Multiple Object Tracking Using Elastic Matching

Xingzhi Luo and Suchendra M. Bhandarkar
Department of Computer Science, The University of Georgia
Athens, Georgia 30602-7404, USA

Abstract

A novel region-based multiple object tracking framework based on Kalman filtering and elastic matching is proposed. The proposed Kalman filtering-elastic matching model is general in two significant ways. First, it is suitable for tracking of both, rigid and elastic objects. Second, it is suitable for tracking using both, fixed cameras and moving cameras since the method does not rely on background subtraction. The elastic matching algorithm exploits both the spectral features and structural features of the tracked objects, making it more robust and general in the context of object tracking. The proposed tracking framework can be viewed as a generalized Kalman filter where the elastic matching algorithm is used to measure the velocity field which is then approximated using B-spline surfaces. The control points of the B-spline surfaces are directly used as the tracking variables in a grid-based Kalman filtering model. The limitations of the Gaussian distribution assumption in the Kalman filter are overcome by the large capture range of the elastic matching algorithm. The B-spline approximation of the velocity field is used to update the spectral features of the tracked objects in the grid-based Kalman filter model. The dynamic nature of these spectral features are subsequently used to reason about occlusion. Experimental results on tracking of multiple objects in real-time video are presented.

1 Introduction and Background

Multiple object tracking is challenging in several aspects. The first challenge comes from mutual occlusion between objects. When occlusion occurs, some objects are partially or totally invisible. This makes it hard to accurately localize the position of the occluded object and track the occluded object continuously. The second challenge is the formulation of a good object model. A good object model should be able to capture the most important and relevant information about the object and facilitate fast and reliable tracking. The ability to deal with occlusion handling depends, to a great extent, on the object model. The third challenge is to meet the real time constraints of most tracking applications in the real world. Fast and accurate object localization over time is the ultimate objective of a tracking system.

Generally speaking, there exist three broad categories of object models in the context of tracking: contour-based models [1, 7, 8], region-based models [2, 3, 4], and feature point-based models [9, 11, 12]. Since the contour-based model does not encode any color or edge information within the interior of the object, the contour information by itself is not enough to handle instances of occlusion. A region-based object model is more suitable when occlusion is present. A simple region-based object model uses the color distribution within the tracked object to represent the object [3, 4], thus making it computationally efficient. However, in the absence of any object shape information, the object tracking is largely dependent on the background model used in the extraction of the object boundaries. Furthermore, when several objects move together, it is not possible to achieve accurate tracking due to occlusion. A grid-based object model can potentially encode more detailed spectral (color) and structural (shape) information of the tracked objects. In their tracking algorithm Isard and MacCormick [2] extract six color features for each grid. Although their scheme encodes the spectral information of the tracked objects, since the grids are not structurally organized, the shape information is not well maintained [2].

Feature point-based tracking is popular on account of its simplicity and relative ease of encoding object shape information. Malik *et al.* [11] and Lepetit *et al.* [12] use corner points whereas Rucklidge [9] uses simple edge points as the feature points of interest in their algorithms. Feature point-based tracking is typically designed to encode the object shape information and is more robust to occlusion. However, in the absence of any spectral information, feature point-based tracking methods are easily distracted by noisy feature points in the background and are, by their very nature, limited to objects rich in feature points. When the image regions that correspond to the tracked objects are largely homogeneous, few salient feature points are available thus causing the robustness of the tracking to suffer.

Occlusion handling is another important issue that arises in multiple object tracking systems and is closely intertwined with the choice of the object model. In the case of contour-based models, the robustness of the occlusion reasoning is highly dependent on the quality of object seg-

mentation and typically, only simple cases are well handled [7]. Region-based object models that rely primarily on color/gray level histograms of the moving regions are not well suited to handle occlusion since no object shape information is available. Correspondence-based schemes [3] typically compute only a statistical probability that a pixel of a given color belongs to a specific object which does not enable accurate object localization.

2 Overview of the Approach

In this paper, a region-based model that combines the Kalman filtering algorithm with elastic matching is proposed for multiple object tracking. Each object is modeled as a network of grids. The color information and the feature points are extracted for each grid. The contour and the object shape information is automatically encoded within the grid network. The Kalman filtering algorithm is used as the velocity prediction model on account of its simplicity. The velocities of the tracked objects are represented using B-spline surfaces. One of the advantages of the proposed model is that the tracked object can possess different velocity (or displacement) vectors in the different image regions that comprise the tracked object. This permits tracking of both rigid and elastic objects. Another advantage of the proposed model is that the approximation of the object contour as a convex contour or by a combination of several elliptical contours as in [7] is not necessary. More elastic objects such as fish can be easily modeled and tracked using the proposed method.

