Real-Time Imaging **2**, 67–79 (1996)

Tracking and Synthesizing Facial Motions with Dynamic Contours

Many researchers have studied techniques related to the analysis and synthesis of human heads under motion with face deformations. These techniques can be used for defining low-rate image compression algorithms (model-based image coding), cinema technologies, videophones, as well as for applications of virtual reality, etc. Such techniques need a real-time performance and a strong integration between the mechanisms of motion estimation and those of rendering and animation of the 3D synthetic head/face. In this paper, a complete and integrated system for tracking and synthesizing facial motions in real-time with low-cost architectures is presented. Facial deformations curves represented as spatiotemporal B-splines are used for tracking in order to model the main facial features. In addition, the system proposed is capable of adapting a generic 3D wire-frame model of a head/face to the face that must be tracked; therefore, the simulations of the face deformations are produced by using a realistic patterned face.

P. Nesi and R. Magnolfi

Department of Systems and Informatics, Faculty of Engineering
University of Florence, Via S. Marta 3, 50139 Florence, Italy
E-mail: nesi@ingfil.ing.unifi.it, www: *http://www-dsi.ing.unifi.it/~nesi*

Introduction

In recent years researchers have studied the techniques related to the analysis and synthesis of human heads/faces under motion and deformation. These techniques can be used for defining low bit-rate image compression algorithms [following the paradigm of model-based image coding e.g. [1]] for videophones, video-conferencing, as well as for applications of virtual reality, and cinema technologies, etc. In order to be effectively used, such techniques have to integrate mechanisms for motion estimation with those of 3D head/face modeling, rendering an animation (i.e., head/face synthesis). For most of these new applications, the processes of motion estimation and synthesis must be mandatorily performed in real-time.

The head/face motion estimation problem can be divided into two sub-problems – i.e., the estimation of head motions (global motions) and the estimation of facial deformations due to changes of expression (local motions) (Figure 1). The first problem is also known as *head tracking* and can be solved with traditional techniques for 3D motion estimation e.g., [2,3]. To this end, both matching and gradient-based techniques [4–6] could be used. In the literature, the second problem, i.e., the problem of estimation of facial deformations and motions (lips tracking, eyes tracking, etc.), has been addressed by using several techniques. These techniques can be classified in three main categories which can be distinguished on the basis of the mathematical elements they adopt for modeling facial features (i.e., mouth, eyes, eyebrows, nose, etc.) that have to be tracked: (i) deformable or dynamic contours

1077-2014/96/020067 + 13 $18.00

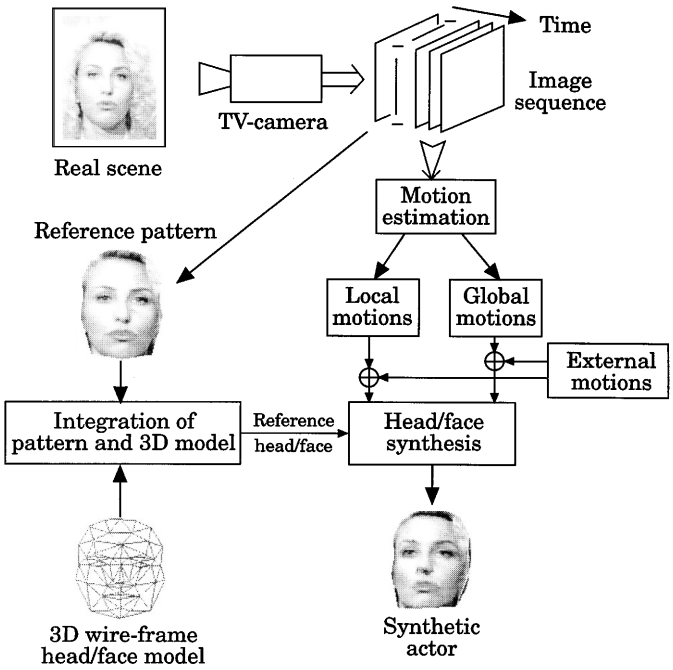

Figure 1. Process of analysis and synthesis of animated actors.

(splines/snakes, B-splines) [7–11]; (ii) deformable templates [12,13]; (iii) points or patterns (by using optical-flow or matching-based techniques) [14]. These features are tracked in subsequent image frames in order to estimate face deformations i.e., by measuring motions and deformations of the features selected.

It should be noted that most of the above mentioned techniques, which are based on deformable curves and/or templates, have been used to define methods for recognizing faces and objects in general [15–19].

As regards the head/face synthesis, this is obtained by using as a reference a synthetic model generated by (i) modeling the head/face 3D structure as a wire-frame object [20–22], and (ii) smearing the real face/head patter (i.e., texture) on the corresponding 3D wire-frame model instead of using classical algorithms for shading with uniform colors (e.g., Phong, Gouraud). Since the animated model must be as close to the real model as possible, corresponding points between the facial features (which are used for tracking the facial deformations) and the mathematical structures in the reconstructed synthetic model must be defined. The associations between these two domains are defined in a phase in which a parametrized 3D wire-frame model is adjusted in accordance with the real measures of the face under analysis. The process of adjustment in accordance with the real measures of the face under analysis. The process of adjustment can be simplified by deforming the wire-frame model in order to match a frontal image of the face shape [23]. In some cases, the structural model of the face be defined by also considering facial muscles for a certain depth [24]. Better results are obviously obtained by measuring the head/face structure directly from the real subject, but his can be a very difficult and thus unfeasible task in real applications.

Once a 3D synthetic model is obtained, this can be considered as a reference model, and the animation of the head/face is performed by applying the motions (i.e., global motions and local deformations) estimated from the image sequence to the reference model [25]. In this context, *animate* means translate, rotate, and deform the reference head/face model in the 3D synthetic space in order to present the corresponding projection on the screen. In the last phase, global and local motions can also be modified by considering external motions, for example by applying the deformations measured on the face observed from a different point of view (adding global motions) or for enhancing some expressions (adding deformations), etc.

In this paper, a complete system for tracking the main facial features (like mouth and eyebrows) and for reconstructing the deformations estimated on a synthetic

head/face in quasi-real-time by using low-cost architectures is presented. The method adopted for tracking facial features is based on dynamic contours, which in turn are mathematically modelled as spatiotemporal B-splines. The wire-frame model of the human head used has been obtained by improving the well-known CANDIDE model [21,26]. In the same section, it is shown how the 3D wire-frame model can be adapted to the face under analysis by means of a method derived by Reinders *et al.* which is based on the face pattern of a single frame [23]. Moreover, an *ad hoc* algorithm for guaranteeing as fast smearing of the real facial pattern on the synthetic model has also been defined.

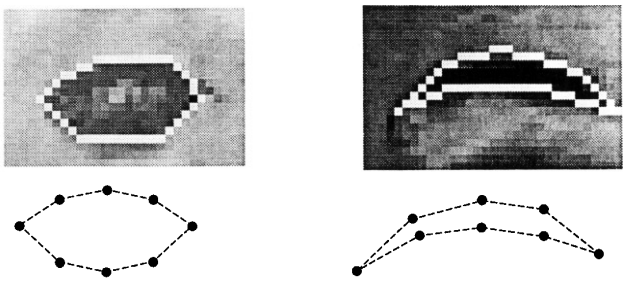

Figure 2. Modeling with STB-snakes: an open mouth (left) where the teeth are visible; and an eyebrow (right), with their possible respective representation.

