Physically-Based Forehead Animation including Wrinkles

Mark Warburton
Department of Computer Science,
The University of Sheffield,
M.Warburton@dcs.shef.ac.uk

Steve Maddock
Department of Computer Science,
The University of Sheffield,
S.Maddock@sheffield.ac.uk

Abstract

Physically-based animation techniques enable more realistic and accurate animation to be created. We present a fully physically-based approach for efficiently producing realistic-looking, natural animations of facial movement, including animation of expressive wrinkles. This involves simulation of detailed models using a GPU-based total Lagrangian explicit dynamic finite element (TLED FE) system. The flexibility of our approach enables detailed animations of gross and fine-scale soft-tissue movement to be easily produced with different muscle structures and material parameters, for example, to animate different aged skin. While we focus on the forehead, our approach can be used to animate any multi-layered soft body.

Keywords: physically-based animation, facial animation, soft-tissue animation, wrinkle animation, finite element method

1 Introduction

Facial modelling and animation is one of the most challenging areas of computer graphics. Currently, most facial animation requires recorded performance-capture data or models to be manipulated by artists. However, using a physically-based approach, the effects of muscle contractions can be propagated through the facial soft tissue to automatically deform the model in a more realistic and anatomical manner.

Physics-based soft-tissue simulation approaches often focus on either efficiently producing realistic-looking animations for computer graphics applications [1, 2], or simulating models with high physical accuracy for studying soft-tissue behaviour [3, 4] or surgical simulation [5, 6]. The former simulate large areas, such as the face, with just enough physics to efficiently produce the desired realism of animations, and normally only simulate gross movement using physics-based techniques, rather than fine details like wrinkles [7, 8]. Such approaches often use the efficient mass-spring (MS) method [2, 9], or a physics engine that focusses on performance and stability [10, 11]

On the other hand, applications with high accuracy requirements often use much more detailed models accurately to simulate small areas, like a block of skin for wrinkle simulation [4, 12], or they involve small deformations [5, 13]. These approaches are normally required to use the more accurate but computationally-complex finite element (FE) method [6, 3], or the FE-based but precomputation-heavy mass-tensor (MT) method [14]. Indeed, increases in computational power, and the use of GPU computing architectures mean that complex FE simulations are now possible in real time [15].

By modelling more physics-based behaviour than current computer graphics approaches, the aim of this research is to develop a fully physically-based approach for efficiently producing realistic-looking, natural animations of facial movement, including animation of expres-

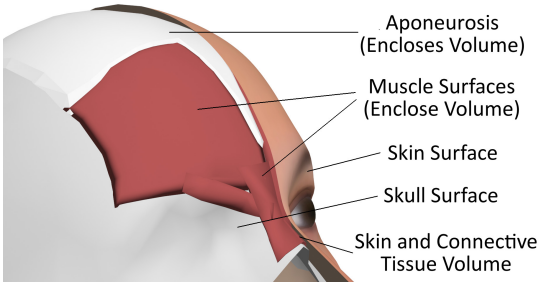

Figure 1: Surfaces and volumes of a forehead soft-tissue model. The volume between the skull and skin surfaces is discretised and simulated.

sive wrinkles, focussing on the forehead region of the face. This involves simulation of forehead soft-tissue models (see Figure 1) using the accurate non-linear total Lagrangian explicit dynamic (TLED) FE method, which has been implemented on the GPU for increased performance. Our approach can also be used to animate any multi-layered soft body (not just soft tissue). The following sections discuss related work, followed by an overview of our animation approach, and details of the simulation process, finishing with example animations.

2 Related Work

2.1 Physically-Based Facial Animation

Physically-based facial animation systems for computer graphics applications normally consist of muscle and skin models, sometimes along with a skull model and wrinkle models. Skull models range from low-resolution offsets from facial surfaces [16], to accurate models from medical data [8], facilitating more anatomical muscle attachments and collision detection. For increased realism of jaw movement, a rotatable mandible can be used [2].

Muscles can be represented as vectors [17], although more realistic approaches consider geometric [2], or physics-based volumes [18]. Many muscle contraction models are based on a Hill-type model [18], some of which are biologically inspired [12]. The direction of contraction can be approximated as parallel to a central action curve [19], or, more anatomically, by using a fibre field [8].