The elastic matching algorithm is used to accurately localize the tracked objects in the proposed model. Elastic matching has been used widely for deformable object recognition [5]. The elastic matching algorithm itself can guide the movement of the elements within the templates in order to achieve a good matching. This feature makes elastic matching potentially well suited for tracking of deformable objects. Moreover, in the event of inaccurate prediction by the Kalman filtering algorithm, the elastic matching algorithm can still guide the tracking towards the optimal solution. This overcomes the limitations of the Gaussian distribution assumption in the Kalman filtering algorithm. Another advantage of elastic matching is that the tracking results are not dependent on the accuracy of the background subtraction used to extract the moving objects. This makes it possible to track moving objects with a moving camera.

The rest of the paper is organized as follows. The overall tracking model is described in Section 3 and the three submodels, i.e., the object model, the prediction model and the velocity measurement model are described in Section 3.1, 3.2 and 3.3 respectively. The B-spline approximation algorithm is described in Section 3.4. Section 4 describes the occlusion reasoning algorithm in detail. Section 5 presents experimental results for the proposed tracking scheme on video data from indoor and outdoor scenes. Section 6 concludes the paper.

3 The Proposed Tracking Model

The proposed tracking model is composed of three submodels: the *object model*, the *velocity estimation model* and the *velocity measurement model*. The *object model* defines the features for object representation and the tracking parameters. The features used in the object model incorporate both color (spectral) and edge (structural) information of the tracked objects. The *velocity estimation model* uses a canonical Kalman filtering algorithm. B-spline surfaces are used to approximate the object velocity field considering that the velocity in the image plane could potentially vary at different pixels belonging to the same object. For example, when a person bows or starts to run, the upper and lower portions of the person's body may undergo different displacements. The control points of the B-spline surfaces are used as estimation variables in the Kalman filtering algorithm. The B-spline approximation smoothes the velocity field obtained via elastic matching, resulting in a velocity field that more closely approximates the real world motion. Using a B-spline surface approximation also allows for a finite number of tracking variables to be used in the Kalman filtering algorithm. The *velocity measurement model* uses a generalized elastic matching algorithm to measure the velocity of each grid. The overall tracking framework is shown in Figure 1. The Kalman filtering algorithm is used to estimate/predict the velocities of the control points on the B-spline surfaces. The estimated control point positions are used to compute the grid velocities, which are then used to initialize the elastic matching algorithm. The elastic matching algorithm determines the new locations of the tracked objects by seeking to optimize an energy function defined over the velocity field. The B-spline surface control points are updated using the velocity field computed by the elastic matching algorithm. The updated B-spline surface control points in turn are used to update the Kalman filter parameters. The details of the proposed tracking scheme are provided in the following subsections.

3.1 The Object Model

Given an image frame $F(t)$, non-overlapping grid cells of size $l \times l$ pixels are imposed on $F(t)$ and the relevant features extracted for each grid cell. The parameter l controls the granularity of the object model. Let $F_G(t)$ denote the grid image at time t, i.e. the set of all grid cells at time t. An object $O(i)$ is modeled as a set of grid cells $O(i) = \{G, G_B\}$, where $G = \{g_k\}$ is the set of grid cells

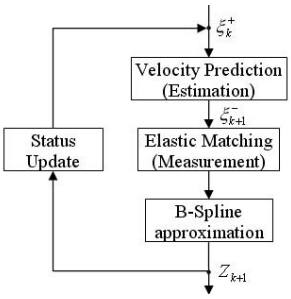


Figure 1: Tracking Model.

that belong to the object and G_B is the set of grid cells that contain the boundary points. Note that $G \subset F_G(t)$ and $G_B \subseteq G$. The boundary grid cells G_B are used to localize the boundary of the tracked object. Henceforth we refer to each grid cell as a grid for the sake of convenience.