Modeling Facial Features as Spatiotemporal B-Snakes

The mouth, the eyes, the eyebrows, the nose, etc. can be considered as the most important facial features. In order to reproduce their position and shape, it suffices to know the position of a draft shape which models these features (Figure 2). For this reason, the tracking of facial features has often been reduced to the problem of tracking curves which model the features shapes. Since the feature contour can change its form in subsequent frames, a method for contour tracking in time is needed. This process can be obtained by defining an energy model for deformable contours [7–10], or for dynamic contours [11,27], or for deformable templates [12,13]. This classification follows that proposed by Blake [28].

It should be noted that the approaches based on deformable contours (splines) and/or templates are usually computationally too heavy to be used for real-time tracking on low-cost architectures. Moreover, deformable contours are so flexible that in many cases it is very difficult to maintain their shape under control. In fact, in many applications of that technique the energy model also contains an energy factor which models the user's actions for manually deforming the splines in order to adjust their shape i.e., the so-called external energy. Moreover, deformable templates work well only when the shape of the feature under tracking is known and feature deformations are small and the feature shape structure does not change in time (e.g., these approaches present some limits for shapes which invert their curvature in time). On the contrary, dynamic contours are based on B-splines and attempt to integrate the above aspects since they model curves as a combination of elementary templates. In addition, they use a parametrized representation of the curve which makes their estimation cheaper with respect to classical splines and templates. This model for representing curves is defined as "B-snake" [27]. Moreover, since the model proposed extends the adoption of B-snakes to track curves in time by also considering energy factors expressing the changes in time, it can be viewed as a "spatiotemporal B-snake based model" (i.e., STB-snake).

An STB-snake is a deformable parametrized surface controlled by the temporal behavior of internal and image forces which act at each point of the surface in the spatiotemporal domain. The internal forces, F_{int}, represent the constraints on the shape curve (regularity, elasticity, etc.), while the image forces, F_{img}, guide the contour to match certain image features (luminance, contrast, etc.). By integrating these forces along the curve $v(s,t)$ the corresponding energies are obtained and from these the total energy:

$$E_{tot} = E_{int} - E_{img} = \int_T \int_s \left(F_{int} - F_{img}\right) \mathrm{d}s\mathrm{d}t, \qquad (1)$$

where $v(s,t)$ is the parametric description of the curve and $v(s,t) = v(x(s,t),y(s,t))$. The goal is to find the surface that minimizes the total energy in time. When a minimum for E_{tot} is reached, the expressions $x(s,t)$ and $y(s,t)$ define a curve which best fits the feature contour according to its definition in terms of E_{img}.

The *Internal Energy*, E_{int}, is defined as:

$$E_{int} = E_1 + E_2 + E_t, \qquad (2)$$

where E_1 and E_2 take into account the tension and the rigidity of the curve shape (the surface at a given time instant), respectively (i.e., they impose the regularity of the curve

shape). The corresponding forces are weighted with functions $\alpha(s)$ and $\beta(s)$ respectively:

$$E_1 = \int_T \int_s \left[\alpha(s) \mid v_s(s,t) \mid^2\right] \mathrm{d}s\mathrm{d}t, \quad (3)$$

$$E_2 = \int_T \int_s \left[\beta(s) \mid v_{ss}(s,t) \mid^2\right] \mathrm{d}s\mathrm{d}t, \quad (4)$$

E_t takes into consideration the temporal regularity of the surface in time:

$$E_t = \int_T \int_s \left[\tau(s) \mid v_s(s,t) \mid^2\right] \mathrm{d}s\mathrm{d}t, \quad (5)$$

where $v_s()$, and $v_{ss}()$ are the first and second order partial derivatives of v with respect to s, and $v_t()$ is the first partial derivative of v with respect to t.

The *Image Energy*, E_{img}, consists of two terms: E_c, that depends on the contrast of the image points corresponding to those belonging to the curve, and E_v, that considers the changes in image contrast with time:

$$E_{img} = E_c + E_v, \quad (6)$$

where:

$$E_c = \int_T \int_s \left[\rho(s) H\left(I\left(x(s,t), y(s,t), t\right)\right)\right] \mathrm{d}s\mathrm{d}t, \quad (7)$$

$$E_v = \int_T \int_s \left[\gamma(s) I_t\left(x(s,t), y(s,t), t\right)^2\right] \mathrm{d}s\mathrm{d}t, \quad (8)$$

and where: $H()$ is a gradient operation, $I(x(s,t),y(s,t),t)$ is the value of the image brightness at time t in the point $(x(s,t),y(s,t))$,$I_t()$ is the first order partial derivative of the image brightness with respect to time, $\rho(s)$ and $\gamma(s)$ are suitable weight functions. It should be noted that the operator $H()$ must be capable of identifying the shape of the curve that must be tracked in the image sequence.

At each time step, the minimization of Equation (1) is reduced to estimate the solution of the system of equations, which in turn has been obtained by taking the derivatives of the functional with respect to the unknowns (i.e., points through which the approximation curves must pass). Thus, a system of $2(p + 1)$ unknowns is defined where $p + 1$ is the number of curve points. Using a curve representation based on B-splines the dimension of the system of equations is strongly reduced since the curve is defined on the basis of the *control points* (i.e., the knots) which are usually much less than the *curve points*:

$$x(s) = \sum_{i=0}^{m} X_i B_i(s); \quad y(s) = \sum_{i=0}^{m} Y_i B_i(s),$$

where $Bi()$ for $i = 0,..,m$ are polynomials defining the basis of the B-spline representation, and (X_i,Y_i) for $i = 0,..,m$ are the knots of the curve. Thus, with this representation the number of unknowns is reduced from $2(p + 1)$ to $2(m + 1)$ where $m << p$, and the equation set can be written as:

$$\begin{aligned} &\mathbf{AX} + \mathbf{G}_x\left(x(s,t), y(s,t), t\right) + \mathbf{VX}_t + \mathbf{E}_{vx} = 0, \\ &\mathbf{AY} + \mathbf{G}y\left(x(s,t), y(s,t), t\right) + \mathbf{VY}_t + \mathbf{E}_{vy} = 0, \end{aligned} \quad (9)$$

where $\mathbf{A}$ is an $(m + 1) \times (m + 1)$ matrix and $\mathbf{G}_x$, $\mathbf{G}_y$ are $(m + 1)$-dimensional vectors.

$$A_{ij} = 2\sum_{h=0}^{p}\left[\alpha(s_h)B_{si}(s_h)B_{sj}(s_h) + \beta(s_h)B_{ssi}(s_h)B_{ssj}(S_h)\right],$$

$$G_{xi} = \sum_{h=0}^{p}\left[\rho(s_h)B_i(s_h)\frac{\partial H\left(x(s_h,t), y(s_h,t), t\right)}{\partial X_i}\right],$$

$$G_{yi} = \sum_{h=0}^{p}\left[\rho(s_h)B_i(s_h)\frac{\partial H\left(x(s_h,t), y(s_h,t), t\right)}{\partial Y_i}\right],$$

$$V_{ij} = 2\sum_{h=0}^{p}\left[\tau(s_h)B_i(s_h)B_j(s_h)\right],$$

$$E_{vxi} = 2\sum_{h=0}^{p}\left[\gamma(s_h)I_t\left(x(s,t), y(s,t), t\right)I_{txi}\left(x(s,t), y(s,t), t\right)\right],$$

$$E_{vyi} = 2\sum_{h=0}^{p}\left[\gamma(s_h)I_t\left(x(s,t), y(s,t), t\right)I_{tyi}\left(x(s,t), y(s,t), t\right)\right],$$

where the above values are estimates for $i,j = 0,..,m$ and $B_s()$, and $B_{ss}()$ are the first and second order partial derivatives of $B()$ with respect to s.

