A Physically-based Model with Adaptive Refinement for Facial Animation

Yu Zhang†, Edmond C. Prakash†† and Eric Sung†

†School of Electrical and Electronic Engineering, ††School of Computer Engineering
Nanyang Technological University, Singapore 639798

Abstract

This paper presents a physically-based 3D facial model based on anatomical knowledge for facial expression animation. The facial model incorporates a physically-based approximation to facial skin and a set of anatomically-motivated facial muscles. The skin model is established through the use of a mass-spring system with nonlinear springs which simulate the elastic dynamics of a real facial skin. Muscle models are developed to emulate facial muscle contraction. Lagrangian mechanics governs the dynamics, dictating the deformation of facial surface in response to muscular forces. We show that when surface regions influenced by the large muscular force, the local deformation becomes inaccurate. The conventional method to deal with this problem is using a fine network, but it also increases the cost of computation. We present therefore an approach to adaptively refining the mass-spring facial model to a required accuracy. It generates more pleasing results at low computational expense.

1. Introduction

Human face is among the most important and interesting objects simulated in computer animation, but they are also most difficult to realistically model and animate. Indeed, attempts to model and animate realistic human faces date back to the early seventies [22]. During last three decades, researchers worked extensively on the development of computational models of human face.

In the literature, early works restricted themselves to pure geometric deformations, mostly carried out directly on parametric surface models. However, with physically based modeling paradigms, more realistic facial models arose. There are two major types of physical models. One applies Finite Element Method (FEM) for the solution of the inherent partial differential equations (see e.g. [15]). The other is based on mass-spring system and uses finite difference schemes to solve tissue dynamics (see e.g. [18]). These two methods have common aspect, that is, they all minimize the energy function of the elastic model subject to deformation.

In the FEM techniques, the finite element equations of motion describe the complete (rigid and non-rigid) motion of an object in a single system of equations. Typically these equations are very complex and their solution computationally expensive; as a result, it is less compatible with real-time applications. In the past few years, some global deformation models have been proposed for interactive animation by restricting deformations to the combination of a given set of vibration modes or of a specific class of global deformations [24, 35]. However, such restrictions on the behavior considerably affect the realism of the animation. To reduce computation time, other approaches appeared, allowing real-time simulation of elastic bodies by preinverting the stiffness matrix or applying superposition theorem to use a linear combination of the precomputed displacement of each external nodes [3, 4]. Unfortunately, these methods used quasi-static models, thus losing the dynamic behavior.

The facial models based on mass-spring system feature a linear approximation of the facial surface. The popularity of the mass-spring models is primarily due to its ease of both implementation and integration with the simulation of other bodies. The current state-of-the-art facial animation techniques use 3D facial model discretized at fixed resolution in space. The discretization rate has to be defined by the animator a priori. If too coarse an approximation is employed then an incorrect animation will be generated, but unfortunately the animator will often have no way of knowing how right this solution is. Conversely, if too fine a spring mesh is adopted then a more correct result may be obtained, at the expense of increased computation. Therefore animator often has to go through a series of trials and corrections before obtaining desirable computation. A promising way to optimize the computations, maximizing the overall realism while guaranteeing the efficiency, is to adaptively refine the model according to the complexity of the occurring motion. In facial animation, to ensure a precise geometrical description of the deformation, the spatial sampling (hence the ac-
curacy) is adapted to concentrate the computational load into areas undergoing significant local deformation while the resolution of the stable regions remains unchanged. It in effect can save large amount of calculations while ensuring a realistic facial deformation within a given accuracy threshold.

In this paper, we propose a physically-based model of human face from the anatomical perspective, which has a three-dimensional structure of the skin and muscles. The mechanism of generating facial expressions with our model is very close to the actual one in the human face. The skin model is constructed by using the nonlinear spring mesh which can simulate the elastic dynamics of real facial skin. Three kinds of muscle models are developed to emulate facial muscle contractions. We address the problem of the inaccurate results due to the coarse discretisation by proposing a mechanism for adaptive refining the mass-spring facial skin network to concentrate effort only where it is needed. To refine the facial surface, we define an initial facial approximation and a measure which indicates the accuracy required for this case. To achieve real-time performance, our facial animation system automatically adapts the local resolution at which potential inaccuracies are detected, in order to concentrate the refinement effort in regions that deform the most. Then the refined network is adopted for the further simulation. This mechanism allows more pleasing animation results to be produced at a reduced computational cost.

