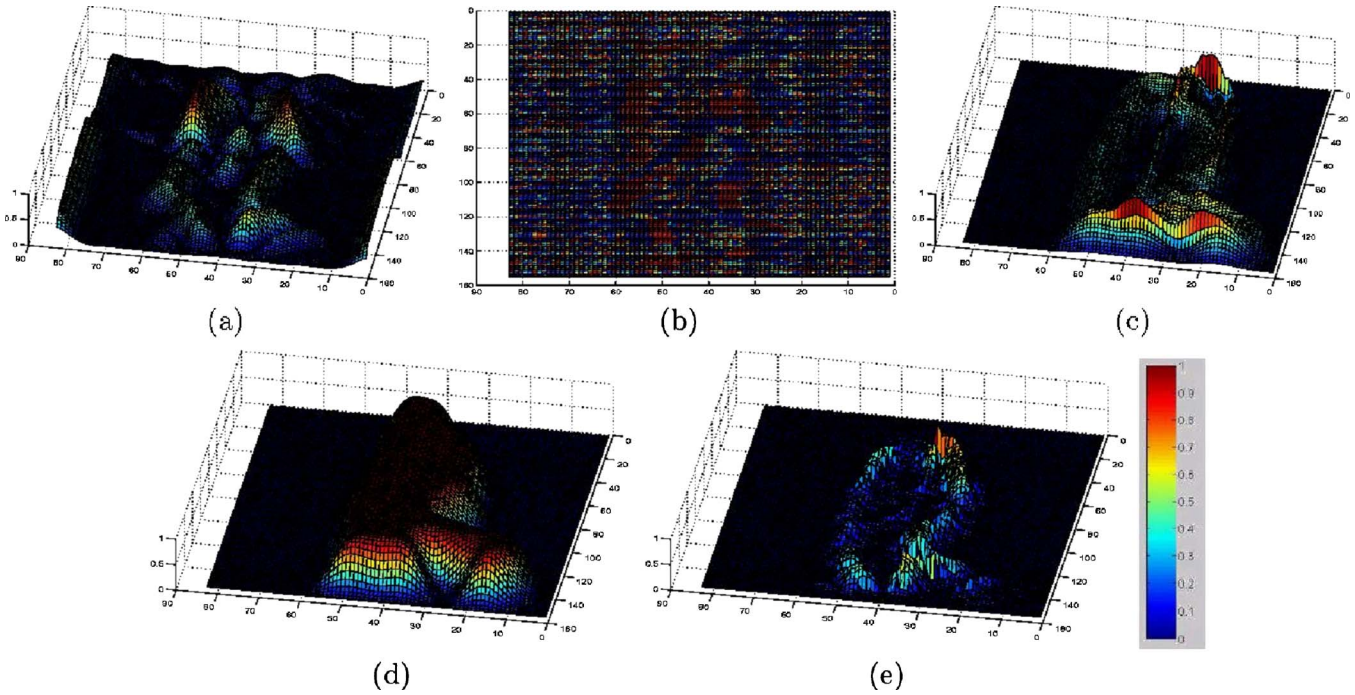

Fig. 4. (Color online) Illustration of fusion from different visual sources, including the probability maps of (a) gradient magnitude, (b) gradient orientation, (c) shape template, (d) foreground, (e) fusion.

(1) Using the regular edge map *(2) Using the feature maps $\mathcal{FI}$*

Fig. 5. (Color online) Performance comparisons on a cluttered scene. A stabilization step is applied to obtain the second set of results to obtain $\mathcal{FI}$ because the sequence was acquired by a moving camera.

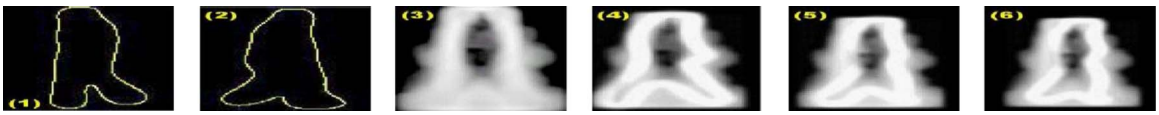

Fig. 6. (Color online) Illustrations of adaptive probability shape templates for pedestrians. (1) and (2), shape training samples; (3), shape prior model; (4), (5), and (6), examples of probability shape templates in different frames.

shape template observation Y_s contains the shape prior model $\mathcal{A}_{\mathbf{S}}$ and the dynamic template $\mathcal{A}_{\hat{\mathbf{Q}}}$. The former is a static template created from the training data, and the latter follows a Gaussian distribution, the mean of which is set to the transformed contour $\hat{\mathbf{Q}}$. The probability is given by

$$P_t(Y_s|\hat{\mathbf{Q}}) = a_t \mathcal{A}_{\mathbf{S}} + (1 - a_t)\mathcal{A}_{\hat{\mathbf{Q}}}, \tag{25}$$

where $0 < a_t < 1$ is the weight that controls the integration of $\mathcal{A}_{\mathbf{S}}$ and $\mathcal{A}_{\hat{\mathbf{Q}}}$. Figure 4(c) is an example of the probability map from the shape template cue. We show more examples of probability templates in Fig. 6.

The construction of $\mathcal{A}_{\mathbf{S}}$ is straightforward. All the sample images in the training data are first normalized to the same template size. Normalization here is defined as translating each sample so that the centroid of the contour sample matches the geometrical center of the sample image. After normalization, for each pixel of the template, the probability $p(x,y)$ of it being on a contour is calculated by counting the occurrence of it belonging to the contour in each sample, and then normalizing to a probability.

