
ELSEVIER

Pattern Recognition Letters 20 (1999) 879–887

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

Motion tracking of human mouth by generalized deformable models

Syin Chan [a,*], Chong Wah Ngo [a], Kok F. Lai [b]

[a] *Nanyang Technological University, School of Applied Science, N4-02a-32, Nanyang Avenue, Singapore 63978, Singapore*
[b] *Kent Ridge Digital Labs, Singapore 119613, Singapore*

Received 27 May 1998; received in revised form 28 May 1999

Abstract

We propose and evaluate four trackers for tracking the shape, motion and deformation of a human mouth in video sequences. The trackers are suitable for use in very low bitrate video coding systems. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Motion tracking; Deformable models; Model-based coding

1. Introduction

Recent developments in multimedia applications have emphasized the demand for image sequence tracking, analysis and coding. Scene tracking and analysis are particularly useful in model-based facial image coding systems which promise very low bitrate communication (e.g. Samal and Iyenger, 1992; Aizawa and Huang, 1995). Such systems are suitable for applications like videophone and videoconferencing. These applications can be constructed by preparing a 3D human face model at both the transmitting and receiving ends of a visual communication system. Input images at the transmitter are analysed in terms of motion and deformation and the necessary analysis parameters are then transmitted. At the receiver, these analysis parameters are used together with the 3D model to synthesize the output images. Transmission of the analysis parameters requires a much smaller bandwidth than transmission of the actual image data. Therefore, we have proposed and implemented four trackers which can encode the shape, deformation and motion of the tracked feature through refinement, synthesis and match of deformable models. We evaluate the trackers by applying them to the tracking of a human mouth in video sequences.

The proposed trackers are based on the generalized active contour model or g-snake (Lai and Chin, 1995). G-snake is capable of describing any arbitrary contour while retaining its global and local semantics. We denote two energy functionals, internal and external, to represent the goodness of fit between the contour model and the desired image feature. The internal energy is based on a shape matrix which represents the shape of the desired feature. The external energy models local deformation and attracts the contour to salient image features. By minimizing the combined

* Tel.: +65-790-5748; fax: +65-792-6559; e-mail: asschan@ntu.edu.sg

0167-8655/99/$ - see front matter
PII: S0167-8655(99)00054-9

energies, the contour model will be drawn towards the contour of the desired feature. In all four trackers that we have investigated, the initial contour of the tracked feature is obtained from the first image frame using Gaussian pyramid images, edge maps, generalized Hough Transform and energy minimisation (Ngo et al., 1995). For subsequent frames, the four trackers will use different methods to obtain the new contour. In the first tracker, the preceeding contour is simply overlaid on the new image and the deformation process is restarted to obtain the new contour. The second tracker imposes affine motion smoothness constraints to exploit temporal motion redundancy existing in the image sequence. The third tracker applies principal component analysis to synthesize a codebook of contour templates. The last tracker combines the above ideas and synthesizes templates along several major modes of motion and tracks the object by selecting the best matched template. Since these trackers, with the exception of the first tracker, require only a few parameters to describe the shape and motion changes of image features, they are suitable for very low bitrate image coding.

Several motion trackers have evolved from the framework of deformable models (Kass et al., 1987). These trackers include Kalman snake (Terzopoulos and Szeliski, 1992), deformable templates (Yuille et al., 1992) and dynamic contours (Blake et al., 1993; Blake and Isard, 1994). The proposed g-snake trackers are built upon the probabilistic framework and consider some special features such as affine invariant shape template in dynamic contours. They differ from the other snake models in the following aspects:

- A contour is represented by a set of points. Based on 2nd order Markov model, each point is expressed as a linear combination of its two neighboring points. With this characteristic, an affine invariant shape matrix can be derived to tackle both local and global deformations, and ensure spatial smoothness. The shape representation is more general compared to deformable templates and is able to model any arbitrary contour (Lai and Chin, 1995).
- Unlike physical model, we do not consider mass and damping coefficients which require expert knowledge on the target object. Instead, we apply an adaptive prediction algorithm to exploit temporal motion smoothness, and analyse the translation and deformation process of an object.
- We use principal component analysis to determine the statistics of the contour points over a collection of training examples. In this way, we obtain several main modes of object deformation. We use these modes to ensure temporal smoothness during tracking. This approach is different from deformable templates where every mode of variation must be defined explicitly by a human expert.