We will first give a review of the previous work in facial animation and adaptive meshes in Section 2. In Section 3, we present a biomechanical facial skin model, which employs a biphasic strain-stress elastic constitutive relationship to the membrane undergoing deformations. Section 4 describes the muscle modeling process in which the distributions of muscular force are modeled. Section 5 illustrates the motion dynamics and numerical simulation of the facial model. Section 6 details our adaptive refinement approach and its implementation. The subsequent section demonstrates some experimental results. Finally, the paper closes with concluding remarks and future work.

2. Background

Facial animation is a very challenging task. A variety of approaches have been proposed for the deformable animation of synthetic face. Apart from some recent image-based approaches [2, 25] and techniques that using captured facial motion to drive the performance of a synthetic one [12, 34], the methods can be classified into two groups: geometric animation and physically-based animation.

2.1. Geometric Facial Animation

In the geometrical facial animation, the most widely used approach is morphing between fixed and variable polygon topology [22, 31]. This keyframing method is easy to implement and can effectively create facial expressions in a short amount of computational time, but usually requires the animator to model all the facial expressions before hand, thus cannot be easily used for 3D facial animation. Parametric models [7, 23] control facial shape and facial expressions by modifying the parameters that define the face model, eliminating the need for a complete bank of models. Waters [32] presented a facial animation technique which modeled the effects of muscle tensions over a region of skin. While successful in the application, the model was purely geometrical rather than dynamic. It did not respond in a physically realistic fashion to external forces. For facial animation to be flexible and realistic, the force that generated by facial muscle contraction must be modeled. Kalra et al. [14] described interactive techniques for simulating facial muscle actions using Rational Freeform Deformation (RFFD). Tao et al. [28] proposed a mesh-independent free-form deformation model. The common aspect of their approaches is to deform the predefined skin with a space-filling function whose purpose is to deform the skin surface corresponding to desired muscle actions. MPEG-4 allows the animation of synthetic face through Face Animation Parameters (FAPs) [21]. Each FAP defines the animation by specifying feature points and geometric transformation applied on them. Kshirsagar et al. [17] used these FAPs to control their facial animation system.

2.2. Physically-based Facial Animation

Platt and Badler [26] developed an early structural face model which is based on the anatomy of the face. In their model, skin is the outermost level represented by a set of 3D points defining a surface which is movable. Bones represent innermost level which cannot be moved. Muscles are groups of elastic arcs between the two levels. Terzopoulos and Waters [29] has extended the physical model, introducing three layers of deformable mesh for the facial tissue. The three layers correspond to the skin, the fatty tissues, and the muscle, the lower part of which is tied to the bone. Lee et al. [18] produced realistic facial animation using a hierarchical physical model of a human face. By exploiting high-resolution laser scanner data, the structured generic facial mesh can be conformed to individual automatically. Rolf Koch et al. [15] presented an approach for surgical planning using finite element models. They also extended it for animating human emotions [16]. Using finite element model they achieved a better precision in comparison to particle systems. However, the FEM technique
can be quite computationally expensive due to the number of degrees-of-freedom in the stiffness matrix, thus exclude such system from real-time application in facial animation. Face model developed by Wu et al. [36] focuses on the viscoelastic properties of skin: muscles are presented by B-spline patches, which can be specified interactively. Their model is able to generate expressive wrinkles and skin aging effects. All the existing physically-based models discretize at predefined resolution in space. However, compute deformation at fixed resolution can be inefficient: a higher level of detail is required in the highly deformed regions, while a coarser resolution is sufficient in stable areas.

2.3. Adaptive Mesh and Refinement

In [30], Terzopoulos et al. developed adaptive meshes for nonuniformly sampling and reconstructing input data. Adaptive meshes are dynamic fixed-size models assembled as topologically regular collections of nodal masses connected by adjustable springs. The springs can automatically adjust their stiffnesses in order to concentrate nodes near rapid shape variations. Huang et al. [13] developed an adaptive-size algorithm with isometric and conformal motion constraints and applied it in nonrigid motion analysis. Forsythe et al. [9] proposed hierarchical B-spline which provides local refinements of a B-spline surface such that the new patches are added within a specified region. In [11], the idea of refining the sampling rate is applied to a particle system that simulates the mud flow. The particles discretizing the material subdivide and cluster according to a local energy criterion. While this model is adequate for simulating viscous fluids, it is not well conditioned for structured objects like human face. In order to get both contacts precision and real-time simulation, an adaptive subdivision of the surface mesh is used in [19]. A tetrahedral mesh for a finite element simulation of the crack of a stiff object is refined along fracture lines in [20]. Debunne et al. [5, 6] proposed a multiresolution approach to animate elastic deformable materials at interactive rate using finite differences or mixed finite volume/finite element method.