5. Foreground Cues

To suppress the contamination from background clutter, we use a foreground probability map $P_t(Y_f|\hat{\mathbf{Q}})$ that estimates the likelihood of a pixel belonging to the tracked object. This map is calculated by comparing the current frame to a set of background models representing the static parts of the scene. The pixelwise background models are updated by the previous values for static camera setups. For moving cameras, these models can be fitted after consecutive frames are aligned on a mosaic by global motion estimation. We define the background as layers of multivariate Gaussian functions $\{(\mu_t^i, \mathbf{\Sigma}_t^i, \kappa_t^i, \upsilon_t^i)\}_{i=1\ldots\mathcal{K}}$ where μ_t^i is the posterior mean, $\mathbf{\Sigma}_t^i$ is the marginal posterior covariance, υ_t^i is the degrees of freedom, κ_t^i is the number of prior measurements of the ith layer, and $\mathcal{K}$ is the number of layers in 3D color space. At each frame, we update the layer parameters using an online Bayesian estimation method as described in [35]. We order the layers according to confidence scores. Our confidence measure is inversely proportional to the determinant of covariance. Then we select the layers having confidence value greater than a layer threshold. We measure the Mahalanobis distance of observed color $I(\mathbf{x})$ from the layers,

$$d_i(\mathbf{x}) = (I(\mathbf{x}) - \mu_{t-1}^i)^T (\mathbf{\Sigma}_{t-1}^i)^{-1} (I(\mathbf{x}) - \mu_{t-1}^i), \tag{26}$$

and update the parameters of the confident layers. Pixels that are outside the 99% confidence interval of all confident layers of the background are considered as foreground pixels. After the update, the foreground probability map at a pixel is determined as

$$P_t(Y_f|\hat{\mathbf{Q}}) = \alpha \exp\left(-\min_{i=1}^{\mathcal{K}} d_i(\mathbf{x})\right), \tag{27}$$

where α is a normalizing constant. Figure 4(d) shows an example of the probability map based on the foreground cue.

5. REGULATION ON SHAPE DEFORMATION

Based on the estimated global motion determined by $\hat{\theta}$ and local deformation, we could obtain the deformed contour $\hat{\mathbf{r}}$ and therefore track the contour from frame to frame. However, what if the deformation estimate is severely corrupted by noise? For example, the exact contour points may not always coincide with maximum probability pixels but may be present at weaker secondary pixels. In such cases, shape regulation may be used as a constraint to recover from errors. The intuition is that learning the prior shape knowledge of the object from the training set could help in delineation. Ideally, the training samples should cover all deformation variations. If an object in one frame exhibits a particular type of deformation not present in the training set, the system searches for the deformation in the subspace that is closest to the target; i.e., the system projects any deformation onto the subspace. The regulation is therefore achieved.

A. Generic Shape Model

There are several approaches to subspace construction. Cootes and Taylor introduced the active shape model [36] based on the points distribution model (PDM) described in [37] for obtaining the shape subspace. Our training method is based on the idea of PDM. A shape model is defined in terms of x and y coordinates of every "landmark" point lying on the outline of the target. The number of landmark points is fixed at equal intervals along the contours. The control points of B-splines are regarded as these landmark points. Table 2 gives the steps for training a prior model from a set of N samples, each represented by a set of columnized L control points $\mathbf{Q}_{\mathbf{S}}^i = \{q_s^{(i,j)} | 1 \le i \le N, 1 \le j \le L\}$.

B. Adaptive Shape Model

The constructed shape model is a generic model that can apply to all cases as long as a target belongs to the corresponding object category. However, what we really need is a deformation model that more accurately represents the shape variations in the sequence being considered. One solution is to update the existing principal components analysis (PCA) model with the initial contour of the current sequence (either manually marked or automatically detected) [38]. Denoting x_0 as the aligned initial contour and the vector $\mathbf{b}_s = \mathbf{P}^T(\mathrm{x}_0 - \bar{\mathrm{x}})$ as the subspace component, the projection residue is obtained as

$$\mathbf{x}_r = \mathbf{x}_0 - \bar{\mathbf{x}} - \mathbf{P}\mathbf{b}_s. \tag{28}$$

The residue part represents the shape variation that is not being accounted for by the prior model, so the generic subspace $(\bar{\mathbf{x}}, \mathbf{P}, \Lambda_t)$ is updated by the following equations:

$$\bar{\mathbf{x}}^* = \beta\bar{\mathbf{x}} + (1-\beta)\mathbf{x}_r, \tag{29}$$

$$\mathbf{e}_r = \frac{\mathbf{x}_r^T}{\|\mathbf{x}_r\|}(\mathbf{x}_0 - \bar{\mathbf{x}}), \tag{30}$$

$$\mathbf{C}^* = \beta\begin{bmatrix}\Lambda_t & \mathbf{0}\\ \mathbf{0}^T & 0\end{bmatrix} + \beta(1-\beta)\begin{bmatrix}\mathbf{b}_s\mathbf{b}_s^T & \mathbf{e}_r\mathbf{b}_s\\ \mathbf{e}_r\mathbf{b}_s^T & \mathbf{e}_r^2\end{bmatrix}, \tag{31}$$

where β is the update weight. By applying singular value decomposition to Eq. (31), we obtain $(\mathbf{P}^*, \Lambda_{t+1}^*)$ satisfying $\mathbf{P}^*\Lambda_{t+1}^*(\mathbf{P}^*)^T = \mathbf{C}^*$. The above formulas are extensions of incremental PCA or the eigenspace merging formula of [39] with tunable update weight β between the old and the current data. For brevity, we still use $(\bar{\mathbf{x}}, \mathbf{P}, \Lambda_t)$ to denote the updated shape subspace in the rest of the paper.