Various facial soft-tissue models have been proposed, including physics-engine-based models that focus mainly on efficiency and stability [10, 11]. MS models can also be used, and many MS facial models are based on Terzopoulos and Waters' layered model [1], which was modified by Lee et al. [16], with various enhancements and different muscle models [9]. With MS systems, additional functions to simulate, for example, incompressibility and volume preservation, are necessary as these models by default contain no volume representation [2]. While uncommon due to their lack of accuracy, the MS method, and variations of it, can be used for surgical applications [20].

More anatomical and realistic facial models have been developed using the FE method [8, 13]. Muscle activations can be estimated from performance capture data to drive FE facial models [8]. However, the FE method is mainly used with scientific [13, 12] or surgical applications [5, 6] that require high accuracy, which normally use simple linear isotropic constitutive models to simulate small displacements.

2.2 Skin Wrinkle Animation

For computer graphics applications, skin wrinkles are usually layered onto a mesh using texture-mapping techniques [21], which are computationally efficient and suitable for fine wrinkles, or by using a heuristic function to modify the geometry of the mesh [22], which can produce more realistic looking wrinkles. Geometric approaches require a high-resolution mesh, although the resolution of meshes can be adaptively refined when required during animations [23]. Some approaches require the wrinkles to be specified by the user [24], which can be blended between poses, whereas others are calculated automatically [25], although with less anatomical accuracy. Performance-capture techniques can also be used to produce wrinkle animations [26], which require the facial or wrinkle model to be configured to the performer. Rather than using physics, geometric or texture-mapping wrinkling approaches have also be used to layer expressive and aging wrinkles onto MS facial models [27, 7].

2.3 *Detailed Soft-Tissue Simulation*

Multi-layered FE models of small areas of soft tissue have been developed for accurate simulation of expressive or aging wrinkles [28, 4], some of which simulate factors such as anisotropy and viscoelasticity [3]. Complex constitutive models have also been proposed specifically to simulate soft tissue [29], which are able to simulate complex skin properties such as hysteresis and preconditioning. The MT method [14] has also been used for such simulations. Due to its efficiency, the TLED algorithm has been used for various non-linear FE soft-tissue simulations of biological organs [30], including GPU implementations [15], resulting in large speed-ups.

Rather than modelling a small area, we focus on using detailed FE facial models. However, unlike current such models [8, 13], we also model detail such as skin layers using volumetric finite elements, which are necessary to simulate fine details like wrinkles [3], in order to produce both realistic-looking gross and fine-scale facial movement in a fully physics-based manner. Our simulation models are also optimised for GPU simulation. As well as computer graphics applications, due to the detail and accuracy of the simulations, our animation approach could also be useful in other fields, such as biomechanics and surgery.

3 Animation Process Overview

Figure 2 shows an overview of our entire animation process, which involves three major stages:

1. Creating the surface mesh for an object
2. Creating a suitable simulation model
3. Simulating and visualising the model over time

The surface mesh can be created using any 3D modelling software. Extending our previous work [31], our model creation system is then used to automatically discretise the volumes enclosed by this multi-surface mesh (see Figure 1) into a collection of nodes that are connected to form elements, and compute FE model parameters to produce a simulation model. These

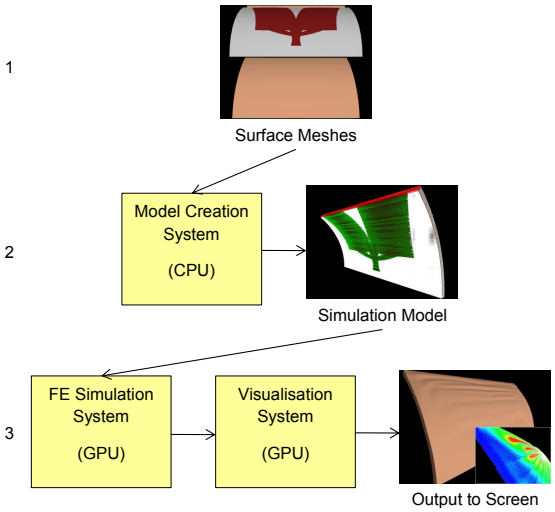

Figure 2: The stages of our animation process.

parameters include skin layers and element material composition, muscle fibre directions, and boundary conditions. We use non-conforming (voxel-based) hexahedral models with bound surface meshes due to model creation, performance and stability advantages [31]. The final stage of the animation process (the focus of this paper) involves simulating and visualising the models using our GPU-based FE simulation and visualisation system, either interactively or via batch mode.