In order to meet the request of real-time computation, it has been necessary to choose linear B-splines as adopted by Menet *et al.* [27]. Therefore, in the following, discrete versions of the above energies have been obtained:

$$E_1 = \frac{p}{\Delta^2}\sum_{i=1}^{p}\alpha(i)\left[\left(x_i - x_{i-1}\right)^2 + \left(y_i - y_{i-1}\right)^2\right], \quad (10)$$

$$E_2 = \frac{p}{\Delta^2}\sum_{i=1}^{p-1}\beta(i)\left[\left(2x_i - x_{i-1} - x_{i+1}\right)^2 + \left(2y_i - y_{i-1} - y_{i+1}\right)^2\right], \tag{11}$$

where $\Delta = \sqrt{\left(x_p - x_o\right)^2 + \left(y_p - y_o\right)^2}$ is the distance between the curve extremes, and the factor $\frac{p}{\Delta^2}$ is inserted for normalizing the energy in order to make the energy measure independent of the scale factor and of the number of knots and points of the snake. As a consequence, the value of E_1 is greater than 1 in most cases and equal to 1 only for rectilinear curves, while $E_2 \geq 0$. The above energies depend on $p + 1$ points and can be expressed in terms of $m + 1$ nodes by using B-splines. Moreover, in order to simplify the calculus the values of weight functions have been chosen to be constant along the curve; therefore:

$$E_1 = \frac{m\alpha}{\Delta^2}\sum_{k=1}^{m}\left[\left(X_k(t) - X_k - 1(t)\right)^2 + \left(Y_k(t)\right)^2\right], \tag{12}$$

$$E_2 = \frac{m\beta}{n\Delta^2}\sum_{k=1}^{m-1}\left[\left(2X_k(t) - X_{k-1}(t) - X_{k+1}(t)\right)^2 + \left(2Y_k(t) - Y_{k-1}(t) - Y_{k+1}(t)\right)^2\right], \tag{13}$$

The structure of the above expressions is equal to that of Equations (10) and (11) since: (i) due to the division of each part of the m parts of the B-spline into n segments having a constant length, each segment $(X_{k-1},Y_{k-1}) - (X_k,Y_k)$ is as long as $1/n$ of the B-spline part, and (ii) $p/n = m$. The same process can also be applied to energy E_t, thus obtaining:

$$E_t = \frac{\tau}{(m+1)\Delta^2(t)}\sum_{k=0}^{m}\left[\left(X_k(t) - X_k(t-\Delta t)\right)^2 + \left(Y_k(t) - Y_k(t-\Delta t)\right)^2\right]. \tag{14}$$

On the other hand, the energies depending on the image brightness cannot be expressed by using only references to the knots since their values also depend on the energy of the intermediate points: therefore:

$$E_c = \rho\sum_{i=0}^{p} H\left(I(x_i, y_i, t)\right), \tag{15}$$

$$E_v = \gamma\sum_{i=0}^{p}\left[I(x_i, y_i, t) - I(x_i, y_i, t-\Delta t)\right]^2 \tag{16}$$

where $x_i = \frac{i-nk}{n}(X_{k+1} - X_k) + X_k, y_i = \frac{i-nk}{n}(Y_{k+1} - Y_k) + Y_k$, and $k = [i/n]$ – i.e., the integer part of the i/n ratio. Hence, by considering the above expressions, the first derivatives of the total energy with respect to the vectors $\mathbf{X} = (X_0,X_1,\ldots,X_m)^T$ and $\mathbf{Y} = (Y_0,Y_1,\ldots,Y_m)^T$ assume the form:

$$\mathbf{E}_{\mathbf{X}}(t) = (\alpha\mathbf{A}_1 + \beta\mathbf{A}_2)\mathbf{X}(t) + \frac{2\tau}{(m+1)\Delta^2(t)}[\mathbf{X}(t) - \mathbf{X}(t-\Delta t)] + \mathbf{C}\left(E_{1(r-1)}, E_{2(r-1)}, E_{t(r-1)}, t\right) - E_{img\mathbf{X}}(t),$$

$$\mathbf{E}_{\mathbf{Y}}(t) = (\alpha\mathbf{A}_1 + \beta\mathbf{A}_2)\mathbf{Y}(t) + \frac{2\tau}{(m+1)\Delta^2(t)}\left[\mathbf{Y}(t) - \mathbf{Y}(t-\Delta t)\right] + \mathbf{C}\left(E_{1(r-1)}, E_{2(r-1)}, E_{t(r-1)}, t\right) - E_{img\mathbf{Y}}(t),$$

where $\mathbf{C}$ is a vector of functions depending on $E_{1(r-1)}$, $E_{2(r-1)}$, $E_{t(r-1)}$, which takes into account the dependence of $\mathbf{E}_{\mathbf{X}}$ and $\mathbf{E}_{\mathbf{Y}}$ on the energies calculated at the previous iteration $(r-1)$; and $E_{img\mathbf{X}}$ indicates a vector whose k-th element is the derivative of E_{img} with respect to X_k. $E_{img\mathbf{Y}}$ is defined in a similar manner. In addition, the structures of the matrices are:

$$\mathbf{A}_1 = 2\frac{m}{\Delta^2}\begin{bmatrix} 1 & -1 & & & & & 0 \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \ldots & \ldots & \ldots & & \\ & & & -1 & 2 & -1 & \\ 0 & & & & -1 & 1 & \end{bmatrix}$$

$$\mathbf{A}_2 = 2\frac{m}{n\Delta^2}\begin{bmatrix} 1 & -2 & 1 & & & & & 0 \\ -1 & 3 & -3 & 1 & & & & \\ 1 & -3 & 4 & -3 & 1 & & & \\ & \ldots & \ldots & \ldots & \ldots & \ldots & & \\ & & 1 & -3 & 4 & -3 & 1 & \\ & & & 1 & -3 & 3 & -1 & \\ 0 & & & & 1 & -2 & 1 & \end{bmatrix}$$

Solving the above system of non-linear equations leads to an estimated value of minimum of the functional representing of the total energy and, thus, the positions of the $m + 1$ knots of the STB-snake at the current time instant.

Unfortunately, if traditional methods for solving non-linear systems of equations are adopted, the solution of the above system of equations can be computationally very heavy.