3. The Mass-spring Face Model

In order to faithfully simulate the deformation of the facial skin tissue, it is crucial to investigate the biomechanical nature of soft tissue deformation under applied loads. Experimental data have been collected in Biomechanics about human tissue elasticity [10]. The study shows that tissues do not have a linear response: the curve representing the stretch (strain) of a tissue as a function of the applied force (stress) is typically a J-shaped curve; as the tissue gets closer to tearing, the increase in stretching becomes smaller per additional unit of exerted force. Moreover, the tissue response exhibits hysteresis: the curves for increasing and decreasing force are different. Each branch of a specific cyclic process can be described by a non-linear pseudo-elastic function. Since the difference is insignificant, we approximate the non-linear relationship by a biphasic curve illustrated in Fig. 1.

![Figure 1. Stress-strain relationship of facial tissue](image)

It is often preferable to use triangular rather than quadrangular elements in object modeling, because the triangle (the 2D simplex) is topologically more flexible and fixes irregular meshes of the parameter domain. We construct a coarse 3D face model which is a wireframe of numbered vertices in 3D coordinate space, with connections specified to form triangles (Fig. 2(a)). The face model is composed of 702 vertices and 1284 triangles. Prescribed colors are added to each triangle to form smooth-shaded surface (Fig. 2(b)).

![Figure 2. (a) Wire frame face model (b) Face model with smooth shading](image)

To physically simulate the deformation of the skin, we use the mechanical law of mass-spring system. Networks of masses, connected by springs, attempt to simulate the behavior of deformable bodies using a primitive model for the transmission of energy. The motion of a particle in the system is defined by its physical nature and by the position of other particles in its neighborhood. In our specific case, we only consider the representation of facial surface.
The facial surface is composed by a set of particles with uniform mass density $m$. Their behavior is determined by their interaction with the other particles that define the face surface. In a correspondence with the geometric structure of the 3D face model, each point of the mesh corresponds to a particle in the physical model. To simulate elastic effects of facial skin tissue, we connect each face skin point with its neighbors by massless nonlinear springs of natural length non equal to zero. The nonlinear elastic property of the spring is represented by the biphasic curve shown in Fig. 1. Fig. 3 illustrates the spring connection.

Figure 3. The mass-spring dynamic facial skin model

Suppose an arbitrary skin mass point $x_i$ is connected with one of its neighbors $x_j$ by the spring $j$. The internal spring forces applied on $x_i$ is the resultant of the tensions of the springs linking $x_i$ to its neighbors:

$$ F_{spring}(x_i) = - \sum_{j \in N_i} k_{ij} \left( \frac{|x_i - x_j| - d_{ij}}{|x_i - x_j|} \right) (x_i - x_j) $$

where:

$N_i$ is the set regrouping all neighboring mass points that are linked by springs to $x_i$.

d_{ij} is the natural length of the spring linking $x_i$ and $x_j$.

$k_{ij}$ is the spring stiffness of the spring linking $x_i$ and $x_j$.

$$ k_{ij} = \begin{cases} k_L & \varepsilon_{ij} \leq \varepsilon^c \\ k_H & \varepsilon_{ij} > \varepsilon^c \end{cases} $$

The spring forces are computed by multiplying the elongation from the rest length $d_{ij}$ of the spring with its spring-stiffness $k_{ij}$. The low-strain stiffness $k_L$ is smaller than the high-strain stiffness $k_H$. Like real skin tissue, the biphasic spring is readily extensible at low strains, but exerts rapidly increasing restoring stresses after exceeding a strain threshold $\varepsilon^c$.

4. Modeling of Facial Muscles

The muscles of facial expression are superficial, they are mostly attached to both the skull and the facial tissue. One end of the facial muscle attached to skull is generally considered the origin while the other one is the insertion. Normally, the origin is the fixed point, and the insertion is where the facial muscle performs its action. In human face, a wide range of muscle types exist: rectangular, triangular, sheet, linear, sphincter [33]. Three main types of facial muscles are incorporated in our face model. They are linear, sphincter and sheet muscles.