4 Model Simulation

Our CUDA-based FE simulation system uses the non-linear TLED formulation of the FE method [30], which has various advantages for simulating soft tissue. Due to the requirement of a small but efficient simulation timestep, explicit time integration is highly suited for simulating the complex non-linear material behaviour of soft tissue under large deformations, as well as contact and sliding. Explicit methods are also inherently parallel, making them suitable for GPU implementation. To further increase simulation efficiency, the TL formulation enables some variables to be precomputed. Being dynamic, inertial and damping effects are also considered.

The procedure to compute a timestep using our simulation system is summarised by Algorithm 1. The iterations within each loop are in-

Algorithm 1 The process to compute a timestep for non-conforming elements.

```
for all elements, m do
    for all integration points do
        for all muscles overlapping m do
            Calculate active muscle stresses
        end for
        for all materials inside m do
            Calculate material stresses
        end for
        Calculate internal nodal forces
    end for
    Calculate hourglass forces
end for
for all nodes, n do
    if n not rigid then
        Calculate nodal displacements
    end if
end for
for all sliding nodes do
    Update nodal displacements
end for
```

dependent, and are therefore computed in parallel on the GPU. Our system has also been optimised to exploit the computational advantages of using the non-conforming models produced by our model creation process for GPU simulation [31], leading to performance increases of almost 2x compared with using a conforming hexahedral simulation mesh. The hexahedral elements of our simulation models are treated as reduced-integration 8-node hexahedra, which have a single central integration point at which element strains and stresses are evaluated. Factors such as thermal effects and fluid dynamics are not considered necessary for the level of accuracy we desire. The following subsections present details of our simulation approach.

4.1 The Total Lagrangian Explicit Dynamic Finite Element (TLED FE) Method

The full formulation of the dynamic TL FE formulation with explicit time integration has been presented in previous work [30]. As the TL formulation calculates deformation with respect to the initial object configuration, the Green-Lagrange strain tensor, ${}^t_0\mathbf{E}$ (at time t with respect to the initial configuration), is used, with the work-conjugate second Piola-Kirchhoff stress tensor, ${}^t_0\mathbf{S}$, to handle the large deformations and rotations that can occur. Using these, the principal of virtual work, the basis of the displacement-based FE method, can be defined:

$$\int_{{}^0V} {}^t_0\mathbf{S}\, {}^t_0\delta\mathbf{E}\, \mathrm{d}V = \int_{{}^0V} {}^t\delta\mathbf{u}\, {}^t\mathbf{f}^{(\mathbf{B})}\, \mathrm{d}\,{}^0V + \int_{{}^0S} {}^t\delta\mathbf{u}\, {}^t\mathbf{f}^{(\mathbf{S})}\, \mathrm{d}\,{}^0S \quad (1)$$

where ${}^t\mathbf{u}$ are displacements, ${}^t\mathbf{f}^{(\mathbf{B})}$ and ${}^t\mathbf{f}^{(\mathbf{S})}$ are external body and surface forces respectively, and ${}^0V$ and ${}^0S$ are the volume and surface area over which the forces are applied respectively.