C. Subspace Projection

The projection of deformation can be described as representing the deformed contour by a linear combination of basis in the shape subspace. We first align the deformed contour point vector set $\hat{\mathbf{r}}_t$ to $\mathbf{x}_t$, and then apply

$$\mathbf{x}_{p,t} = \mathbf{P}\mathbf{P}^T(\mathbf{x}_t - \bar{\mathbf{x}}) + \bar{\mathbf{x}}, \tag{32}$$

where $\mathbf{x}_{p,t}$ is a linear combination of subspace basis. It is possible that some control points on $\hat{\mathbf{r}}$ may be occluded or not detected along the normal lines. Let us denote the index set of detected points as $I_d = \{i_1, i_2, \ldots\}$. We can recover a complete projected contour as follows:

$$\mathbf{x}_{p,t} = \mathbf{P}\mathbf{P}_{I_d}^{\dagger}(\mathbf{x}_{I_d,t} - \bar{\mathbf{x}}_{I_d}) + \bar{\mathbf{x}}, \tag{33}$$

$$\mathbf{P}_{I_d}^{\dagger} = (\mathbf{P}_{I_d}^T\mathbf{P}_{I_d})^{-1}\mathbf{P}_{I_d}^T. \tag{34}$$

A projection example is demonstrated in Fig. 7 indicating a comparison between tracking results with and without shape regulation. Apparently, using the subspace can preclude the contour from deforming to an irregular shape.

Table 2. Algorithm 2: Construct Prior Shape Model

(a) Align the set of examples into a common frame of reference, $\mathbf{x}^i$=aligned $(\mathbf{Q}_s^i)$ [37];
(b) Calculate the mean of the aligned examples $\bar{\mathbf{x}}$, and the deviations $\delta\mathbf{x}^i = \mathbf{x}^i - \bar{\mathbf{x}}$;
(c) Calculate the eigensystem of the covariance matrix of the deviations, $\mathbf{C} = \frac{1}{N}\Sigma_{i=1}^{N}(\delta\mathbf{x}^i)(\delta\mathbf{x}^i)^T$.
(d) The first t principal eigenvectors of the eigensystem are used to generate $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}$, where $\mathbf{b}$ is a t-element vector of shape variation parameters and $\mathbf{P}$ is a $2L \times t$ matrix of t eigenvectors, which composes the estimated shape subspace. We denote the eigenvalue diagonal matrix as Λ_t,which is a $t \times t$ matrix.

D. Alignment

The process of alignment is to normalize the contour, because the shape subspace is constructed from normalized training samples. It follows the method for rigid shape matching proposed by Cootes and Taylor [36]. The basic idea is to find a transformation matrix (containing the rotation, translation, and scale coefficients) and match the processing contour to the mean of the shape model. An example is shown in Fig. 8. The sequence is collected by magnetic resonance imaging (MRI) of human knees. Our primary interest is in the articular cartilage layer. Figure 8(1) depicts the detected contour pixels. There are some pixels that are too obscure to be detected. Figure 8(2) shows the projected contour using the shape subspace without alignment. Figure 8(3) depicts the contour pixels after alignment, and Fig. 8(4) gives the final result.

6. IMPLEMENTATION AND EXPERIMENTS

A. Algorithmic Implementation

A summary of the complete contour tracking algorithm is given in Table 3. We want to further discuss the initial-

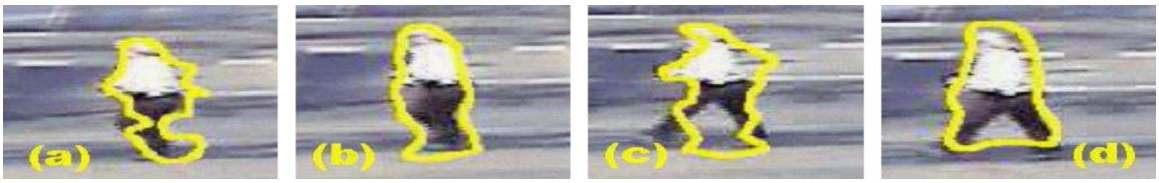

Fig. 7. (Color online) Comparisons of tracking results with and without subspace regulation: (a) and (c) are without regulation, (b) and (d) are with regulation.