In order to solve this problem, a specific and very fast method has been defined. According to many other applications in which splines have been used for modeling curves in vision, the first hypothesis is that the initial data is not very far from the final solution. If the deformations are supposed to be slow or the number of images per second high, the above hypothesis can be expanded to be applied to the changes between two subsequent images. According to these conditions, the method proposed is based on the estimation of the sign of the derivatives of total energy with respect to each variable (X_i,Y_i) for $i = 0,..m$. Once the derivatives are estimates, the coordinates of each point (X_i,Y_i) are increased or decreased of a given amount, δ, according to the corresponding sign. This process is performed for each node and for Q iterations (the stop criterion is based on a threshold applied to the value of the derivative of total energy with respect to the iteration number). In order to decrease the number of iterations and, thus, to improve the system performance, experimental results have demonstrated that the value of δ at the generic iteration q,δ_q, can be profitably obtained on the basis of the initial value δ_o and the iteration number: $\delta_q = \delta_o \sigma^q$ where $\sigma < 1$.

With such a definition for δ_q, it can be shown that the maximum variation (increment or decrement) of each coordinate X_i,Y_i, of the generic knot i, is always less than:

$$\delta_{max} = \sum_{k=0}^{\infty} \delta_k = \delta_o \sum_{k=0}^{\infty} \delta^k = \frac{\delta_o}{1-\delta}. \qquad (17)$$

By indicating with $\mathbf{X}_o$, $\mathbf{Y}_o$ the values of vectors $\mathbf{X}$, $\mathbf{Y}$ at the first iteration, the δ_{max} must be chosen so that the values which minimize the total energy at the end of the iterative process lie onto the hyper-cube specified by:

$$\left\{(\mathbf{X},\mathbf{Y}) : \| \left(\mathbf{X}_o,\mathbf{Y}_o\right) - \left(\mathbf{X}_Q,\mathbf{Y}_Q\right) \|_\infty < \delta_{max} \right\}.$$

Once δ_{max} has been chosen, several values for $\delta_o = 10$ can be obtained by using $\delta_o = 2$ and $\sigma_o = 0.8$, as well as by choosing $\delta_o = 5$ and $\sigma_o = 0.5$. By using the first couple of values, δ_q decreases relatively slowly, and the solution is more exact, while for higher values of δ_o (and lower values of σ) the iterative process is faster (less iterations are needed), but the solution is less satisfactorily approximated. Thus, given δ_{max}, the values for δ_o and σ which must satisfy Equation (17), are chosen according to the processor speed, the time available for each frame and the precision required.

This technique allows the estimation of the minima at each time step using only few iterations, typically no more than 10–15 iterations, with $\delta_o = 1$ and $\sigma_o = 0.75$. The values must also be chosen by realizing that the final goal is to reproduce the synthetic model on a screen; this means that an extreme resolution is often unuseful.

Since the process is driven by the image energy (i.e., when the image energy changes the curve following it changes in order to reach the minima), in certain conditions the curves can lose some points because they find a lower energy due to the presence of more prominent image gradients, as it has been noted for classical splines. A typical example is the case in which an open mouth shows the teeth (the appearing of the teeth changes the conformation of the energy surface) (see Figure 3). In particular, in Figure 3 (left) a closed mouth is reported with the corresponding trend of the total energy. In Figure 3 (right), a frame obtained after a time instant with respect to the previous frame is shown and on the left of the same figure the corresponding trend for the total energy is reported. In this case, at the beginning of the estimation process the curve is located in *b* and the presence of a high gradient generates a different minimum with respect to the correct leap. In these conditions, the points are attracted from the center of the mouth. In order to solve this problem, an *ad hoc* energy of repulsion has been defined among the points belonging to the upper and lower parts of the mouth. This factor has been added to the expression of E_{int}, Equation (2), in the complete model, thus obtaining for the total energy a profile modified according to the dashed line reported in Figure 3. This constraint has also been profitably used for eyebrows since their thickness can be considered to be constant in time (Figure 3).

Synthetic Model

In Figure 4, two views of the generic 3D wire-frame model are reported consisting of 105 points which identify the facet i.e., the triangles. This has been adopted as a generic wire-frame model and derived from the well-known CANDIDE model (76 points) [21] by adding points around the mouth and the nose for improving realism and for providing a correspondence between the points of the synthetic model and the knots of the STB-snakes.

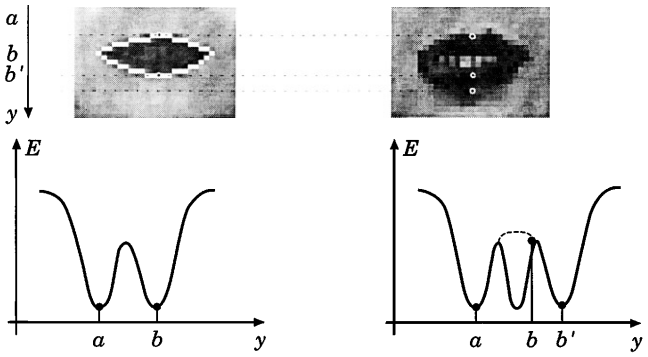

Figure 3. Modeling with STB-snakes: closed mouth (left), and open mouth (right) with the teeth, and the corresponding energy trends. The trend of the total energy with the addition of the repulsion energy is drawn by using a dashed line.

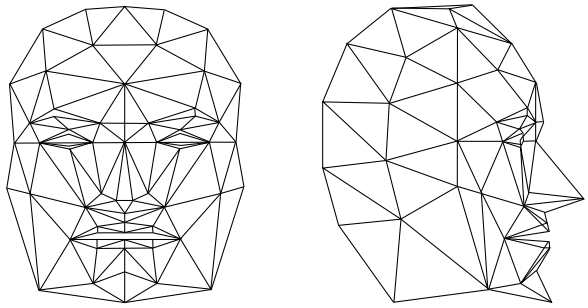

Figure 4. The generic 3D wire-frame model of the head/face: frontal and side views.

Model adjustment

In order to establish the true correspondence between the face under analysis and the synthetic head/face model during the phase of animation, the generic wire-frame model must be adjusted with respect to the real dimension of the face under tracking. To this end, a procedure to adapt the size and shape of the wire-frame facial model to those of the person in front of the camera has been derived and used. This is based on elastic deformations of the model and has been derived from that presented by Reinders *et al*. [23]. The adjustment confers a high realism to the phase of animation of the synthetic model. A better final face model could be produced by also considering the side views of the face under analysis. The process of adjusting is summarized in Figure 5 and proceeds as follows. Firstly, some reference points (corresponding to the most important face features points and to the vertices of the wire-frame model) must be marked. On the basis of these references, the

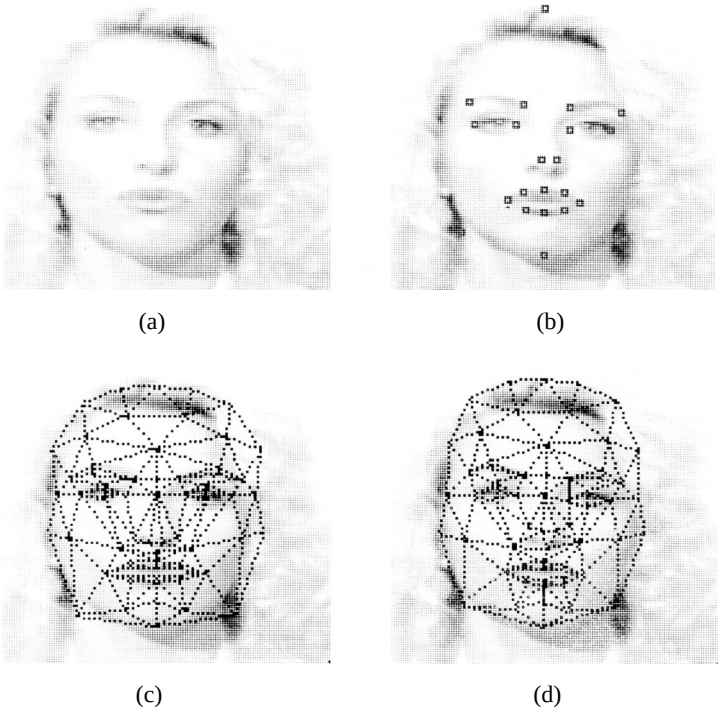

Figure 5. The process of wire-frame adjustment to the actual face dimensions and shape: (a) source image; (b) reference points on the face; (c) image with scaled wire-frame model; (d) adjusted wire-frame model superimposed on the source image (5th iteration).