By discretising an object into elements, and representing the variation of displacement over elements using shape functions, strain and stress vectors can be calculated. Relating element strains to nodal displacements, the strain-displacement matrix at element integration point i, ${}^t_0\mathbf{B}^{(\mathbf{i})}{}_L$, can be defined as:

$${}^t_0\mathbf{B}^{(\mathbf{i})}{}_L = \begin{pmatrix} {}^t_0\mathbf{B}^{(\mathbf{i})}{}_{L_1} & {}^t_0\mathbf{B}^{(\mathbf{i})}{}_{L_2} & \cdots & {}^t_0\mathbf{B}^{(\mathbf{i})}{}_{L_n} \end{pmatrix} \quad (2)$$

$${}^t_0\mathbf{B}^{(\mathbf{i})}{}_{L_a} = {}_0\mathbf{B}^{(\mathbf{i})}{}_{L0_a}\, {}^t_0\mathbf{X}^{(\mathbf{i})T} \quad (3)$$

where ${}_0\mathbf{B}^{(\mathbf{i})}{}_{L0}$ is the initial strain-displacement matrix, constructed from the shape function derivatives, n is the number of element nodes, and ${}^t_0\mathbf{X}^{(\mathbf{i})}$ is the deformation gradient. Evaluation of stresses at integration points is performed using the strains and relevant constitutive equation. As we use non-conforming models, the stress vector for an integration point is a weighted sum of the stress vectors calculated for each material used by the element.

Using the shape functions, and strain and stress vectors, and applying equation 1 for each degree of freedom, this principle of virtual work becomes following equation of motion:

$$\mathbf{M}\, {}^t\ddot{\mathbf{u}} + \mathbf{C}\, {}^t\dot{\mathbf{u}} + k({}^t\mathbf{u})\, {}^t\mathbf{u} = {}^t\mathbf{r} \quad (4)$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is the damping matrix, $k({}^t\mathbf{u})$ is the stiffness matrix, and ${}^t\mathbf{r}$ is the vector of external forces. Here, forces due to inertia and damping are separated from the external body forces. We use a lumped mass approximation with mass-proportional Rayleigh

damping. Internal node forces, ${}^t\mathbf{f}$, are calculated as:

$$ {}^t\mathbf{f} = k({}^t\mathbf{u})\,{}^t\mathbf{u} = \int_{{}^0V} {}^t_0\mathbf{B}_L^T\,{}^t_0\hat{\mathbf{s}}\,\mathrm{d}\,{}^0V \quad (5) $$

where ${}^t_0\hat{\mathbf{s}}$ is the second Piola-Kirchhoff stress vector. To advance simulations, the central difference time-integration method is used.

4.2 Hourglass Control

While using reduced-integration elements overcomes the volume-locking limitations that can occur when simulating incompressible material like soft tissue [15], a hourglass control technique is required to reduce the hourglass effects (zero-energy modes, whereby element deformations occur such that no strains, and hence no forces to resist the deformation, are produced) that can occur, particularly with large strains. We use a stiffness-based method, which adds artificial stiffness to elements to constrain the different hourglass modes based on element stiffness [32]. An element hourglass force matrix, ${}^t\mathbf{F}^{(\mathbf{m})}_{\mathbf{H}}$, for element m is calculated as:

$$ {}^t\mathbf{F}^{(\mathbf{m})}_{\mathbf{H}} = {}_0\mathbf{H}^{(\mathbf{m})}\,{}^t\mathbf{U}^{(\mathbf{m})} \quad (6) $$

$$ {}_0\mathbf{H}^{(\mathbf{m})} = \kappa \cdot k^{(m)}_{Max}\,{}_0\gamma^{(\mathbf{m})}\,{}_0\gamma^{(\mathbf{m})T} \quad (7) $$

where ${}_0\mathbf{H}^{(\mathbf{m})}$ is the hourglass matrix, ${}^t\mathbf{U}^{(\mathbf{m})}$ is a matrix of nodal displacements of element m, κ is a user-defined stiffness parameter, $k^{(m)}_{Max}$ is the maximum element stiffness, and ${}_0\gamma^{(\mathbf{m})}$ is the matrix of element hourglass shape vectors. Each row of the hourglass force matrix is an element hourglass force vector corresponding to an element node, and these vectors are added to the relevant nodal internal forces.

4.3 Muscle Contraction

To enable active muscle stresses to be generated during simulations, a muscle contraction model has been implemented. As muscles are transversely isotropic with preferred deformation in the fibre direction, as well as an active stress component, this direction is used to compute an additional passive stress component.