Each grid g_k has several attributes given by $g_k = \{X, \xi, c\}$ where $X = (x, y)$ is the location of grid g_k, $\xi = (\xi^x, \xi^y)$ is the velocity of grid g_k and c is the vector of extracted features for grid g_k. Note that each grid has its own velocity vector. B-spline surfaces are used to approximate the velocity field of each object. Color and edge features are extracted for each grid and denoted by the feature vector $c = \{c_i^c, c_i^e\}$, where c^c is the color feature vector and c^e is the set of corner pixels associated with the grid.

Given an RGB color image, three new color channels are computed. The three new color channels are a linear combination of the original RGB channels and are given by $I_1 = (R+G+B)/3, I_2 = R-B, I_3 = 2G-R-B$. These three new color channels are more stable to changes in illumination than the original RGB channels [6]. The channel I_1 denotes the image intensity or luminance whereas channels I_2 and I_3 denote roughly orthogonal color components. For each grid, the color features are the spatial average of each color channel computed over all the pixels within the grid. The color feature for a grid is denoted by $c^c = \{i_1, \sigma_1, i_2, \sigma_2, i_3, \sigma_3\}$ where i_k is the spatial average of color channel k computed over all pixels of the corresponding grid in the current image frame and σ_k is the standard deviation of i_k along the temporal dimension. When there is insufficient temporal information for a grid, σ_k is initialized to a default value of σ_k^0. An online scheme, detailed in Section 4 on occlusion reasoning, is used to update the aforementioned parameters for each grid.

The Harris corner detector [10] is used to extract the corner points. The corner feature for a grid is represented by a set of corner points $c^e = \{c_i^e\}$. If the grid size is small (i.e. 5×5), very few or no corner points are detected within the grid resulting in the absence of any structural information for that grid. Hence in our experiments, the corner points associated with a grid are deemed to be those detected within a window that is larger than the grid size and centered within the grid. Thus, corner points from the neighboring grids are included within the set of corner points associated with a given grid.

The similarity between two grids is quantified by the distance between their associated features. The distance between two color feature vectors is considered to be the Euclidean distance $d(c_1^c, c_2^c) = \|c_1^c - c_2^c\|^2$ whereas the distance between two sets of corner points A and B is the Hausdorff distance $H(A, B)$. The Hausdorff distance has been successfully used as a measure of similarity between two point sets [9]. Given the feature vectors associated with two grids g and g', their feature distance is evaluated as: $d(c_g, c_{g'}) = d(c_g^c, c_{g'}^c) + \beta H(c_g^e, c_{g'}^e)$, where β is a predetermined constant.

3.2 Velocity Estimation Model

A Kalman filter is used to estimate/predict the velocity field of an object. Note that the term velocity field, in our case, actually denotes the control point values resulting from the B-spline approximation of the velocity field. The canonical Kalman filter used in this paper can be described using the following equations:

$$\hat{\xi}_{k+1}^- = \hat{\xi}_k^+ + q_k \quad (1)$$

$$Z_k = \hat{\xi}_k^- + v_k \quad (2)$$

where ξ is the estimated/predicted velocity field. and Z_k is the actually measured velocity field. Equation (1) represents the prior estimation of ξ whereas equation (2) describes the linear relation between the estimated ξ and the actually measured velocity field Z_k. Variables q_k and v_k represent random noise in the prior estimation and actual measurement of the velocity field respectively. Both q_k and v_k are modeled as Gaussian white noise with distributions $\mathcal{N}(0, Q)$ and $\mathcal{N}(0, R)$ respectively.

3.3 Velocity Measurement Model

The Kalman filtering algorithm results in an estimation of the velocities for the control points of an object velocity field, which are used to calculate the velocity (displacement) of each grid from one frame to the next. The estimated displacement of each grid is used to initialize the elastic matching algorithm. The elastic matching algorithm searches for the corresponding location of the tracked object in the new image frame. Given the grids G belonging to an object at time $t-1$, $G \subset F_G(t-1)$, the elastic matching can be viewed as a procedure to determine a mapping $f : G \mapsto G'$ where $G' \subset F_G(t)$ is a set of corresponding grids in the new image frame such that: (i) for each $g \in G$, there exists $g' \in G'$ such that $g' = f(g)$, and (ii) the fol-

lowing energy function is minimized

$$E(f) = \sum_{g \in G} o_g d(c_g, c_{g'}) + \lambda \sum_{(g_1,g_2) \in G} [(X_{g_1} - X_{g_2}) - (X_{g'_1} - X_{g'_2})]^2 \quad (3)$$

