

Real-time Facial Expression Animation on An Individualized Face Using Adaptive Simulation Algorithm

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 hierarchical 3D face model of a specific person that conforms to the facial anatomy for realistic facial expression animation and a new fast algorithm for animating high-resolution facial models. Based on the accurate facial mesh reconstructed from the individual facial measurements, we develop a biomechanical face model with hierarchical structure, incorporating the skin, muscle and skull. For computational efficiency, we devise an adaptive simulation algorithm by categorizing the skin nodes into three subsets based on their dynamic characteristics. It uses either a semi-implicit integration scheme or a quasi-static solver to compute the relaxation by traversing the designed data structures in a breadth-first order. The algorithm runs in real-time and successfully synthesizes realistic facial expressions.

Keywords: Facial Animation, Individualized Face, Hierarchical Structure, Adaptive Simulation Algorithm.

1. INTRODUCTION

1.1. Background and Previous Work

To insert a human personality in a computer generated character, one of the powerful approaches is the modeling of face and facial expressions. In recent years, realistic facial modeling and facial expression synthesis have become very essential for the applications such as teleconferencing, man-machine interface, realistic avatars in virtual reality and surgical facial planning. However, the face is endowed with a complex anatomical structure; there are a multitude of subtle expressional variations on the face; moreover, we as humans have an uncanny sensitivity to facial appearance. Due to these factors, until now, synthesizing realistic facial animation in real-time is still a tedious and difficult task.

The first 3D face was proposed in 1972 by F. I. Parke

[9]. Animation was based on morphing between two masks previous generated and stored in the library of masks. To eliminate the need for a complete bank of models, parametric models are developed to allow the animator to create facial expressions by specifying appropriate set of parameter value sequence [3, 10, 11]. Specifically, for the multimedia communication and very low bit rate applications, some researchers used facial animation parameters (FAPs) defined in MPEG-4 standard to control their facial animation system [5, 8]. To deform the skin surface corresponding to desirable muscle actions, [6] proposed an approach to deform predefined skin with a space filling function. There are also some physically-based models using a particle system to synthesize more nature facial motions [12, 14, 15]. The finite element method is also employed for more accurate calculation of skin deformation, especially for potential medical applications [7].

However, all these models are adapted from a template face with low resolution, and therefore could not represent a personalized face in an anatomically accurate way. The goal of modeling a specific person's face with nature looking display calls for models that are based on real measurements of the structures of human face. Moreover, computational complexity is a fundamental constraint in facial animation. Research on facial animation has generally relied on explicit numerical integration (such as Euler's method or Runge-Kutta methods) to advance the simulation. Unfortunately, all these explicit methods suffer from a problem: very small time steps are required to ensure stability. Therefore, they can only be considered as conditionally stable and can explode numerically during the simulation. In addition, the existing facial animation techniques deal with only the low-resolution models. For the fast computation of the underlying deformation and force model, efficient data structures and algorithms that can process high-resolution models have to be developed.

1.2. Our Approach

Our modeling approach reconstructs an accurate facial mesh based on the individual facial measurements containing both shape and texture information. Having recovered the geometry, we develop a hierarchical face model by taking in to account the facial anatomy. It incorporates a physically-based approximation to facial skin tissue, a set of anatomically-motivated facial muscle actuators and underlying skull. The facial skin model employs a multi-layer structure constituted of different spring sets to simulate the nonhomogeneity of the soft tissue. The skin model takes into account the nonlinear stress-strain relationship of the skin and the fact that soft tissue is almost incompressible due to its liquid components. The underlying muscle and skull layers are integrated with the skin geometry to build up an anatomy-based facial model. Lagrangian mechanics governs the dynamics, dictating the deformation of facial surface in response to muscular forces. For efficient simulation, we propose an adaptive simulation algorithm. It employs either a dynamic or a quasi-static simulator for the numerical simulation by taking advantage of the facts that facial deformations are local and facial soft tissues are well damped. By propagating forces in an ordered fashion through the facial mesh, the governing equation is adapted locally in terms of approximation quality and the computational load is concentrated on the facial regions that undergo significant deformations. The algorithm runs at an interactive rate with flexible facial expressions to be generated.

This paper is organized as follows: Section 2 describes the process of facial data acquisition. Section 3 presents the individualized hierarchical facial model. Section 4 and 5 elaborates on the facial motion dynamics and our adaptive simulation algorithm used for the relaxation. The simulation results and analysis are shown in Section 6. Section 7 gives concluding remarks and our future work.

2. FACIAL DATA ACQUISITION

The geometry and color information of the facial surface is obtained by scanning a subject using a Minolta VIVID 700 DigitizerTM. This scanner can only capture a partial facial region facing the scanner during each scan. To recover the whole face's geometry, scans are taken from several different viewangles of the face. Software manipulation allows the surface range data scanned from different viewpoints to be semi-automatically registered in a common coordinate frame by specifying a set of feature points in each reflectance image (where visible). After registration, the partial face meshes are merged into a surface model that agrees with all the images. The generated triangle mesh consists of $> 10^4$ triangles. Since the computing time depends mostly on the number of triangles, a good adaptive triangulation of the parameter domain is an essential step.