For a particular muscle, the additional stress produced by the muscle, ${}^t_0\mathbf{S}^{(\mathbf{i},\mathbf{mus})}$, for integration point i is computed by adding the active stress, ${}^t_0\mathbf{S}^{(\mathbf{i},\mathbf{act})}$, and additional passive stress, ${}^t_0\mathbf{S}^{(\mathbf{i},\mathbf{pas})}$:

$$ {}^t_0\mathbf{S}^{(\mathbf{i},\mathbf{mus})} = {}^t_0\mathbf{T}^{(\mathbf{i},\mathbf{pas})} + {}^t_0\mathbf{T}^{(\mathbf{i},\mathbf{act})} \quad (8) $$

$$ {}^t_0\mathbf{S}^{(\mathbf{i},\mathbf{act})} = {}^tJ\,{}^t\alpha\sigma^{(m)}_{act} f_{act}({}^t\lambda) \cdot {}^0\mathbf{d}^{(\mathbf{i},\mathbf{mus})}\,{}^0\mathbf{d}^{(\mathbf{i},\mathbf{mus})T} \quad (9) $$

$$ {}^t_0\mathbf{S}^{(\mathbf{i},\mathbf{pas})} = {}^tJ\sigma^{(m)}_{pas} f_{pas}({}^t\lambda) \cdot {}^0\mathbf{d}^{(\mathbf{i},\mathbf{mus})}\,{}^0\mathbf{d}^{(\mathbf{i},\mathbf{mus})T} \quad (10) $$

where J is the Jacobian of the deformation gradient, $\sigma^{(m)}_{act}$ and $\sigma^{(m)}_{pas}$ are the active and passive muscle stress references respectively, both weighted by the element overlap with the muscle, λ is the muscle fibre stretch ratio, and ${}^0\mathbf{d}^{(\mathbf{i},\mathbf{mus})}$ is the fibre direction. ${}^t\alpha \in [0,1]$ is the muscle contraction parameter, which is varied over time according to a function (e.g. linear or cosine) to simulate muscle contraction. $f_{act}({}^t\lambda)$ and $f_{pas}({}^t\lambda)$ are based on those used by Rörle et al. [18], which follow the tension-length properties observed by real muscle. These functions are defined as:

$$ f_{act}(\lambda) = \begin{cases} -\dfrac{25}{4\lambda^2_{opt}}\lambda^2 + \dfrac{25}{2\lambda_{opt}}\lambda - 5.25 \\ \qquad 0.6\lambda_{opt} \le \lambda \le 1.4\lambda_{opt} \\ 0 \quad \text{otherwise} \end{cases} \quad (11) $$

$$ f_{pas}(\lambda) = \begin{cases} 0 & \lambda \le 1 \\ 0.05(\exp^{6.6(\lambda-1)} - 1) & \\ & 1 < \lambda \le \lambda_{pas} \\ 2.18\lambda - 2.77 \quad \text{otherwise} & \end{cases} \quad (12) $$

where λ_{opt} is the optimal fibre stretch, which is normally similar to the stretch at which the passive stress increases sharply, λ_{pas}.

4.4 Boundary Conditions

To constrain models during simulations, boundary conditions must be set. With our system, it is possible to set nodes as rigid or sliding (bound by a surface). Rigid nodes are simply fixed with zero displacement throughout simulations, and can therefore be used to model muscle attachments on the skull, or the roots of retaining ligaments. Sliding nodes are used to model material that is normally attached to, but can slide over a rigid surface; for example, in reality, superficial

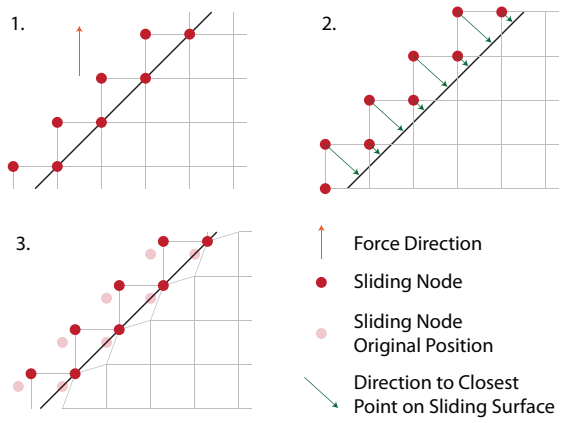

Figure 3: Sliding nodes moving along a sliding surface. The displacement of sliding nodes before (image 2) and after (image 3) the additional sliding displacement is shown. Note this is a 2D illustration of a 3D process.

facial soft-tissue layers are able to slide over the stiff deep layers and skull [33], although retaining ligaments normally restrict the separation of these layers. This sliding phenomenon has often been neglected in previous work, in which nodes on the skull are simply treated as rigid nodes [8, 13].