In equation (3) o_g denotes the occlusion assumption for grid g: $o_g = 0$ if grid g is occluded, otherwise $o_g = 1$; $g'_1 = f(g_1)$; $g'_2 = f(g_2)$; $d(c_g, c_{g'})$ is the feature distance between grids g and g'; X_{g_k} is the location of grid g_k; and c_g denotes the feature vector associated with grid g. The first summation in equation (3) represents the contribution to the energy function arising from the dissimilarity between the feature vectors associated with grids g and g'. This summation is minimized when the feature vectors associated with grids g and $g' = f(g)$ are similar. The second summation in equation (3) represents the contribution to the energy function arising from the difference in mutual distance between corresponding grid pairs which can be viewed as a measure of object shape distortion. This summation in minimized when the object shape distortion between successive frames is minimized.

By constraining the grid pairs (g_1, g_2) to be neighbors, the computational complexity of the second summation in equation (3) can be significantly reduced and the energy function simplified. The simplified energy function can be written as $E(f) = \sum_{g \in G} E(g)_{g \mapsto g'}$, in which $E(g)_{g \mapsto g'} = o_g d(c_g, c_{g'}) + \lambda \sum_{g_i \in O(g)} \|X_{g_i} - X_g - (X_{g'_i} - X_{g'})\|^2$. Note that $E(g)_{g \mapsto g'}$ is the contribution of the mapping $g \mapsto g'$ to the energy function $E(f)$. Given an initial matching $G \mapsto G'$, an iterative local search is performed to minimize the energy function $E(f)$.

Given grids g and g' (where $g \mapsto g'$), we search for a grid g'' in $O(g')$ (the neighborhood of g'), to see whether the alternative mapping $g \mapsto g''$ can result in a lower value of the energy function. If g' is replaced by g'', the overall change in the energy function is localized to grid g and its neighboring grids $g_i \in O(g)$. The contribution of the mapping $g \mapsto g''$ to the energy function can be evaluated as:

$$E(g)_{g \mapsto g''} = o_g d(c_g, c_{g''}) + \lambda \sum_{g_i \in O(g)} \|X_{g_i} - X_g - (X_{g'_i} - X_{g''})\|^2 \quad (4)$$

The overall change in the energy can be evaluated as:

$$\Delta E(f)_{g' \to g''} = E(g)_{g \mapsto g''} - E(g)_{g \mapsto g'} = \sum_{g_i \in O(G)} (E(g_i)_{g \mapsto g''} - E(g_i)_{g \mapsto g'}) \quad (5)$$

The iterative algorithm outlined below is used to minimize the total energy $E(f)$. Given the estimated velocity field computed using equation (1), the initial displacement of each grid is computed and used to initialize the following algorithm for elastic matching:
(1) For each grid g determine its initial matching grid g' as predicted by the Kalman filter.
(2) Compute the initial energy $E(g)_{g \mapsto g'}$ for each pair of matching grids (g, g').
(3) For each grid g and its matching grid g', compute $\Delta E(f)_{g \to g''}$ for all $g'' \in O(g')$. If there exists a $g'' \in O(g')$ such that $\Delta E(f)_{g \to g''} < 0$ and $\Delta E(f)_{g \to g''}$ is the minimum amongst all $g'' \in O(g')$, then replace g' with g'' as the mapping grid of g.
(4) Repeat step 3 until there is no change in the mapping f.

3.4 Velocity Field Approximation Using B-spline Surfaces

The elastic matching algorithm yields a mapping which minimizes an energy function that takes into account both, feature similarity and shape distortion during tracking. The computed mapping determines the displacement of each grid along the X and Y axes, i.e. the velocity of each grid. However, this mapping may also yield some noisy and false matches that do not reflect the actual motion of the object. Hence B-spline surfaces are used to smooth the velocity field and suppress the effect of the noisy and false matches.