We employ Schroder's adaptive reduction algorithm [13] for thinning and retiling. Fig. 1 shows the adaptively reduced geometry and two views of its texture-mapped appearance.

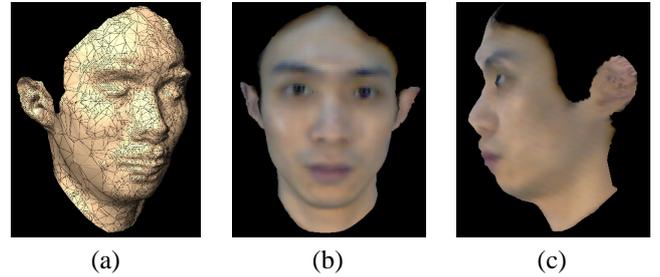


Figure 1. (a) Adaptively reduced facial mesh (3,766 vertices and 7,206 triangles); (b) and (c) Front and side views of texture-mapped facial surface.

3. HIERARCHICAL FACE MODELING

3.1. Multi-layer Soft Tissue Model

In accordance with the structure of real skin, we have developed a multi-layer mass-spring-damper (MSD) tissue model as shown in Fig. 2. In this model, the topmost surface of the lattice is the reduced facial mesh and represents the epidermis. It is a rather stiff layer of keratin and collagen and the spring parameters are set to make it moderately resistant to deformation. The springs in the second layer are highly deformable, reflecting the nature of dermal fatty tissue. Nodes on the bottom-most surface of the lattice represent the hypodermis to which facial muscle fibers are attached. Each mass point \mathbf{x}_i^e of the epidermal layer is connected to the underlying dermal and hypodermal structure by the springs. We distinguish three types of springs:

Layer springs: springs link a vertex on each layer with its neighbors on the same mesh layer.

Connecting springs: springs link a vertex on the layer with its corresponding vertex on the adjacent layer.

Crossed springs: springs link a tissue layer vertex with the neighbors of its corresponding vertex on the adjacent layer.

We use nonlinear springs to simulate nonlinear dynamic behavior of the skin. Suppose an arbitrary soft-tissue point \mathbf{x}_i is connected to one of its neighbors \mathbf{x}_j by a spring with rest length d_{ij} non equal to zero. We introduce a function $K(\mathbf{x}_i, \mathbf{x}_j)$ to modulate a constant spring stiffness k_0 :

$$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + (|\mathbf{x}_i - \mathbf{x}_j| - d_{ij})^2)^\alpha k_0 \quad (1)$$

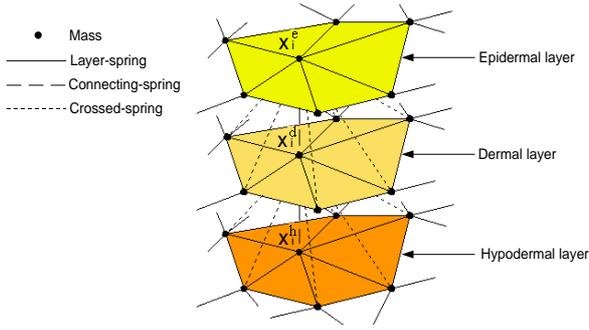


Figure 2. Multi-layer MSD soft tissue model.

and total elastic force applying on node \mathbf{x}_i is:

$$\mathbf{F}_{ela}(\mathbf{x}_i) = - \sum_{j \in \Omega_i} K(\mathbf{x}_i, \mathbf{x}_j) \cdot \frac{(|\mathbf{x}_i - \mathbf{x}_j| - d_{ij})}{|\mathbf{x}_i - \mathbf{x}_j|} \cdot (\mathbf{x}_i - \mathbf{x}_j) \quad (2)$$

where Ω_i is the index set of neighboring mass points of \mathbf{x}_i . In Eq. 1 α is the *nonlinearity factor* controlling the modulation.

In order to simulate the effect of skin volume preservation, we add an external incompressibility constraint to the skin model. In our model, the volume variation is penalized by applying an *incompressibility constraint force* to nodes on the epidermal layer along the normal of each node \mathbf{n}_i (see Fig. 3). The formulation of the constraint force is

$$\mathbf{F}_{incom}(\mathbf{x}_i) = \rho \sum_{j \in \mathcal{N}_i} \left(\frac{V_j - V_j^0}{V_j^0} \right)^2 \cdot \mathbf{n}_i \quad (3)$$

where V_j^0 and V_j are the initial and current volumes of the prismatic element that adjacent to epidermal node \mathbf{x}_i . \mathcal{N}_i denotes the set of the indices of the neighboring elements. ρ is the scaling factor. \mathbf{F}_{incom} acts as a pressure increase inside the prismatic element and keeps the soft tissue volume approximately constant.