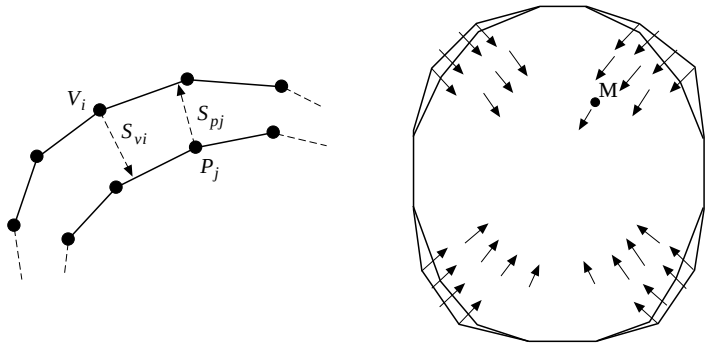

Figure 6. The process of elastic adjustment of the 3D wire-frame head/face mode.

generic wire-frame model of the face is scaled and then, through an elastic process, the model is adjusted with respect to the frontal image. The adjustment is driven by means of an iterative process in which the marked points play the role of attractors and their forces are propagated by using a Gaussian distribution through the edges of the mesh.

In particular, in order to modify the structure of the model vertices locally, the 3D wire-frame model has been assumed to be elastic. With this assumption, the movement of each vertex causes a perturbation of the neighborhood points; a perturbation decreases its effects with the increment of the distance from the vertex considered (see Figure 6). Assume a generic contour of the model, made up of vertices $\mathbf{V}_i$, for $i = 1,..,nv$, which must be modified in order to match the corresponding contour on the face, whose vertices are $\mathbf{P}_j$, for $j = 1,..,np$, (with np which might be different or not from nv), for each vertex, $\mathbf{V}_i$, of the model, a push vector, $\mathbf{S}_{vi}$, which moves $\mathbf{V}_i$ on the face contour (i.e., on the correct position) can be defined. In the same way, for each face vertex, $\mathbf{P}_j$, a pull vector, $\mathbf{S}_{pj}$, which attracts $\mathbf{P}_j$ and brings it on the model contour is defined.

The process of pushing and pulling vectors defines a vector displacement field i.e., in each point **M** of the model a force is present which moves the point itself by a (vector) quantity $\mathbf{D}_\mathbf{M}$, depending on the vectors $\mathbf{S}_{vi}$ and $\mathbf{S}_{pj}$, a scaling factor ϵ, and a rigidity factor ψ. The function defining the vector field is a sort of Gaussian distribution, thus, it is used to weight vectors $\mathbf{S}_{vi}$, and $\mathbf{S}_{pj}$ (29):

The propagation of the forces field produced by the displacement vector depends on the rigidity factor, ψ: the higher the value of ψ the wider the Gaussian distribution. Decreasing ψ, leads to a reduction of the interactions between close points, thus allowing a better local matching. At the beginning of the iterative process, the difference between the two contours can be high; therefore, high values for ψ are chosen. Then, ψ is decreased until the differences between the two contours become less than a predefined threshold. Our experiments have indicated that satisfactory results are achieved even by using $4 \div 5$ iterations which correspond to a couple of seconds on i486 DX 33MHz machines.

Model synthesis

Once the process of adjustment is completed the pattern of the frontal image (called source image) is smeared on the adjusted wire frame model. In this way, a reference synthetic model with the original pattern is obtained. On the basis of the reference model, the effective synthetic head/face is obtained by applying rotations, translations and local deformations according to the measured movements. This process of rendering must be repeated each time the real head/face under analysis changes its position or presents deformations (at least for the triangles that have been changed), considering projection law, hidden area removal, and texture mapping.

$$\mathbf{D}_\mathbf{M} = \frac{1}{\epsilon}\left[\frac{\sum_{i=1}^{nv} S_{vi} exp\left(-\frac{\|M - V_i\|^2}{\psi^2}\right)}{\sum_{i=1}^{nv} exp\left(-\frac{\|M - V_i\|^2}{\psi^2}\right)} - \frac{\sum_{j=1}^{nv} S_{pj} exp\left(-\frac{\|M - P_j + S_{pj}\|^2}{\psi^2}\right)}{\sum_{j=1}^{nv} exp\left(-\frac{\|M - P_j + S_{pj}\|^2}{\psi^2}\right)}\right].$$

Therefore, in order to synthesize the head/face images, it has been necessary initially to establish a technique for the 2D representation of 3D scenes i.e., projection law. Thus, to satisfy real-time requests the orthogonal model has been chosen for the projection law; by using this solution, only six multiplications per point are needed.

On this basis, a hidden line removal algorithm has been developed in order to establish the model facets that have to be displayed, depending on their orientation and/or deformation. A procedure based on the above mentioned projection law, applying rotations, translations and local deformations according to the measured movements has been used for estimating model facet vertexes on the image plane and thus analysing the model facet by facet, instead of line by line. With such an algorithm, the surface external normal vector for every model facet is firstly calculated, then the inner product with the normal of perspective plane is performed: if the product is negative, the facet is not visible and, thus, that face is not drawn. Finally, a simple technique was used in order to avoid superposition errors. The facets corresponding to the nose are the last to be evaluated and displayed, so that the parts of the face possibly concealed by the nose are correctly covered. The use of this method is possible when the nose is supposed to be closer to the observer with respect to the zone around the nose itself (i.e. the face is not oriented backward). In general, the triangles are produced starting from the farthest to the nearest with respect to the observer.

Therefore, the facets of the synthetic reference model that must be displayed, and the new position of each vertex for the model facets have been identified and estimated, respectively with the above algorithm. On this basis, every point of each visible facet must be transferred from the reference model to the actual model by considering the appropriate color (texture mapping). In order to fill a triangle with the appropriate texture, a linear transform has been defined which [for each point of the "destination" triangle i.e., the one to be filled, see Figure 7 (right)] gives the corresponding point on the "source" triangle (left) (the one containing the reference texture map). For a better understanding of the texture mapping procedure, consider a generic triangular facet, and suppose **A**, **B**, **C** to be its vertices projections on the perspective plane (source triangle). Let $\mathbf{A}'$,$\mathbf{B}'$,$\mathbf{C}'$ be the vertices projections of the destination triangle. The mechanism consists of estimating for each image point, $\mathbf{P}'$, of destination triangle $A'B'C'$ the corresponding color of point, **P**, located in the source triangle *ABC* (see Figure 7). This process has been defined in order to guarantee at least the evaluation of a pixel color

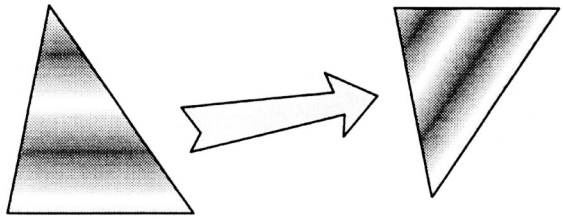

Figure 7. Pattern smearing: reference pattern (left), and destination triangle (right).

for each point of the destination triangle, since the opposite process (starting from the source triangle and projecting each pixel on the destination) can produce several holes in the destination map when strong deformations are present.