We have also considered some of the techniques used in face recognition (Pentland et al., 1994). For example, using principal component analysis, any face can be generated from a weighted sum of eigenface. This allows a very efficient coding of deformable objects, as the most probable appearance of a tracked feature can be synthesized readily.

By combining the above ideas, we propose a synthesis and matched paradigm (Lai et al., 1996) based on the g-snake. Through a motion learning algorithm, the tracker incorporates the shape as well as motion constrained in the contour model. It then synthesizes a few possible match templates along several major modes of motion and tracks the feature by selecting the best matched template. Template matching can be done in parallel so as to make real-time video tracking possible.

The rest of the paper is organized as follows. Section 2 describes the setup and formulation of the four trackers. Section 3 compares the performance of these trackers in terms of speed efficiency, tracking accuracy and data compression. Section 4 discusses the results and Section 5 concludes the paper.

2. Tracking by synthesis and match

We define a contour as the vector containing an ordered set of points, $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n]$. Each $\boldsymbol{v}_i$ is defined on the finite grid: $\boldsymbol{v} \in \mathbb{E} = \{(x, y) : x, y = 1, 2, \ldots, M\}$, thus $\boldsymbol{V} \in \mathbb{E}^n$. We also denote $\boldsymbol{U} \in \mathbb{E}^n$, where each $\boldsymbol{u}_i = \boldsymbol{v}_i - \boldsymbol{g}$ represents the displacement from an arbitrary reference point $\boldsymbol{g}$.

In the subsequent experiments, we use eight points to represent a mouth contour. Two points are placed at the two corners of the mouth, and three points each are spaced along the boundary of the upper lip and the lower lip.

Given a time series of contours $\boldsymbol{V}(1), \boldsymbol{V}(2), \ldots, \boldsymbol{V}(t-1)$, we wish to synthesize $\hat{\boldsymbol{V}}(t)$ such that it is as close to the actual contour $\boldsymbol{V}(t)$ at time t as possible. We present four approaches whereby this can be accomplished. Trackers 2 and 4 incorporate motion smoothness constraint whereas trackers 1 and 3 do not.

2.1. Tracker: refinement and tracking

G-snake models and extracts deformable contours by integrating both global and local deformations in a regenerative shape matrix. An internal energy function can incorporate a global model, while an external energy function can deform the resulting contour to match with the underlying image features. To accurately track a mouth sequence, we train the shape matrix and deformation variance from various mouth samples. The learned contour is eventually applied to a lip tracking process. Once a g-snake locks on to a global minimum on the current frame, it restarts refinement from the current position to track the desired features of the next frame.

In general, tracker 1 is an unconstrained g-snake, where the resulting contours can be distracted over time, as illustrated in Fig. 1. In this case, we found that the boundary of the lower lip is unclear and the teeth generate stronger edges compared to it. As a result, the stronger edges cause the resulting contours of tracker 1 to shrink.

2.2. Tracker: affine motion smoothness constraints

To overcome the deficiency of the unconstrained snake in tracker 1, tracker 2 exploits the temporal motion redundancy existing in the image sequence to predict the motion of the contours. Let $\hat{\boldsymbol{U}}(t)$ and $\hat{\boldsymbol{d}}(t)$ represent the predicted contour model and predicted displacement at time index t, respectively. Applying first order smoothness constraint on global deformation and second order smoothness constraint on displacement, we can predict the evolution of a contour as follows:

$$\hat{\boldsymbol{U}}^{\mathrm{T}}(t) = \boldsymbol{T}(t)\boldsymbol{U}^{\mathrm{T}}(t-1), \tag{1}$$

$$\hat{\boldsymbol{d}}(t) = \omega_1(t)\boldsymbol{d}(t-1) + \omega_2(t)\boldsymbol{d}(t-2), \tag{2}$$

where $\boldsymbol{T}(t)$ is a 2×2 linear transformation matrix modeling scale change, rotation and dilation; $\boldsymbol{d}(t) = \boldsymbol{g}(t) - \boldsymbol{g}(t-1)$ is the displacement; and $\omega_1(t)$ and $\omega_2(t)$ are the weights for the displacement prediction.