Provided $X = \xi_x(i, j)$ is known, the following procedure is used to determine the $M \times N$ B-spline control points in order to approximate X. For each grid location (i, j) of the object, the corresponding B-spline control parameter is estimated as: $(\hat{u}, \hat{v}) = ((M - m + 1) * i/W, (N - n + 1) * j/H)$ where W and H are the width and height of the object respectively. Suppose the velocity field given by elastic matching is $\xi(i_k, j_k) = (\xi_x(i_k, j_k), \xi_y(i_k, j_k)), k = 1, ..., K$ where K is the total number of grids in $\xi(i_k, j_k)$. The set of corresponding control parameters is represented by $(\hat{u}_k, \hat{v}_k)$, The estimated velocity component along the X axis $\hat{\xi}_x$ is expressed in terms of $(\hat{u}_k, \hat{v}_k)$ as follows:

$$\hat{\xi}_x(\hat{u}_k, \hat{v}_k) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} d^x_{i'_k, j'_k} N_i^m(u_k^1) N_j^n(v_k^1) \quad (6)$$

where the $d_{i,j}$'s are the control points which determine the association of a given point on the B-spline surface with the control parameters (u, v), $N_i^m(u)$ and $N_j^n(v)$ are the basis functions along the u and v axes respectively, and n and m are the orders of the B-spline ($m = n = 4$ in our case). $i'_k = i + u_k^0$, $j'_k = j + v_k^0$, $(u_k^0, v_k^0) = (\lfloor \hat{u}_k \rfloor, \lfloor \hat{v}_k \rfloor)$, and $(u_k^1, v_k^1) = (\hat{u}_k - u_k^0, \hat{v}_k - v_k^0)$. Equation (6) can be further generalized as $\hat{\xi}_x(\hat{u}_k, \hat{v}_k) = \sum_{i=0}^{M} \sum_{j=0}^{N} d^x_{i,j} B_k(i, j)$, where $B_k(i, j) = N^{m-1}_{i - \hat{u}_k^0}(\hat{u}_k^1) N^{n-1}_{j - \hat{u}_k^0}(\hat{v}_k^1)$, if $\lfloor \hat{u}_k \rfloor \leq i <$

$\lfloor \hat{u}_k \rfloor + m - 1$ and $\lfloor \hat{v}_k \rfloor \leq j < \lfloor \hat{v}_k \rfloor + n - 1$; $B_k(i,j) = 0$ otherwise. For each grid associated with an object, minimization of the following objective function is used to determine the values of the $L = M \times N$ control points $E = \sum_{k=1}^{K} \|\xi_x(i_k, j_k) - \hat{\xi}_x(\hat{u}_k, \hat{v}_k)\|^2$, where $(\hat{u}, \hat{v})$ are the estimated control parameters. The minimization entails solving a system of equations given by $\partial E / \partial d^x_{i,j} = 0$, which, in turn, can be represented by $Ad^x = C$, where A is an $L \times L$ matrix given by

$$\begin{pmatrix} \sum_{k=1}^{K} B_k(0,0)B_k(0,0) & \dots & \sum_{k=1}^{K} B_k(0,0)B_k(M,N) \\ \sum_{k=1}^{K} B_k(0,1)B_k(0,0) & \dots & \sum_{k=1}^{K} B_k(0,1)B_k(M,N) \\ \dots & \dots & \dots \\ \sum_{k=1}^{K} B_k(M,N)B_k(0,0) & \dots & \sum_{k=1}^{K} B_k(M,N)B_k(M,N) \end{pmatrix}$$

C is a vector given by $(\sum_{k=1}^{K} B_k(0,0)x(i_k, j_k) \ \dots \ \sum_{i=1}^{K} B_k(M,N)x(i_k, j_k))^T$. The value of d^x is obtained by solving the system of equations $Ad^x = C$, which can be solved using LU decomposition. Note that the matrix A is symmetric and is the same for both the velocity component fields X and Y. It is necessary to point out that in equation (2), $Z = ((d^x)^T, (d^y)^T)^T$. The value of M is usually small for rigid object tracking. In our experiments, 4×4 control points can approximate the velocity field in the image plane resulting from 3-D movement of a planar object (translation, rotation or a combination of the two) with negligibly small mean squared error (MSE). Although we have not examined the MSE resulting from the approximation, using 4×4 control points, of the velocity field in the image plane resulting from 3-D movement of a 3-D object, our experiments on human tracking yield very good results. More accurate approximation can be achieved by using a smaller grid size and more control points. Since an elastic object with restricted movement can be usually approximated by a rigid object with articulated motion, the resulting velocity field can still be approximated using 4×4 control points. More control points are necessary only for modeling complex and/or abrupt motion and deformation of a highly elastic object. To further reduce the computational complexity, the values of N_i^m and N_i^n can be precomputed and stored in a lookup table.