4.1. Linear Muscle Model

Linear muscle consists of a bundle of fibers that share a common emergence point in bone and pulls in an angular direction. One of the examples is the zygomaticus major which attaches to and raises the corner of the mouth. Fig. 4 illustrates the linear muscle model with the following definitions:

$m^A_j$: arbitrary facial skin point
$m^A_k$: attachment point of linear muscle $j$ at the skull
$m^l_j$: insertion point of linear muscle $j$ at the facial skin
$B_j$: the maximal radius of influence
$\varphi_j$: the maximal angle of influence
$\varphi_{ji}$: the angle between muscle vector and $x_i$

$l_{ji}$: the distance between muscle attachment point $m^A_j$ and skin point $x_i$

Figure 4. Linear muscle model

On contraction, facial regions close to the skin insertion point of a muscle are affected. The effect of facial muscle contraction is to pull the surface from the area of the muscle insertion point to the muscle attachment point. The muscular influence decreases with both the decreasing of the distance from muscle attached point $l_{ji}$ and increasing of the angle from muscle vector $\varphi_{ji}$. It is assumed that there is zero influence at the point of muscle attachment to the bone ($m^A_j$) and that maximal influence occurs at the muscle insertion point ($m^l_j$). Consequently, a fall-off of the muscle force is dissipated through the adjoining tissue in the influence area of the muscle.
\( l_{ji} \) is calculated as:

\[
l_{ji} = ||m_j^A - x_i||
\]

(3)

\( l_{ji} \) and \( \varphi_{ji} \) weight the influence of muscle \( j \) at vertex \( i \) separately for length factor \( \lambda_{ji} \) and angular factor \( \gamma_{ji} \):

\[
\lambda_{ji} = \frac{l_{ji}}{||m_j^A - m_j^j||}
\]

(4)

\[
\gamma_{ji} = \frac{\varphi_{ji}}{\varphi_j}
\]

(5)

\( \lambda_{ji} \) defines the longitude distance of vertex \( i \) to muscle \( j \) normalized to values between 0 and 1. A value around 0, or 1 signifies that vertex \( i \) lies close to the muscle attachment point, or close to the insertion point, respectively. The influence of the muscle \( j \) increase with \( \lambda_{ji} \). \( \gamma_{ji} \) is defined in [0,1] and represents the latitude distance between vertex \( i \) and muscle \( j \). \( \varphi_j \) defines maximal influence angle. Increasing \( \gamma_{ji} \) results in decreasing the influence of muscle \( j \). The muscular force applied at vertex \( x_i \) can be computed as

\[
\overrightarrow{f}_i = \alpha_L \Theta_1(\lambda_{ji}) \Theta_2(\gamma_{ji}) \frac{(m_j^A - x_i)}{||m_j^A - x_i||}
\]

(6)

In equation(6), \( \alpha_L \) is the muscular force scaling factor which controls the magnitude of muscular force. Function \( \Theta_1 \) (Fig. 5 (a)) scales the muscle force according to the length ratio, while \( \Theta_2 \) (Fig. 5 (b)) scales the muscle force according to the angular ratio \( \gamma_{ji} \) at node \( x_i \). We define

\[
d_j = \frac{R_j}{||m_j^A - m_j^j||}
\]

(7)

and

\[
\Theta_1(\lambda_{ji}) = \begin{cases} 
\cos((1 - \lambda_{ji}^\frac{\eta_j}{2}) \cdot \frac{\pi}{2}) & 0 \leq \lambda_{ji} \leq 1 \\
\cos((\lambda_{ji}^\frac{\eta_j}{2} - 1) \cdot \frac{\pi}{2}) & 1 < \lambda_{ji} \leq d_j 
\end{cases}
\]

(8)

\[
\Theta_2(\gamma_{ji}) = \cos(\varphi_{ji} \gamma_{ji}) \cos(\gamma_{ji} \cdot \frac{\pi}{2}) \quad 0 \leq \gamma_{ji} \leq 1
\]

(9)

The constant \( \eta_j \) defines the strength of muscle \( j \). A decrease in the value of \( \eta_j \) increases the muscle influence along the longitude.