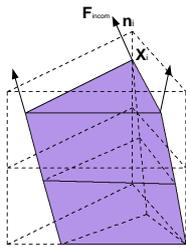


Figure 3. The incompressibility constraint force to penalize element compression.

3.2. Skull and Muscle Modeling

We use a generic skull model that is independent from the skin mesh to map general anatomical attributes to the

individual facial surface. In the skull model fitting process, affine transformations - rotation, translation and non-proportional scaling are applied on the skull model by the user interactively. Fig. 4 shows the fitted skull. For clarity only the hypodermal layer of the skin is shown.

In our previous work [16], we have developed three kinds of muscle models which can efficiently simulate the contraction of linear, sheet and sphincter facial muscles. Based on the Facial Action Coding System [4], we select 23 major functional facial muscles to simulate facial expressions. 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 symmetric about the sagittal plane. We have developed a graphical user interface (GUI) that allows the user to interactively define the central muscle fibers of the linear and sheet muscles based on facial anatomy. By simple mouse clicking and dragging, one end of each fiber can be attached to the skull and the other end can be moved to a skin vertex easily. For the sphincter muscles, an elliptical geometry is scaled to the size of the mouth and the eyes and the contraction center is specified. Fig. 4 illustrates the integrated face model with hierarchical structure of the skin, muscle and skull.

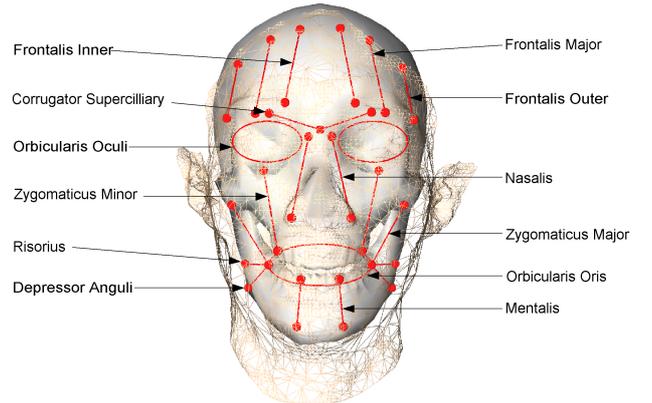


Figure 4. Integrated face model with hierarchical structure.

4. FACIAL DYNAMICS AND SEMI-IMPLICIT INTEGRATION (SII)

Based on the Lagrangian dynamics, the deformable facial model equations of motion can be expressed in 3D vector form by the second-order ordinary differential equation (ODE) of type:

$$\mathbf{M} \frac{d^2 \mathbf{x}(t)}{dt^2} + \mathbf{D} \frac{d\mathbf{x}(t)}{dt} + \mathbf{K}\mathbf{x}(t) = \mathbf{F}_{mus}(\mathbf{x}(t)) + \mathbf{F}_{incom}(\mathbf{x}(t)) \quad (4)$$

We can take the elastic force expression as an external force $\mathbf{F}_{ela}(\mathbf{x}(t), \mathbf{K}) = \mathbf{K}\mathbf{x}(t)$, and take \mathbf{F}_{ela} to the right hand side of the Eq. 4. This new form of the equation will simplify the formulation procedure.

$$\mathbf{M} \frac{d^2 \mathbf{x}(t)}{dt^2} + \mathbf{D} \frac{d\mathbf{x}(t)}{dt} = \mathbf{F}_{mus}(\mathbf{x}(t)) + \mathbf{F}_{incom}(\mathbf{x}(t)) - \mathbf{F}_{ela}(\mathbf{x}(t), \mathbf{K}) \quad (5)$$

Given n nodes \mathbf{x} represents a $3n$ vector of nodal displacement, \mathbf{M} , \mathbf{D} and \mathbf{K} are $3n \times 3n$ diagonal matrices describing the mass, damping and stiffness between nodes in the skin mesh respectively. \mathbf{F}_{mus} , \mathbf{F}_{incom} and \mathbf{F}_{ela} are vectors of dimension $3n$ and represent muscular, skin incompressibility constraint and elastic forces respectively.