4 Object Parameter Updating and Occlusion Reasoning

After computation of the B-spline surface fit to the 2-D velocity field, the grid mapping is recomputed based on the B-spline surface representation. Suppose there are n objects at time $t-1$, where object i is associated with a corresponding grid set G_i^{t-1}, $i = 1, ..., n$. By mapping G_i^{t-1} to grid image F_G at time t, the corresponding grid set G_i^t can be recomputed such that for each grid in G_i^t, there is a corresponding grid in G_i^{t-1}.

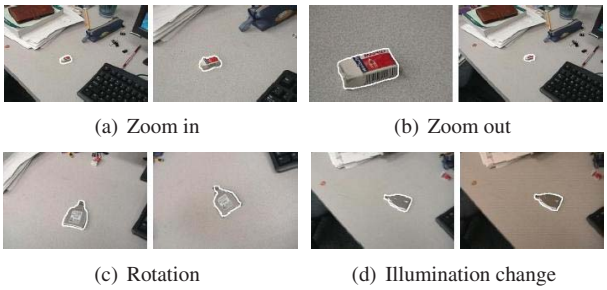
In a multiple object tracking scenario, occlusion reasoning is invoked if several grids from different objects correspond to a single grid g_i^t. The occlusion analysis is based on the estimation of the conditional probabilities that the grid corresponds to each of the several objects. The grid is deemed to correspond to the object for which the conditional probability is maximized.

Suppose grid g^t at time t corresponds to m grids $g_{k_i}^{t-1}$ at time $t-1$, where $0 \leq i < m$ and $0 \leq k_i < n$. Let $g_{k_i}^{t-1}$ be a grid associated with object k_i, that is $g_{k_i}^{t-1} \in G_{k_i}^{t-1}$. We assume that the color feature vector for each object k_i at time t exhibits a normal distribution $\mathcal{N}(c_{k_i}^{c,t-1}, \sigma_{k_i}^{t-1})$ with mean $c_{k_i}^{c,t-1}$ and standard deviation $\sigma_{k_i}^{t-1}$. The conditional probability that g^t is the corresponding grid of object k_i at time t is given by $p(g^t | G_{k_i}^{t-1}) = \mathcal{N}(c^{c,t} | c_{k_i}^{c,t-1}, \sigma_{k_i}^{t-1})$. The grid g^t is labeled as k_i if the conditional probability $p(g^t | G_{k_i}^{t-1})$ is a maximum over all possible objects. However, this could result in incorrect labeling due to noise and inherent inaccuracies in the object model. To minimize such labeling errors, we exploit spatial coherence. For every grid labeled using the above procedure, we check its eight neighbors; if five or more of the eight neighboring grids have the same label k_i, we change the label of this grid to k_i. The occlusion parameter $o_{g_{k_i}}$ is set to 1 and $o_{g_{k_j}}$ is set to 0, where $0 \leq j < m$ and $j \neq i$.

After the grids within the confusion area are classified, the parameters of each non-occluded grid are updated. If the color feature of grid g is $c^{c,t-1}$ with standard deviation σ^{t-1} at time $t-1$, and the observed color feature at time t is $c^{c',t}$ then the grid parameters are updated as $c^{c,t} = c^{c,t-1} + \rho(c^{c',t} - c^{c,t-1})$ and $(\sigma^t)^2 = (\sigma^{t-1})^2 + \rho[(c^{c',t}, -c^{c,t-1})(c^{c',t} - c^{c,t})^T - (\sigma^{t-1})^2]$, where $\rho = \alpha \mathcal{N}(c^{c',t} | c^{c,t-1}, \sigma^{t-1})$

5 Experimental Results

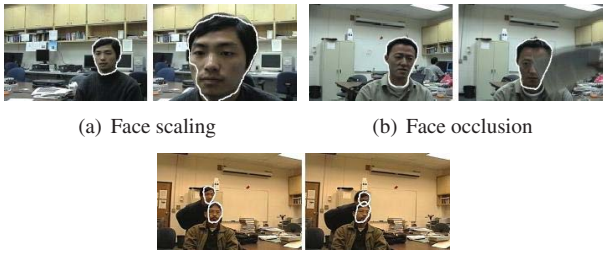
The proposed tracking algorithm has been applied to various tracking scenarios. Figure 2(a) shows the snapshots of the tracking of an eraser in a video while the camera is zooming in. Figure 2(b) shows the snapshots of the tracking of an eraser in a video while the camera is zooming out. Figure 2(c) shows the snapshots while the tracked object is rotating in the image plane and Figure 2(d) shows the snapshots while the scene is subject to global change in ambient illumination. Figure 2 shows that the proposed tracking algorithm can handle large changes in object size (i.e., significant scale changes) and handle object rotation and changes in scene illumination due to the adaptive nature of the object template and the robustness of the features used. Experiments are also conducted on face tracking. Figure 3(a) shows the tracking result when the tracked face exhibits large scale changes in the video.