We used the least-square method to obtain $\boldsymbol{T}(t)$, $\omega_1(t)$ and $\omega_2(t)$:

$$\boldsymbol{T}(t) = \boldsymbol{U}^{\mathrm{T}}(t-1)\boldsymbol{U}(t-2) \times [\boldsymbol{U}^{\mathrm{T}}(t-2)\boldsymbol{U}(t-2)]^{-1}, \tag{3}$$

$$\begin{bmatrix} \omega_1(t) \\ \omega_2(t) \end{bmatrix} = [\boldsymbol{d}(t-1) \quad \boldsymbol{d}(t-2)]^{-1}\boldsymbol{d}(t). \tag{4}$$

$\boldsymbol{T}(t)$ is initialized to the identity matrix for $t < 2$ and $\omega_1(t) = \omega_2(t) = 0$ for $t < 3$.

The performance of this tracker can be seen in Fig. 2. Unfortunately, the tracking performance deteriorates when there are sudden changes in motion or contour shape. Increasing the prediction to that beyond second order did not yield significant improvement, while resulting in more expensive computational cost. This leads to the development of tracker 3.

2.3. Tracker: contour codebook

Berker (1972) deduced that there are only about 13 visually distinct mouth shapes associated with

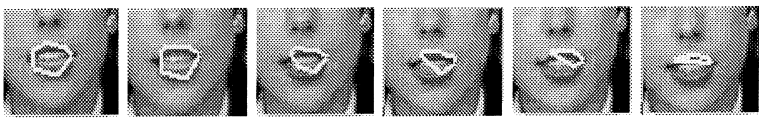

Fig. 1. Tracker 1: the resulting contours tend to be distracted.

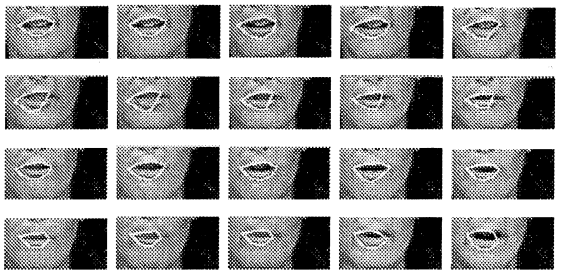

Fig. 2. Tracking the lip area of Miss America sequence by g-snake prediction algorithm.

vowel and consonant phonemes. Therefore, this tracker constructs a codebook of templates from contour samples and later synthesizes the contours by selecting suitable templates from this codebook.

Firstly, we collect a set of training contours $\boldsymbol{V}(1), \boldsymbol{V}(2), \ldots, \boldsymbol{V}(m)$ from the target object. Next, we compute the average contour $\bar{\boldsymbol{U}}$ and the correlation matrix $\boldsymbol{R}_u$:

$$\boldsymbol{R}_u = \frac{1}{m}\sum_{i}^{m}(\boldsymbol{U}(i) - \bar{\boldsymbol{U}})(\boldsymbol{U}(i) - \bar{\boldsymbol{U}})^{\mathrm{T}}. \tag{5}$$

We obtain the main mode of deformation from $\boldsymbol{R}_u$ via principal component analysis. Diagonalizing $\boldsymbol{R}_u$, we have

$$\boldsymbol{R}_u = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^{\mathrm{T}}, \tag{6}$$

where $\boldsymbol{\Phi} = [\phi_1, \phi_2, \ldots, \phi_n]$ consists of ϕ_i as its eigenvectors, and $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ consists of the corresponding eigenvalues in its diagonal. The eigenvectors, which are uncorrelated to one another, are arranged such that ϕ_1 represents the most significant mode of deformation, ϕ_2 represents the next significant mode and so on. Typically, we retain only the first k eigenvectors if

$$\sum_{j=1}^{k}\lambda_j/\mathrm{tr}(\boldsymbol{R}_u) \geqslant p, \tag{7}$$

where $p \in [0, 1]$ is a threshold, and $\mathrm{tr}(\boldsymbol{R}_u)$ is the trace of the correlation matrix. For example, if $p = 0.7$, then the first k eigenvectors account for more than 70% of the variance in $\boldsymbol{R}_u$.