The steady state solution of the above system is computed by traversing the different mass points and iteratively processing each individual equation. To simplify notation, we will only consider the governing equation for a single mass m_i . In order to derive the method we divide the second-order ODE for a single mass m_i into a system of first-order ODEs by introducing the velocity function $\mathbf{v}_i(t)$.

$$\begin{cases} \frac{d\mathbf{x}_i(t)}{dt} = \mathbf{v}_i(t) \\ \frac{d\mathbf{v}_i(t)}{dt} = \frac{\mathbf{F}_{mus}(\mathbf{x}_i(t)) + \mathbf{F}_{incom}(\mathbf{x}_i(t)) - \mathbf{F}_{ela}(\mathbf{x}_i(t), \mathbf{K}) - \mathbf{D}\mathbf{v}_i(t)}{m_i} \end{cases} \quad (6)$$

Note that the vector-valued functions can be separated and solved by calculating three scalar differential equations for each component of $\mathbf{x}_i(t)$ and $\mathbf{v}_i(t)$. We assume that at time t_0 the initial values $\mathbf{x}_i^0 = \mathbf{x}_i(t_0)$ and $\mathbf{v}_i^0 = \mathbf{v}_i(t_0)$. Given the position $\mathbf{x}_i^n = \mathbf{x}_i(t_n)$ and velocity $\mathbf{v}_i^n = \mathbf{v}_i(t_n)$ at time t_n our goal is to compute the values \mathbf{x}_i^{n+1} and \mathbf{v}_i^{n+1} respectively at the next time step.

It has been shown that implicit integration methods are superior to explicit ones in case of stiff equation systems [1, 2]. Although a sophisticated implicit solver enables one to use very large time steps, we focus on a simple method to guarantee high update rates for the animation. To find a compromise between the stability of an implicit method and the simplicity and density of an explicit iteration, we start from using an implicit method to describe the calculation of \mathbf{x}_i^{n+1} and \mathbf{v}_i^{n+1} depending on the given states \mathbf{x}_i^n and velocity \mathbf{v}_i^n at time t_n , and the new, unknown states \mathbf{x}_i^{n+1} and \mathbf{v}_i^{n+1} at time t_{n+1} .

$$\begin{cases} \mathbf{v}_i^{n+1} = \\ \mathbf{v}_i^n + \Delta t \frac{\mathbf{F}_{mus}(\mathbf{x}_i^{n+1}) + \mathbf{F}_{incom}(\mathbf{x}_i^{n+1}) - \mathbf{F}_{ela}(\mathbf{x}_i^{n+1}, \mathbf{K}) - \mathbf{D}\mathbf{v}_i^{n+1}}{m_i} \\ \mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta t \mathbf{v}_i^{n+1} \end{cases} \quad (7)$$

where

$$\mathbf{F}_{ela}(\mathbf{x}_i^{n+1}, \mathbf{K}) = \sum_{j \in \Omega_i} P(\mathbf{x}_i^{n+1} - \mathbf{x}_j^{n+1}) \quad (8)$$

and

$$P = -K(\mathbf{x}_i^{n+1}, \mathbf{x}_j^{n+1}) \frac{|\Delta \mathbf{x}_{ij}^{n+1}| - d_{ij}}{|\Delta \mathbf{x}_{ij}^{n+1}|} \quad (9)$$

$$\Delta \mathbf{x}_{ij}^{n+1} = \mathbf{x}_i^{n+1} - \mathbf{x}_j^{n+1} \quad (10)$$

$$K(\mathbf{x}_i^{n+1}, \mathbf{x}_j^{n+1}) = (1 + (|\Delta \mathbf{x}_{ij}^{n+1}| - d_{ij})^2)^\alpha k_0 \quad (11)$$

Note that, one has to solve a nonlinear system of equations for the variables \mathbf{x}_i^{n+1} and \mathbf{v}_i^{n+1} at every time-step. To bypass the computation of this system, we partially substitute \mathbf{x}_i^{n+1} in the upper row of Eq. 7 by insertion of the bottom equation of (7) and solve it for the velocity \mathbf{v}_i^{n+1} . In this case, the solution of the initially implicit method can be expressed explicitly. With some algebra we get:

$$\begin{cases} \mathbf{v}_i^{n+1} = \\ \frac{m_i \mathbf{v}_i^n + \Delta t (\mathbf{F}_{mus}(\mathbf{x}_i^{n+1}) + \mathbf{F}_{incom}(\mathbf{x}_i^{n+1}) - \sum_{j \in \Omega_i} P(\mathbf{x}_i^n - \mathbf{x}_j^{n+1}))}{m_i + \Delta t D + (\Delta t)^2 \sum_{j \in \Omega_i} P} \\ \mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta t \mathbf{v}_i^{n+1} \end{cases} \quad (12)$$

The second equation in Eq. 12 only depends on the variable \mathbf{v}_i^{n+1} and can therefore be computed after evaluating the first equation. But in the first equation, the functions of muscular force, incompressibility constraint force and magnitude operator of elastic force still depend on \mathbf{x}_i^{n+1} . Since they are nonlinear functions, it turns out to be too expensive to extract \mathbf{x}_i^{n+1} at every time-step. To simplify evaluation, we estimate \mathbf{x}_i^{n+1} using an explicit Euler step.

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta t \mathbf{v}_i^n \quad (13)$$

By inserting this estimated position value into the first equation in Eq. 12 we then solve for new velocity \mathbf{v}_i^{n+1} . Given \mathbf{v}_i^{n+1} , we trivially compute \mathbf{x}_i^{n+1} .