4.2. Sphincter Muscle Model

Unlike the linear muscle, the sphincter muscle attaches to skin both at the origin and at the insertion, and contracts around a virtual center. An example is the orbicularis oris, which circles the mouth and can pout the lips. Because sphincter muscles do not behave in a regular fashion, it can be modeled in elliptical shape and can be simplified to a parametric ellipsoid as shown in Fig. 6. The definition of the parameters list are:

\( x_i \): arbitrary facial skin point
\( a \): epicenter of sphincter muscle influence area
\( a \): the semimajor axis of sphincter muscle influence area
\( b \): the semiminor axis of sphincter muscle influence area

![Sphincter muscle model](image)

Figure 6. Sphincter muscle model

The muscular force applied at vertex \( x_i \) is computed as

\[
\overrightarrow{f}_i = \alpha_s \Theta(r_i) \frac{(o - x_i)}{||o - x_i||}
\]

(10)

where \( \alpha_s \) is the sphincter muscular force scaling factor and

\[
\Theta(r_i) = \cos((1 - r_i) \cdot \frac{\pi}{2}) \quad 0 \leq r_i \leq 1
\]

(11)

in equation (11)

\[
r_i = \sqrt{\frac{x_i^2a^2 + y_i^2b^2}{ab}}
\]

(12)
4.3 Sheet Muscle Model

Sheet muscle consists of strands of fibers which lie in flat bundles. The obvious example of this kind of muscle is the frontalis major, which lies on the forehead and is primarily involved with the raising of the eyebrows. A sheet muscle neither emanates from a point source, nor contracts to a localized node. In fact, the sheet muscle is a series of almost-parallel fibers spread over an rectangle area, muscle model is illustrated in the Fig. 7. Two points \( m_j^{A1} \) and \( m_j^{A2} \) specify the attachment line of the sheet muscle. \( m_j^{Ac} \) is the middle point of \( m_j^{A1} \) and \( m_j^{A2} \). Similarly, the points \( m_j^{I1} \) and \( m_j^{I2} \) specify the insertion line of the sheet muscle, and \( m_j^{Ic} \) is the middle point of \( m_j^{I1} \) and \( m_j^{I2} \).

- \( x_i \): arbitrary facial skin point
- \( m_j^{A1} \) and \( m_j^{A2} \): attachment points defining attachment line of sheet muscle \( j \)
- \( m_j^{Ac} \): middle point of sheet muscle attachment line
- \( m_j^{I1} \) and \( m_j^{I2} \): insertion points defining insertion line of sheet muscle \( j \)
- \( W_j \): width of the rectangle zone influenced by sheet muscle
- \( L_j \): length of the rectangle zone influenced by sheet muscle
- \( l_{ji} \): the distance between skin point \( x_i \) and sheet muscle attachment line

\[
\delta_j = \frac{L_j}{||m_j^{Ac} - m_j^{Ic}||} \tag{16}
\]

4.4 Jaw Rotation

In facial modeling and animation, it is very important to rotate the jaw, which makes the animation convincing. Human jaw is composed of two parts: upper and lower jaw. The movable part is the lower jaw. In our model, the jaw contains the vertices of the lower region of the face surface which can rotate around the X and Y axes. The rotation of the jaw can be realized by a coordinate transformation as the following equation:

\[
\begin{pmatrix}
    x_i' \\
    y_i' \\
    z_i'
\end{pmatrix}
= \begin{pmatrix}
    \cos \phi_x & -\sin \phi_x & 0 \\
    \sin \phi_x & \cos \phi_x & 0 \\
    0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
    x_i \\
    y_i \\
    z_i
\end{pmatrix}
\tag{17}
\]

\( \phi_x \) and \( \phi_y \) represent the degrees of rotation around the X and Y axes respectively. The origin of the transformation are both sides of the joint points of the jaw and the upper skull.

5. Dynamics of the Facial Expression Animation

The simulation of facial expressions requires a mapping of the desired facial expression into facial muscle activation. The Facial Action Coding System (FACS) was developed by Ekman and Friesen [8] for this purpose. FACS describes the set of all possible basic actions performable on the human face. Each basic action is called an Action Unit or AU. Such actions are based on the anatomy of the face, each being caused by either a single muscle or a small set of closely related muscles. FACS identifies 46 AUs which, separately or in various combinations, are capable of characterizing any human expression [8].

FACS has been employed in our system for facial expressions generation. Various facial expressions are created by the combination of the contractions of certain facial muscles. In our 3D face model, there are 23 muscles. We use 3 sphincter muscles to represent the orbicularis oris and orbicularis oculi. The other muscles are represented as pairs of muscles which have left and right components. The facial muscle structure is shown in Fig.8.

When facial muscles contract, the facial skin points that are in the influence area of the muscle model are displaced to their new positions. As a result, the facial skin points not influenced by the muscle contraction are in an unstable state, and unbalanced elastic forces propagate through the mass-spring system to establish a new equilibrium state.