We can synthesize the templates in the codebook via the equation

$$\boldsymbol{U}(b_1, b_2, \ldots, b_k) = \bar{\boldsymbol{U}} + \sum_{j=1}^{k} b_j\phi_j \tag{8}$$

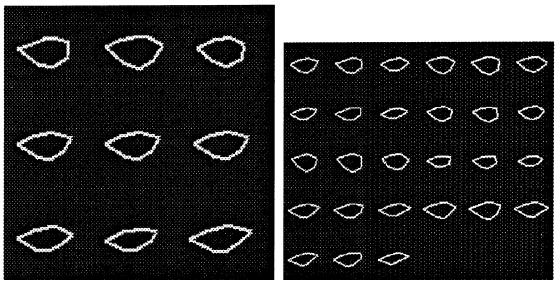

Fig. 3. Codebook of human mouth. (LHS) individual mode: variance = 24.9% (left), 18.4% (center), 14.0% (right); (RHS) combined mode.

by varying the weights b_j within the range given by

$$b_j \in \left[-a\sqrt{\lambda_j}, a\sqrt{\lambda_j}\right], \tag{9}$$

where $a = 2.56$ corresponds to the 99% confidence level in Gaussian distribution.

Fig. 3 shows the synthesized templates of a human mouth obtained from 150 training contours using $p = 0.5$. The vertical columns in Fig. 3 (LHS) show the synthesis along the three most significant deformation modes. Each horizontal row shows the effect of varying their weights. We then combine these modes to synthesize 27 different templates in the codebook, as shown in Fig. 3 (RHS).

Using the above templates, we track the lip motion in the Miss America sequence. Fig. 4 shows the tracking results for the 80–119th frames. The sequence runs from the top left to the bottom right. It can be seen that the image is matched correctly for most of the frames. However, as the head starts tilting and the mouth changes shape from 110 to 119th frame, none of the template match the actual mouth shape accurately. This can be improved by increasing the number of mouth templates, by moving the template in a larger area surrounding the previous local minimum, and/or by refining the initial estimates via g-snake. However, these solutions will result in longer computational time.

The codebook method is efficient as it allows the encoding of within-class variation, using only the most probable templates of a target object. However, this tracker treats each image frame independently. In the next tracker, we adopt a

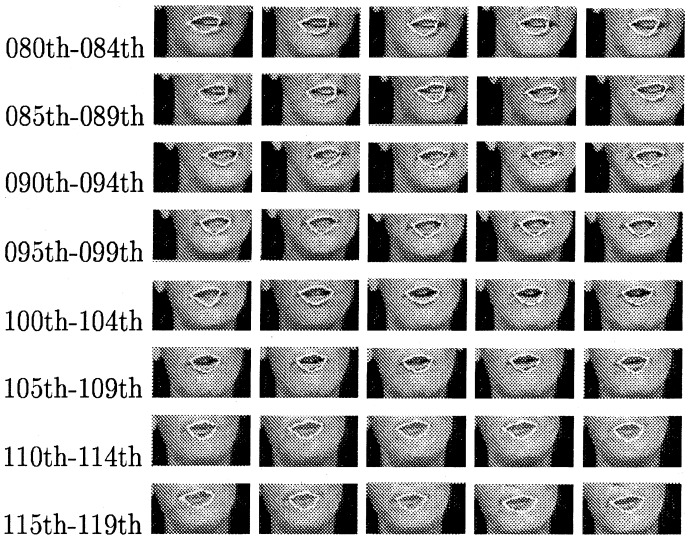

Fig. 4. Motion tracking of Miss America sequence by codebook method.

synthesis approach that exploits both the specific information about the target object, as well as the temporal redundancy existing in the image sequence.