5. ADAPTIVE SIMULATION ALGORITHM (ASA)

The facial skin is based on a multi-layer MSD mesh defined over an arbitrary triangulation of the parameter domain. As stated in Section 2, the facial surface reconstructed from the range data easily generates thousands of triangles. Although in the preprocessing procedure the adaptive triangulation of the parameter domain has significantly reduced the complexity of the mesh, the thinned model still contains about 20k triangles (three layers) in order to ensure a precise geometrical description of the anatomical structure of the face. When solving the system of governing differential equations with the semi-implicit integration (SII) method presented in Section 4, the number

of triangular patches has a direct impact on the computing time and memory consumption. This implies that even with more powerful computers, only facial models with several hundreds of vertices can be dynamically simulated at a rate compatible with real-time graphic animation.

In order to speed up the simulation rate, we developed an adaptive simulation algorithm (ASA) to reduce the computational cost. The algorithm takes advantage of following characteristics of the facial deformation: the local deformation and well damped soft tissue. From the analysis of various expressions generated with muscle's contraction we learned that most significant changes of the continuum occur on average in the direct vicinity of the applied external muscle forces, while mass nodes further from this region taking a smaller influence on the nodal displacement. In addition, in the facial model there are also "fixed nodes" whose position is always stationary no matter what the facial motion, such as the nodes near ears and nodes near the bottom of the neck etc. Second, the facial soft tissue, which is essentially made of liquid components, is well damped. As unbalanced forces propagate through the facial tissue, the damping dissipates kinetic energy through friction. In the biomechanical skin model, the damping parameters are with relatively high value to reflect this property. This makes the basis for our assumption that the velocity of the nodes near the muscle influence areas is small enough so that the mesh achieves static equilibrium at each instant.

Based on the above fact, ASA uses a *dynamic* and a *quasi-static* simulator. The dynamic simulator uses the semi-implicit integration scheme described in Section 4 to solve the governing motion equation, while in a large portion of the facial skin a simpler algorithm based on quasi-static assumptions gives realistic results at a much faster rate.

We divide node set V of the facial model into three subsets: V_d , V_s and V_q . *Dynamic node set* V_d represents the set of nodes that are directly subject to external muscle forces. In the facial model, V_d corresponds to the portions of the skin that are in the muscle influence area. The equilibrium position of each node in V_d is computed online by using SII to solve dynamic equation until the total force on each node is zero. Since facial deformations resulting from muscle contractions are local, there are a large number of nodes that remain static in the facial animation. They are categorized into *static node set* V_s . In the preprocessing, we compute the deformation of the skin mesh once by solving Eq. 5 using SII. If the position of a node is modified by less than a small pre-specified threshold, then it is considered as a node of V_s and ASA stops propagating the deformation further. By cutting out the evolution of the static nodes, large amount of computation can be saved. The remaining nodes on the face receive no muscle forces but are still displaced to the new positions due to the propagation of unbalanced forces through the MSD lattice. They are grouped

into the *Quasi-static node set* V_q . By neglecting dynamic inertial and damping forces, the shape of face is defined by a system of equations expressing that each node $\mathbf{x} \in V_q$ is in static equilibrium:

$$\mathbf{F}_{ela}(\mathbf{x}(t), \mathbf{K}) = \mathbf{F}_{incom}(\mathbf{x}(t)) \quad (14)$$

The quasi-static algorithm runs as following:

1. Compute the positions of all the dynamic nodes.
2. Compute the residual force $\mathbf{f}(\mathbf{x}) = \mathbf{F}_{ela}(\mathbf{x}) - \mathbf{F}_{incom}(\mathbf{x})$ exerted on each quasi-static node.
3. Displace the quasi-static node along the acquired residual force.

It is desirable to update the position of each quasi-static node using the most recently computed positions of the neighboring nodes in step 2, rather than those computed at the previous iteration. This scheme will be most advantageous when nodes are traversed in an order starting at the dynamic nodes and expanding towards the quasi-static nodes farther away from any dynamic node, while in practice the quasi-static nodes may be arbitrarily distributed over the mesh. In our simulation, the traversal of the nodes operates in a wave-propagation order which is precomputed by a breadth-first scan of the skin mesh. Fig. 5 gives an illustration of the traversal method. Let q indicates the topological distance of a node from the outermost dynamic node measured in terms of the smallest number of edges between them. The quasi-static nodes are ordered based on the value of q . The outcome of node ordering is a list of nodes such that if index i appears before index j then the topological distance q_i is less than or equal to q_j . ASA processes the nodes in V_q as they have been ordered in the list.