It is known that mass-spring systems lead to stiff ordinary differential equations (ODE) imposing computational
constraints onto the solution strategies. Based on the Lagrangian dynamics, the deformable facial model equations of motion can be expressed in 3D vector form by the ODE of type

\[ \mathbf{M} \cdot \frac{d^2 \mathbf{x}(t)}{dt^2} + \mathbf{D} \cdot \frac{d\mathbf{x}(t)}{dt} + \mathbf{K} \cdot \mathbf{x}(t) = \mathbf{F}_{ext} \]  

(18)

The steady state solution of the above system can be computed either by assembling the stiffness matrix \( \mathbf{K} \) and calculating the solution vector of the system, or by traversing the different mass points and iteratively processing each individual equation. We can take the elastic force expression as an external force \( \mathbf{F}_{spring}(\mathbf{x}(t), \mathbf{K}) = \mathbf{K} \cdot \mathbf{x}(t) \), and take \( \mathbf{F}_{spring} \) to the right hand side of the Eq. 18. This new form of the equation will simplify the formulation procedure.

\[ \mathbf{M} \cdot \frac{d^2 \mathbf{x}(t)}{dt^2} + \mathbf{D} \cdot \frac{d\mathbf{x}(t)}{dt} = \mathbf{F}_{ext} - \mathbf{F}_{spring}(\mathbf{x}(t), \mathbf{K}) \]  

(19)

Given \( n \) nodes the Eq. 19 establishes the equilibrium of forces for each of the diagonal mass matrix \( \mathbf{M} \in \mathbb{R}^{3n \times 3n} \) with

\[ \text{diag}(\mathbf{M}) = [m_1, m_1, m_1, m_2, m_2, m_2, \ldots, m_n, m_n, m_n] \]

and describes their positional movement \( \mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_n(t)] \) over time \( t \). \( \mathbf{D} \) denotes the damping matrix. It is a sparse matrix, element of \( \mathbb{R}^{3n \times 3n} \) consisting of \( n \) matrices \( \mathbf{D}_k \in \mathbb{R}^{3n \times 3n} \). \( \mathbf{F}_{ext} \) is a vector of dimension \( 3n \) and represents the vectors of external forces. It can be divided into \( n \) 3-dimensional vector \( \mathbf{f}_k \). In our face model, the net externally applied force is the muscular force.

The solution of the nodal displacement, velocity and acceleration at time \( t + \Delta t \) can be obtained by using numerical integration. In the simulation, we use a Euler-Cromer method [27] to integrate ordinary differential equations. The algorithm runs as follows:

1. Initialization:

- Form face model mass matrix \( \mathbf{M} \) and damping matrix \( \mathbf{D} \);
- Calculate the initial displacement, velocity and acceleration: \( x_0, x_0', x_0'' \);
- Store matrix value in memory.

2. Time loop:

- Calculate the external muscular force vector \( \mathbf{F}_{ext} \) based on the muscle models;
- Calculate the elastic force \( \mathbf{F}_{spring} \) from elastic force equations;
- Solve for acceleration at time \( t + \Delta t \) based on the stored matrix values(\( \mathbf{M}, \mathbf{D} \)), muscular force \( \mathbf{F}_{ext} \) and elastic force \( \mathbf{F}_{spring} \);
- Calculate new velocity and position of the mass point at time \( t + \Delta t \).

6. Adaptive Refinement

6.1. Problem of the Dynamic Model with Fixed Resolution

After having implemented this algorithm, we have tested it in various situations, which led to more or less realistic results. We have been particularly interested in finding the manifestations of the lack of realism of the face model, and then in understanding its cause. We will study here the case of rough approximation of face surface deformed under large muscular force. By increasing the muscle force scaling factor or decreasing the muscle strength factor, the underlying muscles applying force on the mass point with larger magnitude. The integration of the Lagrangian dynamics allows us to compute the displacement fields of the face model from the knowledge of the muscular force applied to it. Fig. 9 shows the resulting deformable face in "Anger" expression.

This example clearly shows one of the problems – the polygonal nature become obvious in the regions affected by the muscular force (in this case the facial regions of eyebrows and two lower sides of nose). When animated, the coarseness of the mesh is revealed because the points of flexibility (the vertices and edges of the polygons) do not fully capture the structure of a human face. As the muscles apply force ununiformly to the mass points in its influence area, in a single triangle, the node near the muscle attachment point receives much larger muscular force than that of the nodes further away form the muscle attachment point. The consequent nonuniform displacements make the area of the affected triangles become larger. Some highly curved face regions therefore lost their curvature. Therefore to retain accuracy areas where the tissue is highly deformable must be sampled with a greater density of nodes.
Whilst the finer network with more discretisations can improve the simulation accuracy, unfortunately the animator will often have no way of knowing how ‘fine’ this discretisation should be. If too fine a network is employed, the increased computation will produce excessively complicated and slower simulations.