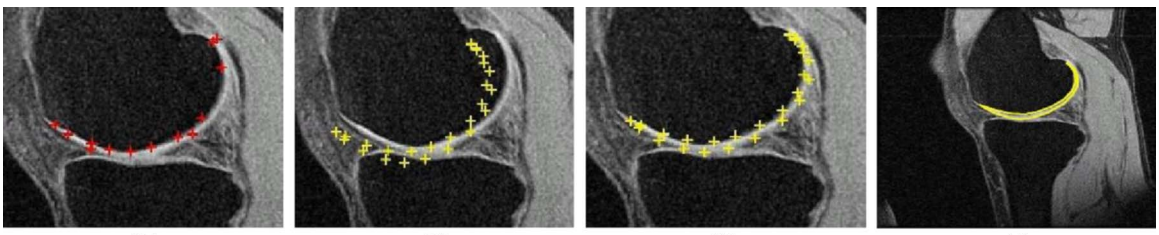

Fig. 8. (Color online) One example frame from an MRI sequence to illustrate shape alignment. The target of interest is the articular cartilage layer. (1) Detected contour pixels, (2) recovered contour pixels using the shape subspace projection, (3) contour pixel after alignment, (4) final result.

ization step. The easiest way to acquire the initial contour is by manually sketching it around the object of interest in the first frame. The pixels along the contour are sorted in clockwise order. The control points are then selected based on the uniform-arc-length rule, which provides the initial contour. We can also apply automatic shape detection methods, such as the direct use of probability shape template to detect pedestrians [40]. A nice property of our tracking algorithm is that it has a high tolerance to localization errors in the initial estimate. In our experiments, we observed that a very approximate but reasonable initial guess can evolve to capture the real boundary of the object of interest in three to six frames, as demonstrated in Fig. 9.

Table 3. Algorithm 3: Active Contour Tracking

Step1. Initialization: Draw a set of particles from the prior $p(\theta_0)$ to obtain $\{\theta_0^{(j)}, \omega_0^{(j)}\}, j=1,\ldots,J$, where J is the number of the particles. Get the initial control point set $\{\mathbf{Q}_0^{(j)}\}$ from θ_0. Set $t=1$.

Step2. Global motion estimation:

Step2.1 Prediction: Estimate the state vector shift ν_t, draw particles $\{\theta_t^{(j)}\}$ and accordingly the control point set samples $\{\mathbf{Q}_t^{(j)}\}, j=1,\ldots,J$.

Step2.2 Update: Calculate the likelihood function $\mathcal{L}(Y_t|\theta_t^{(j)})$ and the posterior $\pi_t^{(j)}=p(\theta_t^{(j)}|Y_{1:t})$ for each sample, then normalize $\{\pi_t^{(j)}\}$ and update $\{\theta_t^{(j)}, \omega_t^{(j)}\}, j=1,\ldots,J$. Find the MAP estimator of the global motion $\hat{\theta}_t=\theta_t^{\arg\max_{j\in\{1,\ldots,J\}}\pi_t^{(j)}}$ and the corresponding $\hat{\mathbf{Q}}_t$.

Step3. Local Deformation Estimation: Based on the estimated $\hat{\mathbf{Q}}_t$, generate the deformation probability map P_t. The deformed contour $\hat{\mathbf{r}}_t$ can be determined by scanning along the adaptive normal lines $\mathbf{n}_{l,t}$ for pixels with maximum deformation probabilities.

Step4. Regulation: Project $\hat{\mathbf{r}}_t$ onto the shape subspace to acquire the final estimated contour.

Step5. $t\rightarrow t+1$, go to step.2.

B. Experimental Results

We applied the sequential contour tracker to different sets of outdoor surveillance video sequences containing moving people and vehicles. All the objects of interests are assumed to be objects moving in nonrigid forms. In most cases, the backgrounds contain clutter that undermines the tracker's performance. Among them, four sets of sequences were captured by moving cameras, which means that in computing the probability maps, the foreground cue is not involved. The other two sets of sequences were captured using static cameras, so the foreground cue is used for generating the deformation probability map. The algorithm speed implemented by C++ code differs from 6 to 10 frames per second on a 1.5 GHz personal computer running Windows XP, depending on different frame sizes. In our experiments, most comparisons are made between our method and the traditional method described in [1].

1. Experiment on Stationary Camera Data

Figure 10 shows a sequence (the frame size is $437\times90\times3$) acquired by a stationary camera. It contains a pedestrian walking through the scene. The traditional active contour tracker does not work, for two possible reasons: (1) the normal lines on some of the control points do not detect the edge points, and (2) in some frames the contour gets intertwined. Our method takes advantage of the adaptive normal-line strategy to avoid the intertwined contour, and subspace projection to recover missing edge points. The proposed method achieves satisfactory result throughout 159 frames that contain the object.

2. Experiments on Moving Camera Data

Figures 11, 13, 14, and 15 demonstrate some typical results for moving camera sequences.

Figure 11 shows a sequence (the frame size is $543\times814\times3$) capturing a moving SUV by a camera from another vehicle following it. Although the rear view of the vehicle is a rigid object, the surrounding disturbances and

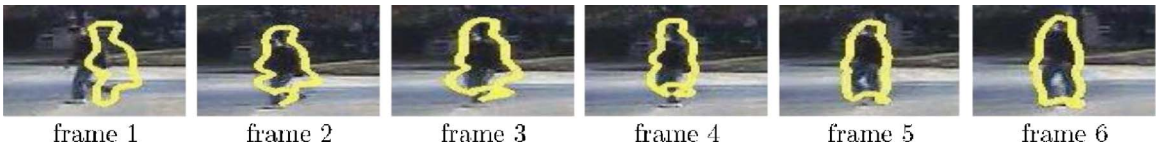

Fig. 9. (Color online) Example of starting tracking using a very rough initial contour. After tracking for six frames, we find that the contour has been attached to the object fairly precisely.