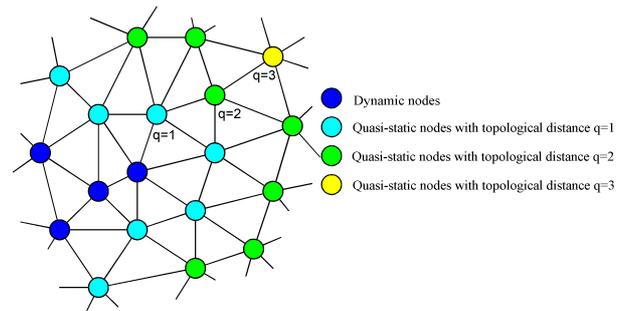


Figure 5. Breadth-first traversal strategy for quasi-static simulation.

The data structure of categorizing the mesh nodes into three subsets enables ASA to be implemented efficiently. While this algorithm is executed, simulation of different expressions only requires re-grouping three node sets and re-invoking quasi-static node ordering to compute a new node

list. In this way, the number of nodes dynamically or quasi-statically treated (hence the computational cost) in simulating each expression adapts automatically.

6. RESULTS

The implementation of our approach has been integrated into a facial animation system which is programmed with C++/OpenGL and runs on an Intergraph Zx10 with dual Pentium III 730MHz, 512MB memory. Fig. 6 illustrates the neutral face and typical expressions synthesized by our system. For clarity, different views of the face model are shown.

We also quantitatively evaluate the performance of the proposed ASA in simulating five typical facial expressions on the hierarchical face model as shown in Fig. 6. By comparing the positions computed by ASA to the actual equilibrium positions, we can measure how the accuracy of the simulation degrades. In each expression simulation, we use both ASA and SII to calculate facial deformation. After each iteration we recorded the 3D position of each node. Errors are computed by the distances between the positions obtained by ASA and the actual equilibrium positions found by running SII. Fig. 7 plots maximum and average errors of simulating face model to generate five different expressions. It shows that errors of simulating “sadness” and “disgust” are larger than those of other three expressions. The more complex nature (more facial muscles involved) of these two expressions leads to fewer nodes that are treated as static nodes, which contributes to increasing the average error. The maximum error has the similar pattern and remains at most about five times of the average error for all cases.

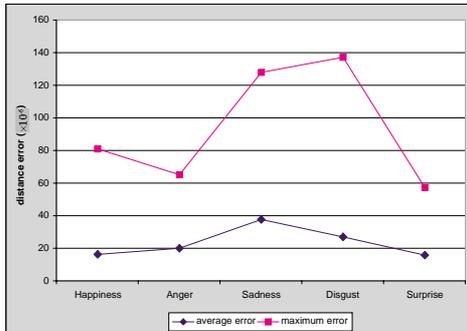


Figure 7. Maximum and average errors in simulating different expressions.

The proposed ASA’s performance can be broken down into the different processes. These processes are precomputing, dynamic simulation, quasi-static simulation and rendering. We measured consumed time for each process. Table 1 displays the algorithm’s performance. The graphs in Fig. 8 plot the performance of ASA and SII for comparison.

Expressions	N_d	N_q	N_s	T_p (s)	Computational time of an iteration (ms)			Frame-rate (fps)
					T_d	T_q	T_r	
Happiness	1599	3516	6183	16.92	14.7	8.0	5.0	36.1
Anger	2883	3069	5346	20.26	27.6	7.0	5.0	25.3
Sadness	3416	3981	3901	22.37	33.8	9.2	5.0	20.8
Disgust	3074	3248	4976	19.85	30.3	7.8	5.0	23.2
Surprise	2548	3218	5532	18.45	21.7	7.2	5.0	29.5

Table 1. Performance of our system using ASA. Notation: number of dynamic nodes (N_d), number of quasi-static nodes (N_q), number of static nodes (N_s), precomputing time (T_p), dynamic simulation time (T_d), quasi-static simulation time (T_q) and rendering time (T_r).

In the precomputing process of each expression simulation, the facial skin nodes are automatically categorized into the dynamic, static and quasi-static node sets. Since precomputing is executed offline, we illustrate the computational time of it and those of other processes separately. To compare the improvements we also performed simulations by using SII alone. Fig. 8 (a) shows the time consumed for precomputing in the simulation of typical expressions shown in Fig. 6 by using both ASA and SII. As expected, ASA introduces extra load, the time cost is about three times that of SII. The bulk of computation in the overhead is consumed by the dynamic simulation of the whole face once, nodes grouping and quasi-static node ordering also contribute to the extra computation. However, the increase of the precomputing time has no impact on the simulation speed because of its offline treatment.