6.2. Local Adaptive Refinement

The idea of our algorithm is to refine the physically-based face model depending on the local approximation quality of the simulation. We have seen in the previous section that when coarse approximation of facial surface is simulated the visible creases occur in the region undergoing significant local deformation. As a consequence, we need a refined sampling whenever the deformation of the contractual springs becomes too large. The deformation rate of the springs can be defined as:

\[
\tau = d - d_0 \quad \frac{d_0}{d} \tag{20}
\]

where \(d_0\) is the natural length of a spring and \(d\) is its length at any time \(t\). Our refinement technique relies on the animator specifying a critical deformation rate \(\tau_c\) which governs the accuracy he requires in the finished animation. The spatial discretization of the model will be adapted if following refinement condition is reached:

- If the deformation rate of a spring is greater than the specified critical deformation rate \(\tau_c\), then we would produce an unacceptable simulation, and so must concentrate effort on refining this region of the model.

Our response to the detection of inaccuracy is the addition of masses and springs in the area where an inaccuracy has occurred. If, and only if, the deformation rate of a spring exceeds \(\tau_c\), then a subdivision is applied to the adjacent triangle elements which contain this constituent spring. We use a hierarchical data structure which models the network at different levels of refinement. Such a nested structure would make it possible to quickly adapt the mesh density locally by moving up and down in the hierarchy. At the beginning of the simulation we begin with a conventional mass-spring mesh at level 0. When inaccuracies are detected, mass points and springs are generated in the level above, where the mass of each new point is exactly the same as those in the level below. Fig. 10 illustrates the division of the triangle of the mesh into 4 children at this new level, where each constituent spring is divided into two at its midpoint and 3 new nodes along with 3 new springs are added to the network structure. For each newly added mass point, its initial velocity is obtained by averaging the velocities of the mass points at two ends of the divided spring. For the hierarchical data structure to work we add a level parameter to indicate in which level the points are on. Such recursive subdivision of the initial mesh has the advantage that subdivided and unsubdivided parts of the mesh can interact through common nodes and edges.

In the refinement, the addition of extra masses and springs make us run the risk of altering the properties of the facial skin. In order to guarantee a global and identical behavior of the deformable face model despite the change of resolution, we adopt the following approach: when we add mass point we always use the same consistent mass, so they respond to forces in the normal way, thereby ensuring every mass point in the model behave in the same manner. We double the spring stiffness for each level of refinement to prevent regions of increased mass behaving differently. It increases the stability of the model, and ensures that different refinement levels behave in a similar manner.

Because the response of the body due to its mass is governed by the time in which it is allowed to respond as well as the magnitude of that mass. Once the space discretization has been adapted accordingly, time discretization has
also to be adapted to prevent instabilities. In our model, the integration time step is adapted locally, so that regions undergoing no or litter muscular forces are not updated as often as regions with large muscular forces.

The mathematical results concerning linear differential equations show that their numerical solving is ill-conditioned if integration time step is greater than the natural period of the system [1], given by:

\[ T_0 \approx \pi \sqrt{\frac{\mu}{k}} \]

where \( \mu \) represents the mass of the particles and \( k \) the spring stiffness. As for each level of resolution, the newly inserted springs connecting added mass points have their spring stiffness doubled. The condition mentioned above stipulates that the time step for these added mass points should be less than \( 1/\sqrt{2} \) of the one its parent uses. As a consequence, and in order to synchronize the time steps easily, we choose the time steps to be a half of that used in adjacent lower level.

If the time step at unrefined level is \( T \) (which corresponds to \( \Delta t \) in Section 2), during each time step \( T \) of the simulation, we iterate over the mass-spring model several times, each time simulating the network for a sub time step \( dt \). In the simulation, the mass points which are refined will feel elastic forces from their neighbors and muscular forces on every sub time step \( dt \), whereas the unrefined areas only feel the forces once over the whole time \( T \). If a mass point is at a time step where it does not feel a force then it is assumed to have constant velocity, the value of which will have been calculated during the last time step in which it felt a force.