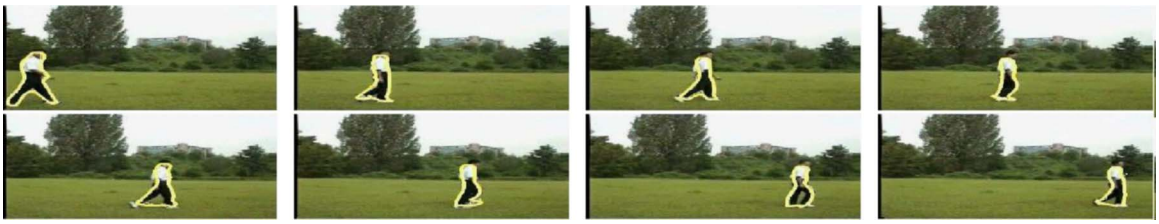

Fig. 10. (Color online) Pedestrian tracking performance with a stationary camera. The thick light (yellow online) outlines represent the final tracked contours.

Fig. 11. (Color online) Results of tracking a vehicle from behind. The comparison result is given in Fig. 5.

Fig. 12. (Color online) Comparisons of the traditional method and our proposed method on a sequence with both background and object moving. Upper: using the traditional method; Lower: using the proposed method.

Fig. 13. (Color online) Sequence with both background and object moving. The challenges of processing this sequence are due to the following: (1) camera motion, (2) background clutter, (3) the pants and the shirt that the pedestrian is wearing are of very different colors.

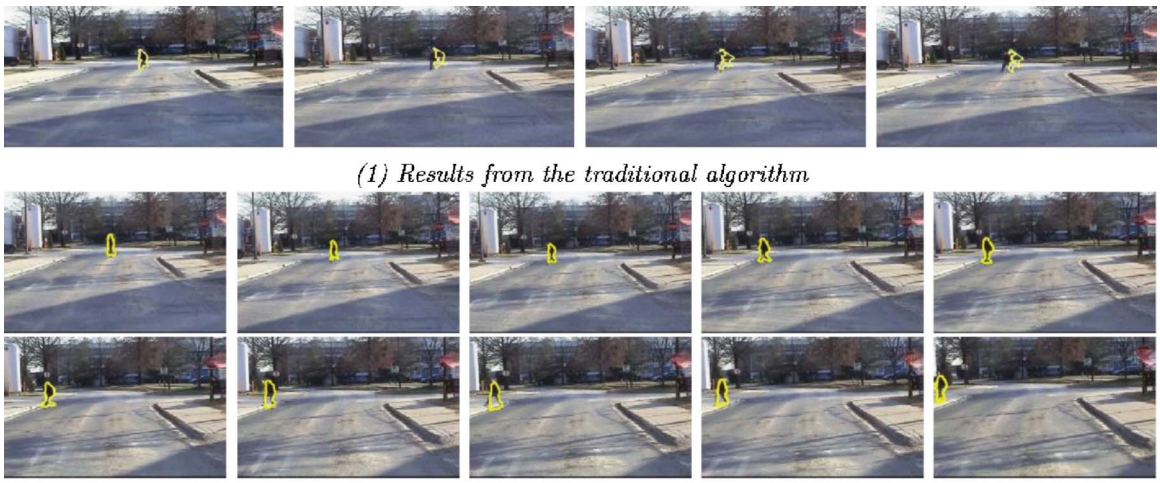

Fig. 14. (Color online) Sequence with both background and object moving. The background is heavily cluttered, and the intensity of the tracking pedestrian is very similar to the background color. A comparison result from the traditional approach is also given.

small view changes throughout the video lead to failures when the traditional rigid-object trackers are used. Two major distractions are from plain road marks and the sudden appearance of another vehicle in front of our target. Still, we obtain good results due to the incorporation of a deformation map based on several statistical cues.

Figures 12–14 illustrate three challenging sequences with walking pedestrians crossing the road. The frame size in all three sequences is $541\times818\times3$. Tracking difficulties arise as a result of the following: (1) The camera is moving forward very fast, and therefore the global motion of the pedestrian includes not only translation and rotation but zoom as well. (2) The background is full of road marks and shadows. The traditional contour trackers get distracted by these background clutters. (3) The pedestrians in Figs. 12 and 13 wear a white shirt and black pants. The strong contrast between the two parts usually leads to shrinking of the tracking result to either the upper part or the lower part of the body. (4) In Fig. 14, the object has a color very similar to that of the background (woods). The results of our tracker are satisfactory. We notice that the poses of the pedestrians vary significantly from frame to frame in both sequences. However, tracking remains robust.

Figure 15 gives an example with a sequence containing a moving truck (the frame size is $480\times720\times3$). Although a truck cannot be taken as a nonrigid object, we still observe 2D shape deformation on the truck due to its 3D rotation. This sequence demonstrates the advantage of using contour-based tracking. The truck in the sequence changes views from the back view to the side view, which means that some of the corresponding points on the object may disappear in the scene. Such a view change will strongly undermine the result of a regular 2D appearance-based tracker unless a 3D appearance model is used. As mentioned in Section 1, contour-based tracking can always yield robust results without requiring correspondences and 3D reconstructions. The results are

promising even with the presence of many challenges, such as obscure boundaries, low color contrast, nonstationary camera, and background clutter.