(a) Zoom in (b) Zoom out

(c) Rotation (d) Illumination change

Figure 2: Zooming, rotation, illumination change



(a) Face scaling (b) Face occlusion

(c) Two face tracking

Figure 3: Face Tracking

Figure 3(b) shows the tracking result in the presence of occlusion thus demonstrating that the occlusion reasoning is very robust in handling occlusions. Figure 3(c) shows the tracking results on two faces where one face occludes another. The various tracking results are available at http://www.cs.uga.edu/∼xingzhi/research/elastic/index.html.

6 Conclusions

A novel multiple object tracking scheme based on Kalman filtering and elastic matching is proposed in this paper. The proposed scheme provides a general framework for tracking of both rigid and elastic objects. It can be viewed as a generalized Kalman filter where elastic matching is used to measure the velocity field which is then approximated (and smoothed) using B-spline surfaces. Control points of the B-spline surfaces are used directly as tracking variables in the Kalman filter. The limitation of the Gaussian distribution assumption in the Kalman filter is overcome by the large capture range of elastic matching which can correct for the prediction errors made by the Kalman filter. The B-spline surfaces are used to update the grid-based object color features. The adaptation of the object color features are subsequently used in occlusion reasoning. Since the proposed tracking method does not rely on background subtraction, it is suitable for object tracking in dynamic scenes captured using both a static camera and a moving camera.

References

[1] P. Li, T. Zhang, A. E.C.Pece, Visual contour tracking based on particle filters, *Img. Vis. Comput.*, Vol. 21, 2003, pp. 111-123.

[2] M. Isard and J. MacCormick, BraMBLe: A Bayesian Multiple-Blob Tracker, *Proc. ICCV*, Vancouver, Canada, Vol. 2, July 2001, pp. 34-41.

[3] S.J. McKenna, S. Jabri, and Z. Duric, A. Rosenfeld, H. Wechsler, Tracking Groups of People, *CVIU*, Vol. 80, 2000, pp. 42-56.

[4] K. Nummiaro, E. Koller-Meier, and L.V. Gool, An adaptive color-based particle filter, *Img. Vis. Comput.*, Vol. 21, No. 1, 2003, pp. 99-110.

[5] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C.V.D. Malburg, and R. Wurtz, Distortion Invariant Object Recognition in the Dynamic Link Architecture, *IEEE Trans. Comp.*, Vol. 42, No. 3, 1992, pp. 300-311.

[6] M. Pietikainen, T. Maenpaa, and J. Virtola, Color Texture Classification with Color Histogram and Local Binary Patterns, *Proc. 2nd Intl. Wkshp. Texture Anal. Synth.*, Copenhagen, UK, June 2002, pp. 109-112.

[7] D. Koller, J.W. Weber and J. Malik, Robust Multiple Car Tracking with Occlusion Reasoning. *Proc. ECCV* Stockholm, Sweden, 1994, pp. 189-196.

[8] D. Terzopoulos and R. Szeliski, Tracking with Kalman Snakes, in *Active Vision*, MIT Press, Cambridge, MA, USA, 1993, pp. 3-20.

[9] W.J.Rucklidge, Locating Objects Using the Hausdorff Distance, *Proc. ICCV*, Massachusetts USA. June, 1995, pp. 457-464.

[10] C. Harris, and M. Stephens, A Combined Corner And Edge Detector. *Proc. The Fourth Alvey Vision Conference*, Manchester, UK, 1988, pp. 147-151.

[11] S. Malik, G. Roth, and C. McDonald, Robust Corner Tracking for Real-Time Augmented Reality, *Proc. Vision Interface*, Calgary, Alberta, Cananda, May, 2002, pp. 399-406.

[12] V. Lepetit, J. Pilet, and P. Fua, Point Matching as a Classification Problem for Fast and Robust Object Pose Estimation. *Proc. IEEE Conf. CVPR*, Washington DC, USA, June 2004. Vol. 2, pp. 244-250.