As we're using non-conforming models, sliding nodes won't necessarily lie on the surface they are bound by. To constrain such nodes to follow the shape of the restricting surface, they are forced to maintain a fixed distance away from this surface. Friction is not considered. As shown by Figure 3, the displacement of a sliding node, q, is updated after computation of a timestep to move this node to a position that is distance $d^{(q,s)}$. (the fixed distance) away from the sliding surface, s. This additional displacement, $\mathbf{u}^{(\mathbf{q},\mathbf{s})}$, is calculated as:

$$\mathbf{u}^{(\mathbf{q},\mathbf{s})} = -b^{(q,s)} \cdot (d^{(q,s)} + b^{(q,s)} \cdot \|(\mathbf{p}^{(\mathbf{q})} - \mathbf{c}^{(\mathbf{q},\mathbf{s})})\|) \cdot \frac{(\mathbf{p}^{(\mathbf{q})} - \mathbf{c}^{(\mathbf{q},\mathbf{s})})}{\|(\mathbf{p}^{(\mathbf{q})} - \mathbf{c}^{(\mathbf{q},\mathbf{s})})\|} \quad (13)$$

$$b^{(q,s)} = \begin{cases} 1 & (\mathbf{p}^{(\mathbf{q})} - \mathbf{c}^{(\mathbf{q},\mathbf{s})}) \cdot \mathbf{n}^{(\mathbf{q},\mathbf{s})} < 0 \\ -1 & \text{otherwise} \end{cases} \quad (14)$$

where $\mathbf{p}^{(\mathbf{q})}$ is the position of node q, $\mathbf{c}^{(\mathbf{q},\mathbf{s})}$ is the closest point on surface s to node q, and $\mathbf{n}^{(\mathbf{q},\mathbf{s})}$ is the surface normal at this point. To increase computational performance, a GPU-based semi-brute-force broad-phase collision detection algorithm with spatial subdivision has been implemented [34] to prune the number of polygons to be tested when finding the closest surface position for each sliding node.

Due to the fixed-distance restriction with non-conforming elements, our sliding procedure can limit the ability of elements to adapt to the shape of the surface as they move, particularly with lower-resolution models and around highly-curved areas. However, these constraints help to suitably restrict our models such that a higher timestep can be used. Alternatively, the penalty method has been used in related work [12], although the produced oscillatory movement can greatly decrease stability, and only penetration is constrained; for example, the movement of soft tissue away from the skull of a facial model would be unrestricted.

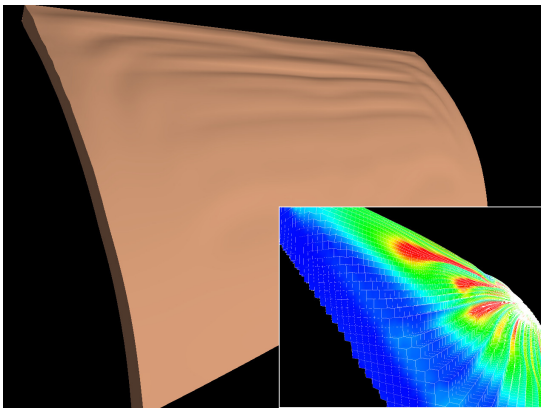

Figure 4: A soft-tissue-block model with forehead-like muscle structure under contraction of the frontalis. Insets show the stresses, where red indicates high, and blue indicates low stress.