3. Experiments on Occlusion Data

We exploit the application of shape regulation to recover occluded contours, the results of which are shown in Fig. 16. This sequence (the frame size is $576\times768\times3$), acquired by a stationary camera, contains walking humans. In the last several frames, the pedestrian is partially occluded by trees. With subspace projection, we see that the occluded parts have been reconstructed. In this sequence, the clutter is heavy due to the presence of parked cars and trees in the scene.

4. Experiments on Medical Sequence

The MRI sequence in our experiment consists of 2D image slices that form a 3D image cube for subsequent 3D visualization, in which we are particularly interested in the articular cartilage layer (a very thin, white, crescentlike layer). The difficulties of tracking these layers are due to surrounding tissues often having similar intensity values, which leads to some boundary points becoming nearly undetectable; further, the sequence is of low resolution due to the preprocessing step that enlarges the original image. The traditional active contour method is not effective in this case, because it lacks means to handle edge point occlusion. The "snake" method also has difficulty in finding the correct boundary due to the similar intensity values between the cartilage layer and the tissues surrounding it. Figure 17 demonstrates the tracking results of applying the proposed algorithm to the MRI sequence, in which the frame size is 584×584. As a comparison, we also provide a set of tracking results using the traditional contour tracker. It can be seen that our method performs well.

5. Performance Evaluation

We use the mean sum of squared distance (MSSD) [41] measure to evaluate the tracking performance of different algorithms. For a sequence with K frames, where the contour $\mathbf{r}_k$ in each frame has L control points, $\{(x_{k,1},y_{k,1}),\ldots,(x_{k,m},y_{k,m})\}$, we define

$$\mathrm{MSSD}=\frac{1}{K}\sum_{k=1}^{K}\frac{1}{L}\sum_{j=1}^{L}(x_{k,j}-x_{k,j}^{0})^2+(y_{k,j}-y_{k,j}^{0})^2, \quad (35)$$

where $[x^0,y^0]$ represents the corresponding ground truth. We compare the proposed algorithm with the active contour tracking method for two cases: sequences with heavily cluttered backgrounds and sequences with strong nonrigid movements. Table 4 shows the values of the MSSD and the variance of the squared distance in these two cases, respectively. From all the experimental results and the comparison table, we find that the sequential tracker performs well in most of the cases.

7. DISCUSSION

The sequential contour tracker we have presented is motivated by the fact that nonrigid movement can be decomposed into a global motion and a local deformation. The algorithm contains three major steps: motion estimation, deformation estimation, and shape regulation. The following discussion covers the major aspects of the algorithm.

Multistep versus Single-Step. Compared with most methods using single-step estimation [1,2,10] to obtain nonrigid contour movement with motion and deformation simultaneously, we choose a sequential framework to estimate motion and deformation separately. One-step estimation presents more systematic formulas, but it suffers from high-dimensional computation and poor efficiency. Fortunately, the multistep approach characterizes an efficient solution. The explanation is as follows: We use an affine transform to model the global motion. Therefore the motion state vector is only six dimensional, which eases the burden on the particle filter. We interpret shape deformation by the deformations occurred on control points, which approximates the infinite-dimensional representation by a finite-dimensional one. Since the processing is carried out in a sequential manner and a rough contour is obtained by motion estimation, the deformation is searched only in adjacent regions of the rough contour and does not involve numerical simulations (particle filter). Thus, the complexity of the entire computation is reduced.

Multicue versus Single-Cue. Although the gradient

Fig. 15. (Color online) Airborne sequence with a white truck moving on the ground. The truck shows a back view in frames (1) and (2), then shows the back-right view in frame (3), a side view in frames (4) and (5) and a front-right view in frame (6). This experiment demonstrates that the contour-based tracker can give satisfactory results on sequences containing 3D object rotations.

Fig. 16. (Color online) Example of an occluded contour being recovered by shape subspace projection. In the last three frames, the pedestrian is partially occluded by the surrounding trees. Our tracking result recovers the partially occluded contour. Bright (red online) arrows indicate the occluded parts. We may also note that the parked cars contribute to background clutter.