To find the corresponding law, a linear transform has been defined $\mathcal{T}(\mathbf{P}') = \mathbf{P}$ – i.e., $\mathcal{T}(\mathbf{A}') = \mathbf{A}$, $\mathcal{T}(\mathbf{B}') = \mathbf{B}$, $\mathcal{T}(\mathbf{C}') = \mathbf{C}$.

Transform, T, can be expressed through a matrix product: $\mathbf{P} - \mathbf{O} = \mathbf{M}(\mathbf{P}' - \mathbf{O}')$ where **O** and $\mathbf{O}'$ are a couple of homologous points of triangles *ABC* and $A'B'C'$ (e.g., $\mathbf{O} = \mathbf{A}$ and $\mathbf{O}' = \mathbf{A}'$), **M** is a 2×2 matrix (of elements m_{11}, m_{12}, m_{21}, m_{22}) while $(\mathbf{P} - \mathbf{O})$, $(\mathbf{P}' - \mathbf{O}')$ are column vectors. On this basis, by considering the vertex of corresponding triangles *ABC* and $A'B'C'$, and tacking as a reference point $\mathbf{A}$,$\mathbf{A}'$, the following equations must hold: $\mathbf{B} - \mathbf{A} = \mathbf{M}(\mathbf{B}' - \mathbf{A}')$, $\mathbf{C} - \mathbf{A} = \mathbf{M}(\mathbf{C}' - \mathbf{A}')$. From these four equations it can be obtained:

$$\begin{aligned} m_{11}\left(x_{B'} - x_{A'}\right) + m_{12}\left(y_{B'} - y_{A'}\right) &= x_B - x_A \\ m_{21}\left(x_{B'} - x_{A'}\right) + m_{22}\left(y_{B'} - y_{A'}\right) &= y_B - y_A \\ m_{11}\left(x_{C'} - x_{A'}\right) + m_{12}\left(y_{C'} - y_{A'}\right) &= x_C - x_A \\ m_{21}\left(x_{C'} - x_{A'}\right) + m_{22}\left(y_{C'} - y_{A'}\right) &= y_C - y_A \end{aligned} \tag{18}$$

The above equations are used to define two systems of two equations for the direct estimation of the coefficients of matrix **M** e.g., Equations (18a,c,d,e). These coefficients are used for the direct estimation of displacement components with respect to point **A** on the reference triangle by using the displacement components from the point $\mathbf{A}'$ (note that the estimation is performed only once per triangle, thus reducing the computational effort):

$$\begin{aligned} \Delta x &= m_{11}\Delta x' + m_{12}\Delta y' \\ \Delta y &= m_{21}\Delta x' + m_{12}\Delta y' \end{aligned}$$

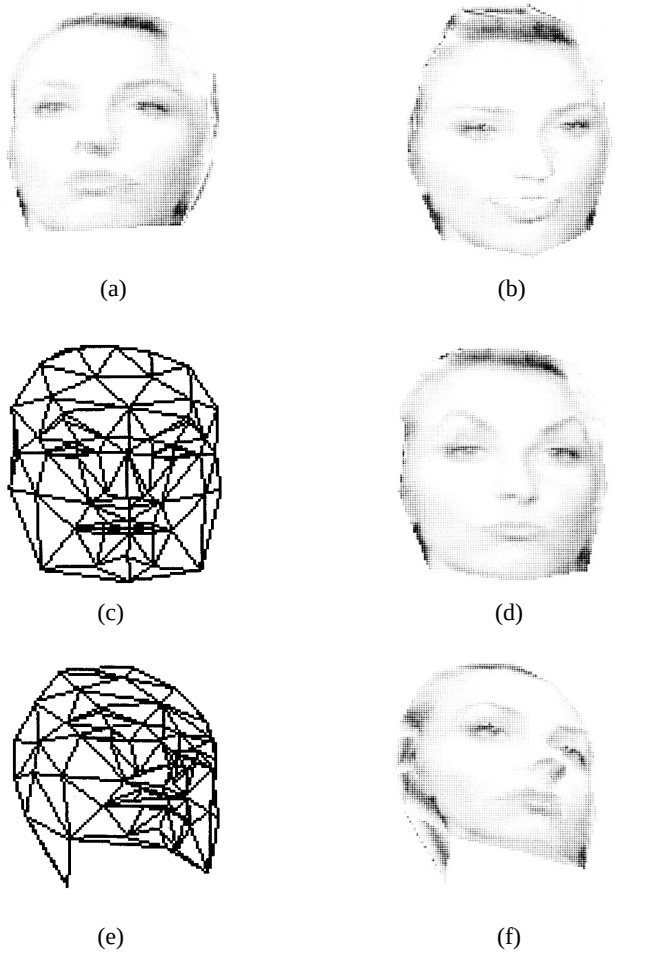

Figure 8. Some examples of animation: (a) rotated model; (b) deformed and rotated model; (c)–(d) wire-frame and patterned models rotated and deformed (see eyebrows and mouth); (e)–(f) wire-frame and patterned models strongly rotated.

The triangles are scanned by using two reference points, **A** and **A**′; then, the point under transformation, **P**′, is moved horizontally pixel by pixel from the left edge of $A'B'C'$ to the right one. When **P**′ reaches the right edge, a new line is scanned in the same way. During the motion of **P**′, **P** is moved according to transform τ. At each step, the color of **P** is copied onto **P**′. These equations are computationally very cheap, since the problem of texture mapping is reduced to the estimation of displacements that must be applied to point **P** on the basis of **P**′ and of the dimensions and orientations of the triangles.

Therefore, the problem is solved by using a two phase process: (i) solving two systems of two equations in two unknowns for estimating transform τ for each triangle, and (ii) the direct estimation of point displacements.