Fig. 8 (b) shows the simulation time of each process for animating different expressions by using both ASA and SII which only consists of semi-implicit integration and rendering processes. Note that all the processes are measured in the average consumed time for a frame. The bar [Ex-ASA] and [Ex-SII] denote each expression Ex simulated by using ASA and SII respectively. As shown in Fig. 8 (b), in the simulation of each expression the dynamic simulation time is dominant in both methods. Comparing to that of SII, the time consumption of the dynamic simulation in ASA is considerably reduced due to fewer portions of the face model involved in the dynamic evolution. The proportion of the computation caused by quasi-static simulation in ASA is relative small compared to that of the dynamic simulation. Since the mesh size for texture-mapping and rendering is the same in simulating different expressions by using both methods, the rendering time remains stable and is short (5ms). The average time reduction rate of each frame by using proposed ASA is about 67.3%. This reduction rate is various

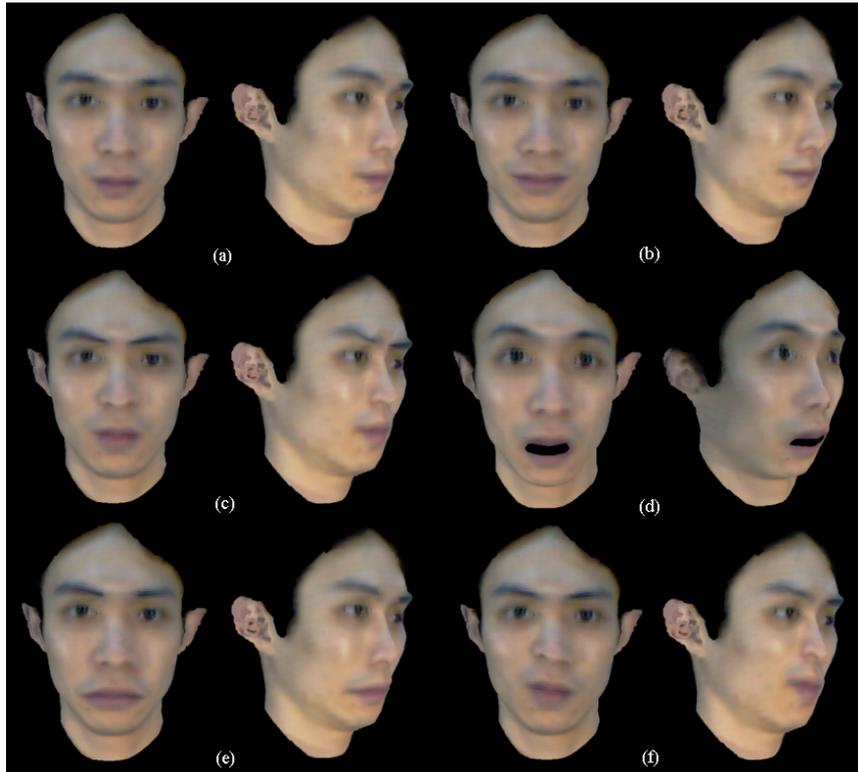


Figure 6. Typical expressions generated by our system: (a) neutral face; (b) happiness; (c) anger; (d) surprise; (e) sadness and (f) disgust.

for the simulation of different expressions. For the expression requiring fewer facial muscles such as happiness, the time reduction rate is up to 74.4% since fewer mesh nodes have to be dynamically evaluated. For the expression generated by the contractions of more muscles such as sadness, the number of dynamic nodes is increased, but the time reduction rate can still reach 60.7%.

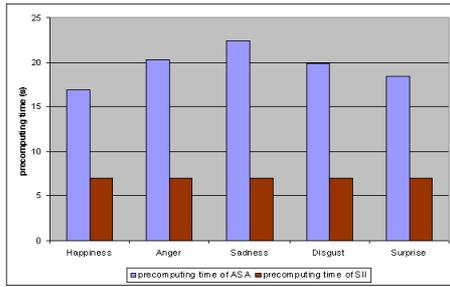
As a function of N_d and N_q which represent the number of dynamic and quasi-static nodes respectively, the simulation rate of ASA is various in animating different expressions. Fig. 8 (c) plots the simulation speed of different expression simulations using ASA and SII, and we can see ASA considerably increases the simulation speed for simulating each expression. For the expression generated by the contractions of more muscles such as sadness, although N_d and N_q are increased, the framerate can still reach about 20 fps.

These experiments clearly verify that for simulating expressions on a high-resolution face model, the proposed new fast algorithm of Section 5 can reasonably maintain a small relative error while gracefully accelerating the simulation speed that would have been impossible otherwise.