2.4. Tracker: modes of motion

This method is motivated by the fact that a real object typically moves only in several prescribed manners. We thus reformulate the problem as such: given the contours $\boldsymbol{V}(1), \boldsymbol{V}(2), \ldots, \boldsymbol{V}(m)$, find the major modes of motion so that every $\hat{\boldsymbol{V}}(t+1)$ may be predicted from $\boldsymbol{V}(t)$. Now, define $\Delta(i) = \boldsymbol{V}(i) - \boldsymbol{V}(i-1)$, we compute the mean $\bar{\Delta}$ and the correlation matrix R_Δ:

$$\boldsymbol{R}_\Delta = \frac{1}{m}\sum_{i}^{m}(\Delta_i - \bar{\Delta})(\Delta_i - \bar{\Delta})^{\mathrm{T}}. \tag{10}$$

As in tracker 3, we perform principal component analysis to obtain

$$\boldsymbol{R}_\Delta = \Theta \Lambda \Theta^{\mathrm{T}}. \tag{11}$$

$\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$ contains the eigenvectors describing the major modes of motion. Assuming that the motion process is stationary, we can now generate templates for the next possible match using only k eigenvectors:

$$\boldsymbol{U}(t+1, b_1, b_2, \ldots, b_k) = \boldsymbol{U}(t) + \bar{\Delta} + \sum_{j=1}^{k} b_j \theta_j. \tag{12}$$

We then select the best matched template synthesized in every frame to track the lip motion in the Miss America sequence. Fig. 5 shows the results. It can be seen that the templates give adequate representation of the actual mouth, even though head tilting is introduced at some stages. This time the movements of the mouth from 110th to 119th frame are tracked correctly.

3. Results

The performance of the four trackers in speed efficiency, tracking accuracy and data compression is evaluated. The evaluation is based on a well-known sequence, Miss Claire, which consists of 168 frames. The tracked feature is a human mouth on an image of 360×288 pixels. The evaluation is conducted on a SUN SPARC 5 workstation with 32 MB primary memory.

3.1. Speed efficiency

We compare the performance in speed efficiency of the four trackers in Table 1. The total amount

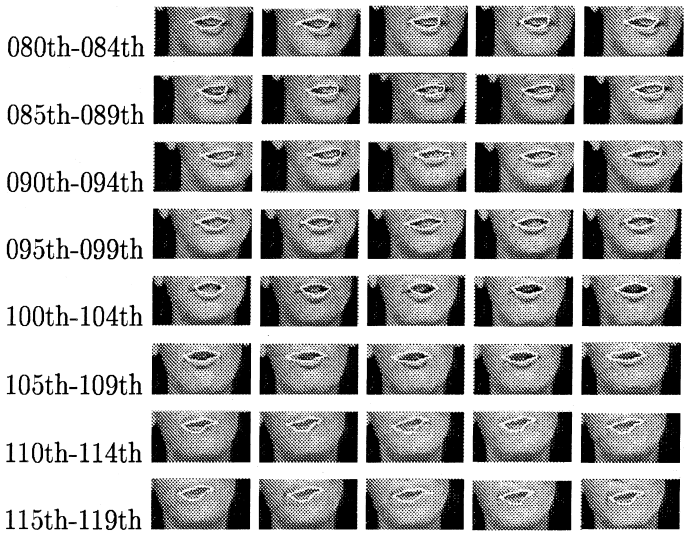

Fig. 5. Motion tracking of Miss America sequence by modes of motion.

Table 1
Speed efficiency of the four trackers (Energy threshold = 0.5)

	Number of reset frames	Frame/s
Tracker 1	14	0.861
Tracker 2	22	0.811
Tracker 3	32	0.700
Tracker 4	19	0.976

of time required to read the image file, track the desired feature and display the result on a screen over the 168 image frames is measured. From this measure, the number of frames tracked per second is calculated.

During tracking, the g-snake's internal and external energy is checked against a threshold measure. Once the resulting energy value exceeds the threshold measure, the previous local minimum contour will be discarded. Generalized Hough transform (GHT) will then take place to re-localize the initially trained contour and this generates a reset frame.

Table 1 shows that tracker 4 has the fastest tracking speed (frame/s). This is because tracker 4 has fewer reset frames (compared to trackers 2 and 3) and it does not perform energy minimization procedure (compared to trackers 1 and 2). It only performs contour matching on the vicinity of the best fit position of the preceding frame almost throughout the entire sequence.