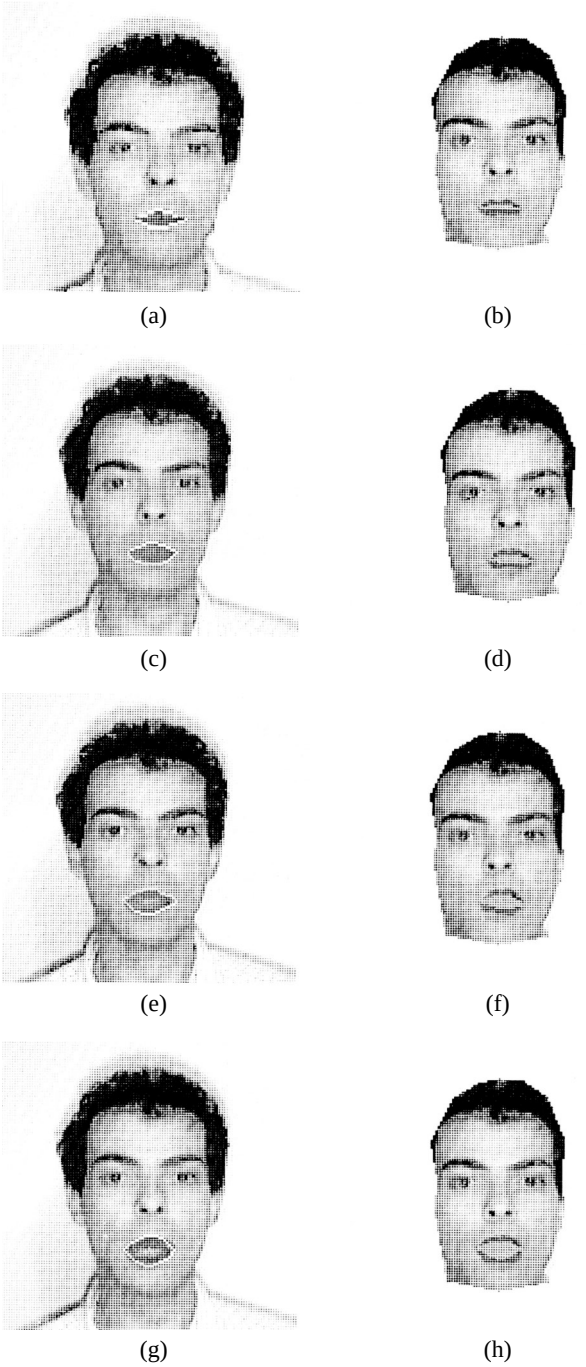

Figure 9. Selected images from a sequence (128 × 218 pixels of resolution) where the face under analysis is opening he mouth: (a), (c), (e), (g) original images; (b), (d), (f), (h) faces synthesized by using the patterned wire-frame model with estimated deformations, with: $\alpha = 1000$, $\beta = 800$, $\sigma = 0.75$, $\tau = 1$, and $\rho = 1$.

In Figure 8, some examples of faces obtained by rotating and deforming the model adjusted in Figure 5 are presented. For two frames, both the wire-frame and the patterned models are shown.

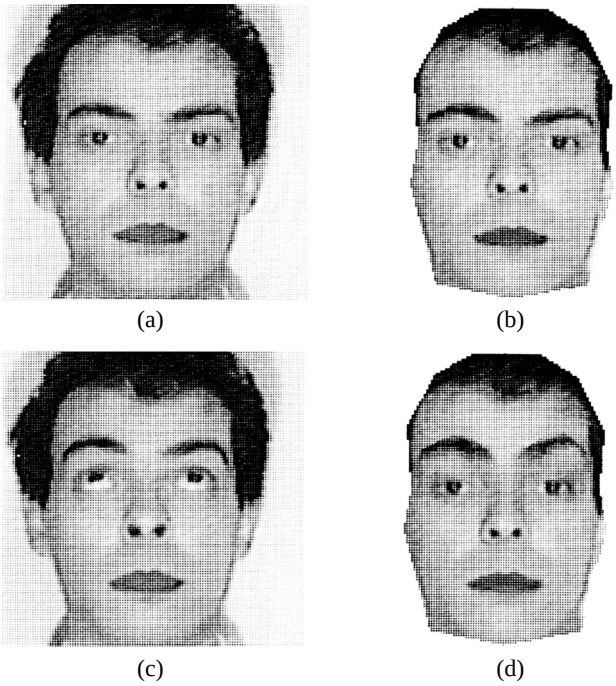

Figure 10. Selected images from an image sequence (128×218 pixels of resolution) where the face under analysis is moving the eyebrows: (a), (c) original images; (b), (d) faces synthesized by using the patterned wire-frame model with estimated deformations, with: $\alpha = 10$, $\beta = 500$, $\sigma = 0.75$, $\tau = 0.1$, and $\rho = 1$. note that in the reconstructed images the eyes are stationary.

Experimental Results

The technique proposed for the notion tracking of face features and the synthesis of estimated deformations on a patterned 3D model has been tested on several real image sequences in which distinct people deform their face. The final application of our methods is the very long-term tracking of faces for video-conferencing, videophones and cinema.

In Figure 9, some image frames of a real sequence where a man is opening his mouth are reported together with the corresponding synthetic reproductions. The snakes estimated have been superimposed on the source images. As can be observed in Figure 9h when the mouth of the synthetic model is open, a gray background is visible. In order to construct a more realistic synthetic model a different pattern can be prepared, for example by presenting synthetic teeth or a structure patterned by a real open mouth.

In Figure 10, some images selected from a sequence where a man is moving his eyebrows are reported. It should be noted that, as a side-effect, he had also opened his eyes. On the contrary, on the right of the same figure i.e., in the images reporting the synthetic reproductions, the eyes are static, since in this case only the eyebrows have been tracked. The same method used for tracking the eyebrows can be adopted for tracking other facial features such as eyelids with the corresponding increment of computational cost. Moreover, eyelids can be tracked simply by following only one point (e.g., the center of eyelid border). On the contrary, for tracking eyes the best results can be obtained by using templates [12,13], since their shape is constant even if sometimes occluded. In Figure 11, some examples of synthesized images obtained by using the deformations estimated from the sequence of Figure 9 are reported. Some of these synthetic images have been obtained by rotating in several directions the synthetic model and/or by assigning the deformations estimated from the sequence of Figure 9 to a difficult model (in particular, the model of a woman). Therefore, in our system, it is also possible to assign the motion of a face to the structure of another. Moreover, the global motions and the deformations estimated can be integrated by global motions and deformations introduced by keyboard or by other means. This opens the way for applications of virtual reality and cinema e.g., a synthetic actor can be animated by using the mimicry of another actor.

Our experiments have demonstrated that the approach proposed for the estimation of face deformations is quite robust with respect to noise, and that it is suitable to track face motions without time limits. Hence, it can be profitably used in non-controlled environments to perform motion tracking in real applications of long-term motion analysis such as videophones, video-conferencing, etc.

The system proposed for tracking facial features differs from other systems presented in the literature since it adopts a specific energy model and is computationally lighter. This is due to the STB-snake model and to the mathematical technique adopted for solving the system of non-linear equations used for estimating the minimum of the functional expressing the total energy.

As a result, the algorithm proposed for motion tracking is computationally very efficient. In fact, our system is capable of tracking a mouth or an eyebrow with 12 images/s (10–15 iterations per frame, 8 knots with 3 points for inter-knot segment) on a 486 DX 33 MHz. By limiting the number of iterations to 10 it is possible to track the mouth and the eyebrows 15 times/s on a 486 DX2 66 MHz.

Moreover, the algorithm for image reconstruction is very fast; it is capable of producing 22 images/s (containing human faces) having a maximum resolution of 128×128 pixels, reproducing rotations, translations, zooming, and

Figure 11. Synthesized images by using the image sequence reported in the previous figure: (a) rotation of the synthetic model of Fig. 9(d); (b) rotation of the synthetic model of Fig. 9(h); (d) synthetic model obtained by using a different wire-frame model and pattern, and the deformations estimated on the image of Fig. 9(a); (e) synthetic model obtained by using a different wire-frame model and pattern and the deformations estimated on the image of Fig. 9(e); (c), (f)–(i) other moved and deformed synthetic faces.

deformations on a 486 DX 33 MHz. Therefore, a quasi real-time head/face motion tracking has been obtained with low-cost architectures.