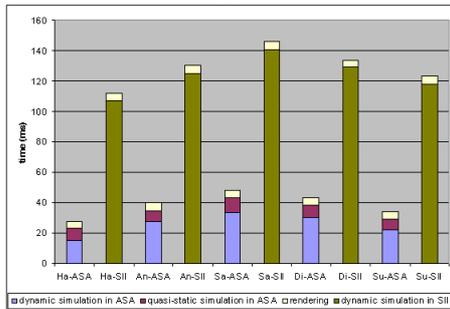
7. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a hierarchical facial model constructed from anatomical perspective for realistic and fast facial expression animation. Based on the accurate facial mesh reconstructed from laser range scans, we developed a physical face model with hierarchical structure, incorporating the skin, muscle and skull. The proposed skin model has a multi-layer MSD structure constituted of different spring sets to simulate the inhomogeneity of the soft tissue by taking into account the nonlinear stress-strain relationship and the incompressibility of the skin. We have proposed an adaptive simulation algorithm to accelerate the numerical simulation. This algorithm takes the advantage of the fact that facial deformation is local and that skin tissue is well damped. By using either a semi-implicit integration method or a quasi-static solver for the relaxation, it concentrates computational load to the facial regions that undergo significant deformations. The resulting system enables us to construct realistic face models of living humans and to generate flexible facial expressions at an interactive rate.

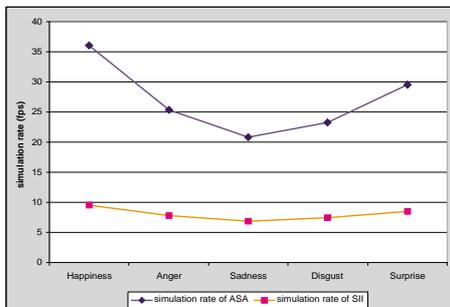
The ongoing work focuses on improving the multi-layer skin model based on tetrahedralizations. The volume tetra-



(a)



(b)



(c)

Figure 8. (a) Precomputing time of ASA and SII in synthesizing different expressions; (b) Comparison of consumed time of each process for synthesizing different expressions between ASA and SII; (c) Simulation speed of synthesizing different expressions using ASA and SII.

hedral meshes will integrate both the reconstructed facial surface and layer springs thus providing much more topological and geometric flexibility for the efficient representation of complex anatomical structures. A more sophisticated approach will be developed for an automatic and more precise skull fit. In order to create more complex and individualized expressions, we plan to exploit facial electromyographic (EMG) data to develop realistic-looking temporal profiles for facial movements.

References

- [1] D. Baraff and A. Witkin, "Large steps in cloth simulation", *Proc. SIGGRAPH'98*, vol.32, pp. 43-54, July 1998.
- [2] Desbrun, M., Schroder, P., Barr, A. "Interactive animation of structured deformable objects", *Proc. Graphics Interface'99*, Kingston, 1999.
- [3] S. DiPaola, "Extending the range of facial types", *Journal of Visualization and Computer Animation*, 2(4): 129-131, 1991.
- [4] P. Ekman and W. V. Friesen, *Facial Action Coding System*, Consulting Psychologists Press Inc., 577 College Avenue, Palo Alto, California 94306, 1978.
- [5] T. Goto, M. Escher, C. Zanardi, and N. Magnenat-Thalmann, "MPEG-4 based animation with face feature tracking", *Proc. Eurographics Workshop on Computer Animation and Simulation'99*, pp. 89-98, 1999.
- [6] P. Kalra, A. Mangili, N. Magnenat-Thalmann, D. Thalmann, "Simulation of facial muscle actions based on rational free form deformations", *Proc. EUROGRAPHICS'92*, pp. 59-69. Cambridge, 1992.
- [7] Koch R., Gross M., Carls F., Buren D., Fankhauser G. and Parish Y., "Simulating facial surgery using finite element models", *Proc. SIGGRAPH'96*, vol.30, pp. 421-428, August 1996.
- [8] J. Ostermann, "Animation of synthetic faces in MPEG-4", *Proc. Computer Animation'98*, pp. 49-55, June 1998.
- [9] F. I. Parke, *Computer generated animation of faces*, Master's thesis, university of Utah, Salt Lake City, June 1972.
- [10] F. I. Parke, "Parameterized models for facial animation", *IEEE Computer Graphics and Application*, 2(9): 61-68, November 1982.
- [11] M. Patel and P. Willis, "FACES: the facial animation, construction and editing system", *Eurographics'91*, pp. 33-45, 1991.
- [12] Platt S., Badler N., "Animating facial expressions", *Proc. SIGGRAPH'81*, vol.15, pp. 245-252, 1981.
- [13] William J. Schroeder, Jonathan A. Zarge, and William E. Lorensen, "Decimation of triangle meshes", *Proc. SIGGRAPH'92*, vol.26, pp. 65-70, July 1992.
- [14] D. Terzopoulos and K. Waters, "Physically-based facial modeling, analysis and animation", *The Journal of Visualization and Computer Animation*, vol.1, pp. 73-80, 1990.
- [15] Y. Wu, P. Kalra, L. Moccozet, and Nadia M. Thalmann, "Simulating wrinkles and skin aging", *The Visual Computer*, 15(4):183-198, 1999.
- [16] Y. Zhang, E. C. Prakash and E. Sung, "A physically-based model with adaptive refinement for facial animation", *Proc. Computer Animation 2001*, pp. 28-39, Nov. 2001.