3.2. Tracking accuracy

We evaluate the accuracy of the trackers by computing the mean-square error (MSE) between the actual and tracked contours:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i}^{n} ||\boldsymbol{v}_i - \hat{\boldsymbol{v}}||^2. \tag{13}$$

The actual contours are selected manually from the image frames. The results are shown in Fig. 6.

As expected, tracker 4 has the lowest MSE. The average MSE of trackers 1, 2, 3 and 4 are 28.4, 40.4, 37.3 and 8.0, respectively. At around the 67th to 72th frame, all four trackers have relatively large MSE values. These frames typically involve 3D head rotation and mouth shearing.

3.3. Data compression

Data compression can be achieved by storing or transmitting the motion parameters instead of the contour location information. In this experiment, we have 91 056 bytes of contour information that

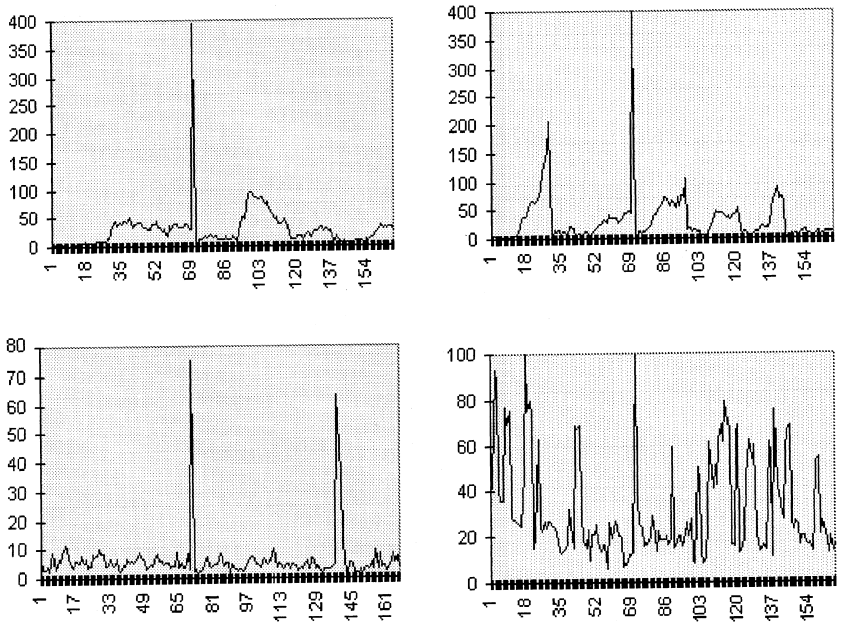

Fig. 6. Mean-square error analysis of trackers 1, 2, 3 and 4 (in clockwise direction starting from top left. The *X*-axis represents the frame number while the *Y*-axis shows the mean-square error).

are manually extracted. The contour information that includes location, shape and deformation variance parameters, are stored in 168 contour files with 542 bytes each. The amount of data compression achieved by each tracker is shown in Table 2.

Tracker 1 does not encode any motion parameter. Tracker 2 encodes the contour motions by maintaining an initial contour file and an ASCII file that describes the transformation matrices and displacement vectors. Trackers 3 and 4 express motions in term of eigenvectors and eigenvalues. In addition to an average contour file, they maintain a codebook of templates and modes of motion, respectively. The results show that trackers 3 and 4 are able to achieve high compression ratios.

Table 2
Results of compressing 91 056 bytes of contour information

	Compressed data (bytes)	Compression ratio
Tracker 1	91 056	1
Tracker 2	22 478	4
Tracker 3	2 056	44
Tracker 4	2 303	40

4. Discussions

The performance of the trackers can be affected by the following factors:

- *Fidelity measure.* There is no standard method for handling the error accumulation problems in model-based motion tracking. One major disadvantage of employing the energy threshold as a fidelity measure is that a distracted contour can still obtain a low energy as long as it locks on to a strong edge. This can be seen in Fig. 1. The choice of a suitable measure to distinguish poor tracking from good tracking is thus important.
- *Training overhead.* To overcome the above problem, trackers 3 and 4 incorporate prior knowledge to restrict the freedom of a g-snake. However, intensive training is required in order

to obtain the appropriate eigenvectors and eigenvalues. In general, this method is good when the tracked feature is approximately known.