These measures have been taken independently, since in most of the applications of low bit-rate image compression mentioned in the introduction (e.g., for videophones, video-conferencing, etc.) the analysis and synthesis are executed on distinct machines.

Conclusions

A complete and integrated system for tracking face deformations and for reproducing the corresponding synthetic head/face was presented. The motion estimation process was based on spatiotemporal B-splines for modeling curves associated with the face features that must be tracked. In addition, an algorithm for adapting the generic 3D wire-frame face model to the face under analysis was used. This has conferred a high realism to the simulations of face motions on the reconstructed faces. Experiments have demonstrated that this approach is robust with respect to noise; in addition, it works well even if low image resolution is used. The system proposed differs from others presented in the literature since it adopts a specific energy model for avoiding spline collapsing and is computationally lighter because it is based on STB-snakes and an *ad hoc* numerical method for solving non-linear systems of equations. Therefore, this approach can be profitably used in non-controlled environments

where robust and fast computations are mandatory, such as for videophones, video-conferencing, etc.

Acknowledgements

The authors would like to thank Professor G. Bucci for his valuable suggestions, and Professor R. Forchheimer for the CANDIDE model.

References

1. Li, H., Lundmark, A. & Forchheimer, R. (1994) Image sequence coding at very low bit rates: a review. *IEEE Trans. Imag. Process*, **3**, 589–609.
2. Borri, A., Bucci, G. & Nesi, P. (1994) A robust tracking of 3D Motion. In *Proceedings of the European Conference on Computer Vision, ECCV'94* (Stockholm, Sweden), pp. 181–188, 2–6 May.
3. Fukuhara, T., Umahashi, A. & Murakami, T. (1992) 3-D Motion estimation for model-based image coding. In *Proceedings of the 4th IEE International Conference on Image Processing and its Applications*, (Maastricht, The Netherlands), pp. 69–72.
4. DelBimbo, A., Nesi, P. & Sanz, J.L.C. (1995) Analysis of optical flow constraints, *IEEE Trans. Imag. Process*.
5. Ben-Tzvi, D., DelBimbo, A. & Nesi, P. (1993) Optical flow from constraint lines parametrization, *Patt. Recogn*., **26**, 1549–1561.
6. Nesi, P. (1995) Real-time motion analysis. In *Real-Time Imaging Directions* (Laplante, P. & Stoyenko, A., eds.). IEEE and IEEE Computer Society Press.
7. Kass, M., Witkin, A. & Terzopoulos, D. Snakes: active contour models, *Int. J. Comp. Vis*. **1**(4), pp. 321–331, VISION.
8. Fujimura, K., Yokoya, N. & Yamamoto, K. (1992) Analysis of optical flow constraints, *IEEE Trans. Imag. Process.*
9. Huang, T.C. & Reddy, S.C. (1992) Human face motion analysis. In *Visual Form Analysis and Recognition, Proceedings of International Workshop on Visual Form*, pp. 287–292, New York: Plenum Press.
10. Terzopoulos, D. & Waters, K. (1993) Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Trans. Patt. Anal. Mach. Intell*. **15**, 569–579.
11. Curwen, R. & Blake, A. (1992) Dynamic contours: real-time active splines. In *Active Vision, Proceedings of Rank Prize Workshop Grasmere, England, 1991* (Blake, A. & Yuille, A. eds.), Cambridge, MA: MIT Press.
12. Yuille, A.L., Hallinan, P.W. & Cohen, D.S. (1992) Feature extraction from faces using deformable templates. *Int J. Comp. Vis*., **8**(2) 99–111.
13. Yuille, A. & Hallinan, P. (1992) Deformable templates. In *Active Vision, Proceedings of Rank Prize Workshop Grasmere, England, 1991* (Blake, A. & Yuille, A. (eds.), Cambridge, MA: MIT Press.
14. Mase, K. (1990) An application of optical flow — extraction of facial expression. In *Proceedings of MVA'90 IAPR Workshop on Machine Vision Applications Tokyo*, pp. 195–198.
15. Yuille, A.L., Honda, K. & Peterson, C. (1991) Particle tracking by deformable templates. In *Proceedings of the Joint Conference on Neural Network*, Vol. 1, pp. 7–12.
16. Samal, A. & Iyengar, P.A. (1992) Automatic recognition and analysis of human faces and facial expressions: A survey. *Patt. Recogn*., **25**(1), 65–77.
17. Huang, C.-L. & Chen, C.W. (1992) Human facial feature extraction for face interpretation and recognition. In *Proceedings of 11th 1APR IEEE International Conference on Pattern Recognition*, ICPR'92 B, pp. 204–207.
18. Cheng, K.-T. & Agrawal, V.D. (1992) Initializability consideration in sequential machine synthesis, *IEEE Trans on Comp*. **41**, 374–379. March 1992.
19. Brunelli, R. & Poggio, T. (1993) Face recognition: features versus templates, *IEEE Trans. Patt. Anal. Mach. Intell*., **15**, 1042–1052.
20. Parke, F.I. (1982) Parametrized models for facial animation. *IEEE CG & A*, pp. 61–68.
21. Rydfalk, M. (1987) CANDIDE, A parameterised face. tech. rep., Department of Electrical Engineering, Linköping University, LiTH-ISY-I-0866, Sweden.
22. Morishima, S. & Harashima, H. (1992) Image synthesis and editing system for a multi-media human interface with speaking head. In *Proceedings of the 4th IEE International Conference on Image Processing and its applications*. (Maastricht, The Netherlands), pp. 270–273, IEE.
23. Reinders, M.J.T. & van der Lubbe, J.C.A. (1992) Transformation of a general 3D facial model to an actual scene face. In *Proceedings of 11th 1APR IEEE International Conference on Pattern Recongition, ICPR'92 C*, pp. 75–78.
24. Terzopoulos, D. & Waters, K. (1990) Analysis of facial images using physical and anatomical models. In *Proceedings of 3rd IEEE International Conference on Computer Vision ICCV'90, Osaka, Japan*, pp. 727–732.
25. Viaud, M.-L. & Yahia, H. (1993) Facial animation with muscle and wrinkle simulation. In *Proceedings of 2nd International Conference Dedicated to Image Communicaitn, IM-AGE'COM'93*, (Bordeaux, France), pp. 117–121, France Telecom, UER, EBU.
26. Li, H. Roivainen, P. & Forchheimer, R. (1993) 3-D Motion estimation in model-based facil image coding, (1993) *IEEE Trans. Patt. Anal. Mach. Intell*., **15**, pp. 545–555.
27. Menet, S., Sant-Marc, P. & Medioni, G. (1990) B-snakes: Implementation and application to stereo. In *Proceedings of Image Understanding Workshop, IUW 90*, pp. 720–726, Morgan Kaufmann.
28. Blake, A. & Yuille, A. (1992) *Active Vision, Proceedings of Rank Prize Workshop Grasmere, England, 1991*. Cambridge, MA: MIT Press.