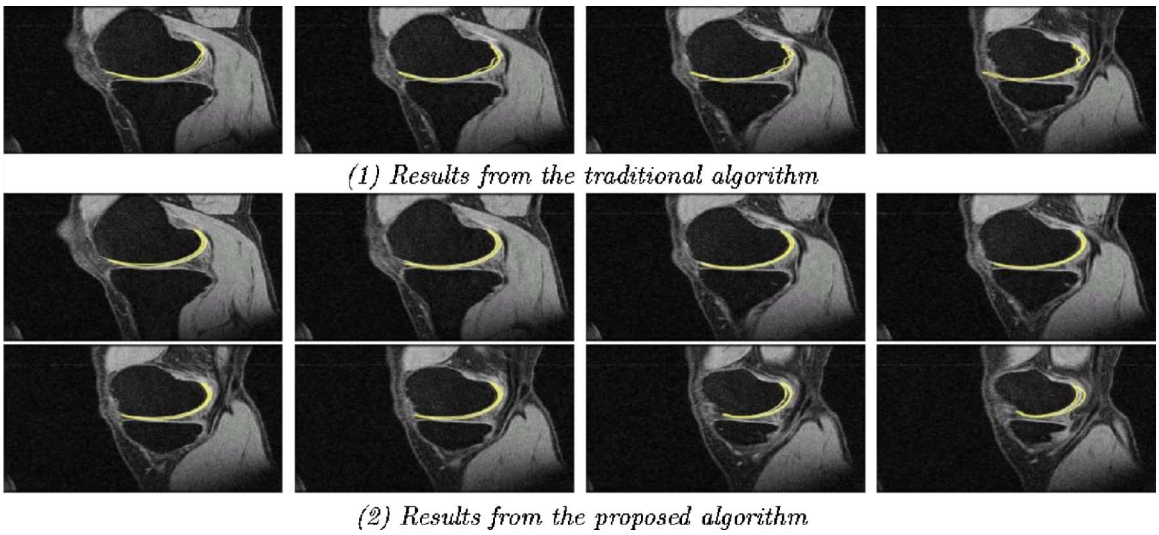

Fig. 17. (Color online) Magnetic resonance imaging scans of human knees. The area of interest is the articular cartilage in each image. The upper row demonstrates the results from application of the traditional algorithm. The middle and bottom rows show the results from application of our proposed method.

Table 4. Comparison of Tracking Results in Heavy-Cluttered Background Sequences and Nonrigid Object Sequences

	ACT[a]		Sequential Tracker	
Sequence	MSSD[b]	VAR[c]	MSSD	VAR
Heavy-cluttered background	14.7584	74.4052	7.9129	12.5602
Nonrigid object	9.9741	24.6597	6.0571	8.5222

[a]ACT is the traditional active contour tracking algorithm.

[b]MSSD is the mean sum of the squared distance.

[c]VAR is the variance of the distances.

magnitude is a very strong cue to estimate the boundary pixels, sometimes it is not robust. In our method, deformation estimation counts more on visual cues with the aim of getting more robust contour detection. We could further improve the fusion method by associating adaptive fusion weights with different cues [42].

Adaptive versus Nonadaptive Normal Line. We adaptively set the scanning normal lines according to prior shape pose variations and previous shape estimate. This reduces the probability of loop occurrence and increases the possibility of correct detection for real boundary pixels, while reducing the chance for false detection.

Regulation versus Nonregulation. Regulation is important for nonrigid contour tracking. It not only constrains shape deformation and corrects estimating errors but recovers the occluding contour pixels as well. Therefore the method is more applicable to situations with nonrigid object movements and heavily cluttered backgrounds.

Our method can successfully track nonrigid objects and get tight contours enclosing the changing shapes of the targets throughout the sequences. We are currently working on extensions of the algorithm to the multitarget tracking problem. Model regulation is also worth further exploitation. We used only the shape prior knowledge as a constraint in this paper. However, training samples contain not only the shape but appearance and motion information as well. Constructing a prior model based on all information to improve the robustness of tracking will be an interesting problem.

ACKNOWLEDGMENTS

Supported by the U.S. Army Research Office under Multidisciplinary University Research Initiative (MURI) grant ARMY-W911NF0410176. The technical monitor is Tom Doligalski. The work of Jie Shao was partially supported by a summer internship at Mitsubishi Electric Research Laboratories, Cambridge, Massachusetts, USA.

REFERENCES

1. M. Isard and A. Blake, "CONDENSATION: conditional density propagation for visual tracking," Int. J. Comput. Vis. **29**, 5–28 (1998).
2. P. Li, T. Zhang, and A. E. C. Pece, "Visual contour tracking based on particle filters," Image Vis. Comput. **21**, 111–123 (2003).
3. M. Kass, A. Witkins, and D. Terzopoulos, "Snakes: active contour models," Int. J. Comput. Vis. **1**, 321–331 (1988).
4. F. Leymarie and M. D. Levine, "Tracking deformable objects in the plane using an active contour model," IEEE Trans. Pattern Anal. Mach. Intell. **15**, 617–634 (1993).
5. K. P. Nikos and D. Rachid, "A PDE-based level-set approach for detection and tracking of moving objects," in *Proceedings of the International Conference on Computer Vision*, (IEEE Computer Society, 1998), p. 1139.
6. M. Rousson and N. Paragios, "Shape priors for level set representations," in *Proceedings of the European Conference on Computer Vision 2002* (Springer, 2002), pp. 78–92.
7. G. Sapiro, *Geometric Partial Differential Equations and Image Analysis* (Cambridge U. Press, 2001).
8. A. Yilmaz, X. Li, and M. Shah, "Contour-based object