- *Codebook size.* In trackers 3 and 4, if the codebook size is small, there may be an abrupt change to a different mouth shape between two consecutive frames. This can be improved by increasing the number of templates (we use nine templates in our experiments). However, this increases the codebook size and is likely to degrade both the speed efficiency and the data compression ratio.
- *Object motion.* All trackers will generally fail when there is significant object motion between consecutive frames. However, tracker 2 can predict the next contour correctly if the object follows the motion smoothness constraints. Otherwise, the actual contour will be very different from the predicted contour and the energy miminization algorithm will not be able to correct it.

For the third and fourth trackers, if sufficient computational power is available, the best matched template $\hat{\boldsymbol{V}}(t)$ can be further refined, or deformed, to obtain contours that better describe the image features. Multiple templates can also be matched to the images using a suitable parallel architecture.

For model-based coding of head-and-shoulder type of video sequences, it is necessary to consider the other facial features apart from the mouth. The motion and deformations of these features are generally highly correlated. In (Ngo et al., 1995), we presented encouraging experimental results on tracking the human face, eye and mouth contours using the first tracker described in this paper.

5. Conclusions

Four motion trackers that are based on the g-snake have been presented. We evaluate the performance of these trackers using an established motion sequence, and discuss their respective merits and demerits. The fourth tracker, which tracks the object by synthesizing the templates along several major modes of motion and selecting the best matched template, is found to be superior in speed efficiency, tracking accuracy and compression ratio. It requires only a few parameters to characterize the motion and is therefore suitable for low bit rate visual communication tasks, such as in model-based image coding. In these applications the templates can be synthesized in advance and stored in both transmitter and receiver. For every frame, the transmitter performs motion tracking and sends the code of the best matched template to the receiver. The receiver then uses an appropriate object model, e.g. Parke's wireframe model together with texture mapping to synthesize the image. An objective comparison of the tracking accuracy with other trackers can then be carried out by examining the original image and the synthesized image.

References

Aizawa, K., Huang, T.S., 1995. Model-based image coding: advanced video coding techniques for very low bitrate applications. Proc. IEEE 83 (2), 259–271.

Berker, K.W., 1972. Speechreading: Principles and Methods. National Education Press, Baltimore.

Blake, A., Isard, M., 1994. 3D position attitude and shape input using video tracking of hands and lips. Computer Graphics Proceedings, Annual Conference Series, pp. 185–192.

Blake, A., Curwen, R., Zisserman, A., 1993. A framework for spatio-temporal control in the tracking of visual contours. Internat. J. Comput. Vision 11 (2), 127–145.

Kass, M., Witkin, A., Terzopoulos, D., 1987. Snakes: active contour models. In: Proc.1st Internat. Conf. on Computer Vision, pp. 259–269.

Lai, K.F., Chin, R.T., 1995. Deformable contours: modeling and extraction. IEEE Trans. Pattern Anal. Machine Intell. 17, 1084–1090.

Lai, K.F., Ngo, C.W., Chan, S., 1996. Tracking of deformable contours by synthesis and match. International Conference on Pattern Recognition, pp. 657–661.

Ngo, C.W., Chan, S., Lai, K.F., 1995. Application of generalized active contour model for model-based image coding. In: Chua, T.S., Pung, H.K., Kunii, T.L. (Eds.), Multimedia Modeling Towards Information Superhighway, World Scientific, Singapore.

Pentland, A., Moghaddam, B., Starner, T., 1994. View-based and modular eigenspaces for face recognition. In: International Conference on Computer Vision and Pattern Recognition, pp. 84–91.

Samal, A., Iyengar, P.A., 1992. Automatic recognition and analysis of human faces and facial expressions: a survey. Pattern Recognition 25 (1), 65–77.

Terzopoulos, D., Szeliski, R., 1992. Tracking with Kalman Snakes, In: Blake, A., Yuille, A. (Eds.), Active Vison, MIT Press, Cambridge.

Yuille, A.L., Cohen, D.S., Hallinan, P.W., 1992. Feature extraction from faces using deformable templates. Internat. J. Comput. Vision 8 (2), 99–111.