5 Animation Examples

Figures 4, 5 and 6 show example animations using two models: a soft-tissue block model, and a forehead model. Tables 1 and 2 show the material properties that were used (based on those reported in literature [4]), and some model statistics. Muscle parameters were estimated based on literature [18] and from testing. Each muscle was assigned an active and passive stress reference of 5MPa, an optimal fibre stretch of 1 (rest

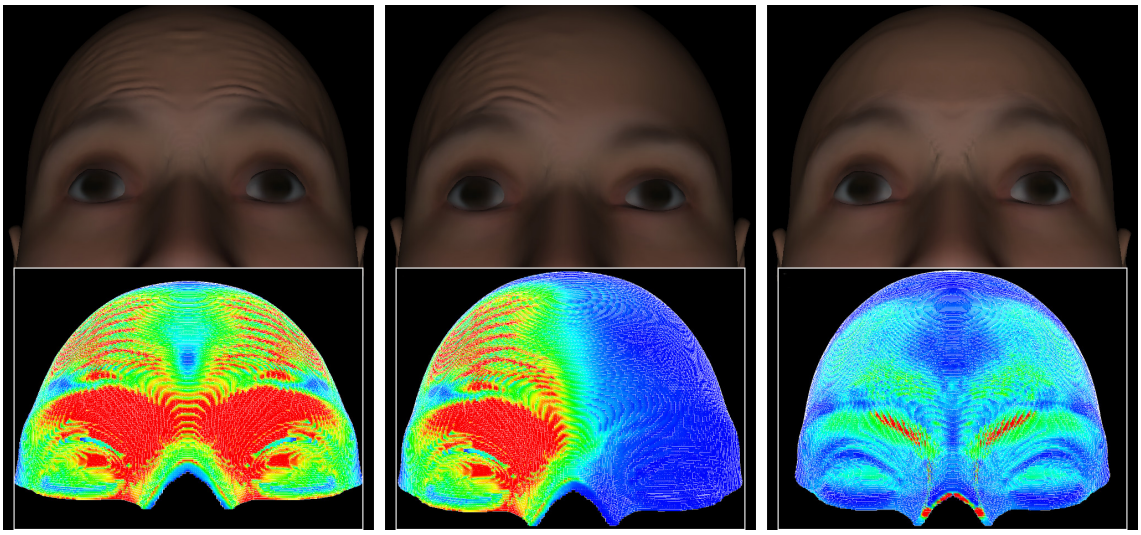

Figure 5: A forehead model under contractions of the frontalis. The example on the right models skin as a single layer (material).

Layer	ρ* ($\mathrm{kg/m^3}$)	E (MPa)	v	Depth (mm)
SC	11,000	48	0.49	0.02
D	11,000	0.0814	0.49	1.8
H	11,000	0.034	0.49	Remains
M	11,000	0.5	0.49	$\sim$1
T	11,000	24	0.49	$\sim$1

Table 1: The neo-Hookean material properties used for the animation examples.
Key: ρ: Density, E: Young's Modulus, v: Poisson Ratio, SC: Stratum Corneum, D: Dermis, H: Hypodermis, M: Muscle, T: Tendon
*Includes mass scaling.

Detail	Skin Block	Face
Nodes	485,400	629,178
Elements	414,868	503,530
Element Length (mm)	0.5	0.5

Table 2: Simulation model statistics

length), and λ_{pas} (see equation 12) was set to 1.2. As the deep layers are tough and fairly rigid, these were not modelled, and the superficial layers simply slide over the skull or bone surface. An estimated mass-proportional damping scale factor of 2 was used, and external forces that have little visual effect on the animations, such as gravity, were neglected.

Mass scaling was used to greatly increase the critical timestep, which is roughly proportional to the material density [15]. The actual density of each material ($\sim$1100$\mathrm{kg/m^3}$ [13]) was scaled by a factor of 10. Time scaling was also used, reducing the number of timesteps over which the animations were run. With each example, animations with linear variations of muscle contraction parameters between 0 and 0.75 over 500ms were produced by varying these parameters over 50ms, and playing back the animations 10x slower. As inertial effects are still relatively insignificant compared to muscle contraction forces when using such scaling parameters, the scaling techniques appeared to have little visual effect on the animations, particularly when played in real time. The timestep used for each animation was 5μs, and it took roughly 10.5ms to compute a single timestep for the forehead model on an NVIDIA GTX 680 GPU.