For this refinement to work, the simulation process of the conventional network should be modified. When an inaccuracy of a triangle element is detected, the algorithm runs in the following process:

1. Back the simulation up a sub time step \( \Delta t \) for the inaccurate mass points which constitute this triangle element.
2. Subdivide this triangle element by inserting new mass points and springs with double stiffness (as shown in Fig. 10).
3. Compute the initial velocities of added mass points by averaging the velocities of mass points at two ends of the divided spring.
4. Rerun that \( \Delta t \) step for both the new mass point and those which have been backed up in order to determine the new position of the facial skin.

7 Results

To demonstrate the physically-based facial modeling approach described and the advantages of adaptive refinement procedure, we have developed a facial animation system which is programmed in C++/OpenGL. The system runs on a SGI Iray Workstation, PII Xeon 550MHz, 512MB.

In the experiment, we used physically-based face model to dynamically generate typical facial expressions such as anger, happiness, surprise, sadness. Fig. 11 illustrates the expressions synthesized by our facial model. Each example shows both the wire-frame mesh, which identifies the nature of the network, and a shaded representation of the facial surface, which shows what a user actually see. It shows that the facial regions which are highly deformed in the simulation have higher spatial resolution for surface approximation than stable areas due to adaptive refinement. Fig. 12 shows the dynamic deformation of face model on “Happiness” in both shaded and wireframe model. The mesh in red shows the facial regions that has been refined. It verifies that the adaptive refinement algorithm can detect the facial regions in which inaccuracy occurs and only add mass points into these regions for hierarchical subdivision.

To demonstrate the ability of the adaptive refinement technique to identify areas of interest and to improve surface smoothness, Fig. 13 compares the simulation results without and with adaptive refinement. In Fig. 13 (a) and (b), the left two images show the shaded and wireframe expressions of “Anger” and “Happiness” without adaptive refinement respectively. Note how obvious the creases occur at two lower sides of the nose and around mouth region. The two images on the right in Fig. 13 (a) and (b) show how the inaccuracy process detects and adaptive refinement is implemented to preserve the surface continuity with a critical deformation rate \( \tau_c = 0.3 \). It clearly shows that highly deformed facial areas have been adaptively refined (red mesh) and the crease effect is totally suppressed with adaptive insertion of mass points in these regions.

Without adaptive refinement a very fine mass-spring network would have been employed. So to compare the improvements of our approach we also perform simulations of the face model with different fixed spatial resolutions. Table 1 shows the computational cost of these simulations. As expected, the adaptive refinement technique offers a good trade-off, taking the advantage of the best discretization rate offered by fine meshes while giving a real-time performance. Because the major computation for the anima-

<table>
<thead>
<tr>
<th>Resolution level</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>Adaptive refinement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes number</td>
<td>702</td>
<td>2673</td>
<td>10453</td>
<td>1376</td>
</tr>
<tr>
<td>Triangles number</td>
<td>1204</td>
<td>5136</td>
<td>20544</td>
<td>2353</td>
</tr>
<tr>
<td>Simulation time(s)</td>
<td>1.4</td>
<td>6.8</td>
<td>28.7</td>
<td>2.7</td>
</tr>
</tbody>
</table>

Table 1. Adaptive refinement vs regular refinement.
Figure 11. Synthesized primary facial expressions with adaptive refinement

Figure 12. Dynamic change of the face in "Happiness"

Figure 13. Comparison of facial expression animation without and with adaptive refinement
tion is consumed by the dynamic simulation, it is clear that the adaptively refined model reduces this considerably since there are fewer particles in the model.

8. Conclusion

We have presented a facial expression animation system based on mass-spring system with adaptive refinement. The 3D face model is physically-based and constructed from anatomical perspective. Facial skin tissue is modeled by a nonlinear spring frame which can simulate the elastic dynamics of real facial skin. Three kinds of muscle models are developed to simulate real facial muscle contraction. We have studied the inaccurate animation results of this coarse approximation surface model. This problem could not be improved by merely use excessively large network without significantly increasing the computational cost. We have therefore proposed a mechanism for adaptively refining portions of the mass-spring facial surface to a required accuracy, concentrating effort only where it needed. We have shown how this method could help us produce more pleasing results at a reduced computational cost.

Our future research is directed towards true 3D volume mass-spring system. We will extend the single layer skin model to the multi-layered structure which fully simulate the real skin tissue. In order to account for individual variation of the different layers and tissue types, the input data of the real skin from CT or MRI source will be incorporated. We also intend to extend adaptive refinement technique to the simulation of this volume skin and develop more optimal schemes for implementing adaptive volume model.

References