tracking with occlusion handling in video acquired using mobile cameras," IEEE Trans. Pattern Anal. Mach. Intell. **26**, 1531–1536 (2004).
9. J. Jackson, A. J. Yezzi, and S. Soatto, "Tracking deformable moving objects under severe occlusions," in *Proceedings of the IEEE Conference on Decision and Control* (IEEE, 2004), pp. 2990–2995.
10. Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi, "Particle filtering for geometric active contours with application to tracking moving and deforming objects," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE Computer Society, 2005), Vol. 2, pp. 2–9.
11. A. Yezzi and S. Soatto, "Deformation: deforming motion, shape average and the joint registration and approximation of structures in images," Int. J. Comput. Vis. **53**, 153–167 (2003).
12. J. Foley, A. van Dam, S. Feiner, and J. Hughes, *Computer Graphics Principles and Practice*, (Addison-Wesley, 1990), Chap. 13, pp. 563–604.
13. R. E. Kalman, "A new approach to linear filtering and prediction problems," Trans. ASME **82** Ser. D, 35–45 (1960).
14. A. Doucet and N. de Freitas, *Sequential Monte Carlo Methods in Practice* (Springer-Verlag, 2001).
15. A. C. Sankaranarayanan, R. Chellappa, and A. Srivastava, "Algorithmic and architectural design methodology for particle filters in hardware," in *Proceedings of the International Conference of Computer Design (ICCD)* (2005), pp. 275–280.
16. M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," IEEE Trans. Signal Process. **50**, 174–188 (2002).
17. N. J. Gordon, D. J. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," IEEE Proc. Radar Signal Process. **140**, 107–113 (1993).
18. S. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," IEEE Trans. Image Process. **13**, 1491–1506 (2004).
19. D. Metaxas and D. Terzopouilos, "Shape and nonrigid motion estimation through physics-based synthesis," IEEE Trans. Pattern Anal. Mach. Intell. **15**, 580–591 (1993).
20. G. Xu, E. Segawa, and S. Tsuji, "A robust active contour model with insensitive parameters," in *Proceedings of the International Conference on Computer Vision* (1993), pp. 562–566.
21. P. J. Phillips, S. Sarkar, I. Robledo, P. Grother, and K. W. Bowyer, "The gait identification challenge problem: data sets and baseline algorithm," in *Proceedings of the International Conference on Pattern Recognition* (International Association for Pattern Recognition, 2002), Vol. 1, pp. 385–388.
22. A. Jain, *Fundamentals of Digital Image Processing* (Prentice Hall, 1989), Chap. 9.
23. P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proceedings of the 9th IEEE International Conference on Computer Vision* (IEEE Computer Society, 2003), Vol. 2, pp. 734–741.
24. K. Toyama and G. Hager, "Incremental focus of attention for robust vision-based tracking," Int. J. Comput. Vis. **35**, 45–63 (Jan 1999).
25. H. Sidenbladh, M. J. Black, and D. J. Fleet, "Stochastic tracking of 3D human figures using 2D image motion," in *Proceedings of the European Conference on Computer Vision 2000* (Springer, 2000), Vol. 2, pp. 702–718.
26. J. M. Odobez and D. Gatica-Perez, "Embedding motion in model-based stochastic tracking," in *Proceedings of the International Conference on Pattern Recognition 2004* (International Society for Pattern Recognition, 2004), Vol. 2, pp. 815–818.
27. B. Birchfield, "Elliptical head tracking using intensity gradients and color histogram," in *Proceedings of the International Conference on Pattern Recognition 1998 Computer Vision and Pattern Recognition*, (International Society for Pattern Recognition, 1998), pp. 232–237.
28. J. Triesch and C. van der Malsburg, "Democratic integration: Self-organized integration of adaptive cues," Neural Comput. **13**, 2049–2074 (2001).
29. J. Vermaak, P. Pérez, M. Gangnet, and A. Blake, "Towards improved observation models for visual tracking: selective adaptation," in *Proceedings of the European Conference on Computer Vision 2002* (Springer, 2002), Vol. 1, pp. 645–660 (2002).
30. P. Pérez, J. Vermaak, and A. Blake, Proc. IEEE **92**, 495–513 (2004).
31. P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," IEEE Trans. Pattern Anal. Mach. Intell. **12**, 629–639 (1990).
32. Q. Zheng and R. Chellappa, "A computational vision approach to image registration," IEEE Trans. Image Process. **2**, 311–326 (1993).
33. J. M. Coughlan and S. J. Ferreira, "Finding deformable shapes using loopy belief propagation," in *Proceedings of the European Conference on Computer Vision* (Springer, 2002), Vol. 3, pp. 453–468.
34. D. Cremers, T. Kohlberger, and C. Schnörr, "Nonlinear shape statistics in Mumford-Shah based segmentation," in *Proceedings of the European Conference on Computer Vision* (Springer, 2002), Vol. 2, pp. 93–108.
35. F. Porikli and O. Tuzel, "Bayesian background modeling for foreground detection," in *Proceedings of the ACM Visual Surveillance and Sensor Network* (Association for Computing Machinery, 2005), Vol. 3, pp. 55–58.
36. T. F. Cootes and C. J. Taylor, "Active shape models," in *Proceedings of the 3rd British Machine Vision Conference*, D. Hogg and R. Boyle, eds. (Springer, 1992), pp. 266–275.
37. T. F. Cootes and C. J. Taylor, "Combining point distribution models with shape models based on finite-element analysis," in *Proceedings of the 5th British Machine Vision Conference*, E. Hancock, ed., (British Machine Vision Association, 1994), pp. 419–428.
38. X. S. Zhou, D. Comaniciu, and A. Gupta, "An information fusion framework for robust shape tracking," IEEE Trans. Pattern Anal. Mach. Intell. **27**, 115–129 (2005).
39. P. Hall, D. Marshall, and R. Martin, "Merging and splitting eigenspace models," IEEE Trans. Pattern Anal. Mach. Intell. **22**, 1042–1048 (2000).
40. H. Nanda and L. Davis, "Probabilistic template based pedestrian detection in infrared videos," in *Proceedings of the IEEE Intelligent Vehicle Symposium* (IEEE, 2002), pp. 15–20.
41. Y. Akgul and C. Kambhamettu, "A coarse-to-fine deformable contour optimization framework," IEEE Trans. Pattern Anal. Mach. Intell. **25**, 174–186 (2003).
42. M. Spengler and B. Schiele, "Towards robust multi-cue integration for visual tracking," Machine Vision Appl. **4**, 50–58 (2003).