Due to the low thickness of the epidermal layer (stratum corneum) in relation to element size, the dermis dominates the outer layer of elements. When the material properties of these skin layers are combined for these elements, this results in a stiffness much lower than that of the epidermis alone. The epidermal layer was therefore assigned an unrealistically high stiffness to produce a large enough difference between the stiffness of the two outer layers of el-

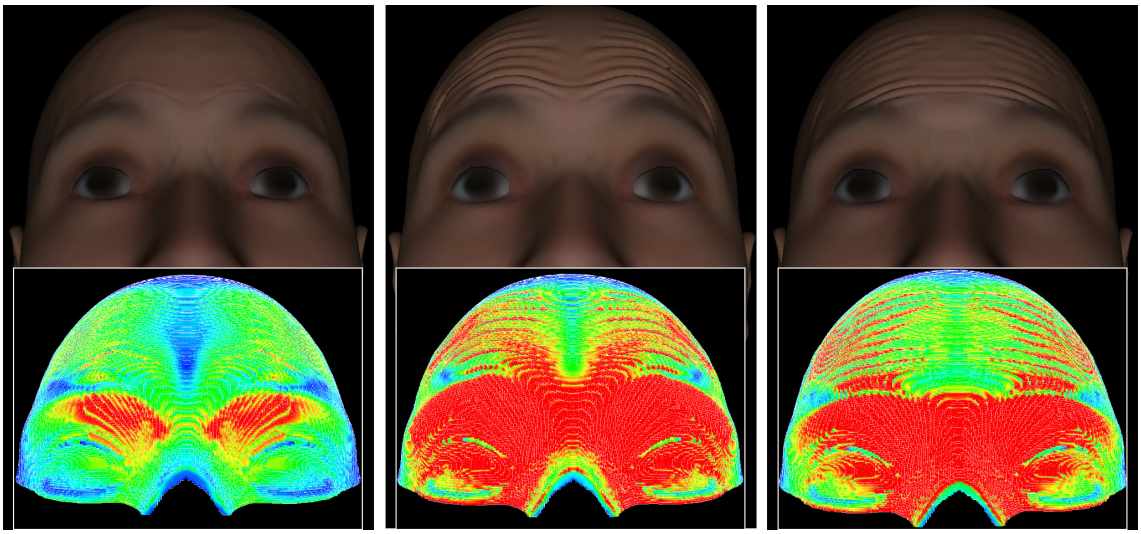

Figure 6: Variations of the forehead model under contractions of the frontalis using a lower epidermal stiffness of 24MPa, representing younger skin (left), using higher active and passive muscle stress references of 50MPa (middle), and where only the frontalis, and not the procerus and corrugator supercilli muscles, is modelled (right).

ements, which is necessary to simulate wrinkles [3]. This is unavoidable without using different-shaped elements that can capture the thickness of the epidermal layer.

The outer skin surface of the facial mesh was produced using FaceGen[1], while all other surfaces were manually created using 3D modelling tools. The simulation models, both of which have constant soft-tissue thickness, were then generated using our model creation system. As the frontalis has no skull attachment, the galea aponeurotica has been modelled on the forehead model (see Figure 1) to anchor this muscle and restrict soft-tissue movement towards the top of the head when it contracts. The region of overlap between these structures represents the smooth blend of fibres. Attachments of other muscles were defined by rigid nodes at anatomical attachment locations on the skull.

The animation examples demonstrate the ability of our approach to produce animation of realistic large and fine-scale soft-tissue behaviour, including wrinkles, on the face. When considering skin as a single layer (material), like with many current approaches [8, 13], no wrinkles are produced. As shown by Figure 6, by simply changing the material parameters of a model, it is possible to achieve different effects, such as the animation of different aged skin. Also, by using a different muscle structure, or a completely different model, different objects can be animated. This shows the flexibility of our approach to automatically deform different human or non-human soft-tissue structures, or even arbitrary multi-material soft bodies, producing animations of both large and fine details using an accurate physics-based technique.

However, there are some artefacts on the forehead model due to the inaccuracy of the surface mesh. The animations produced are highly dependent on the model and parameters; for example, the direction of movement, and wrinkling behaviour, such as number and size of wrinkles, depend highly on muscle shape, size and direction, thickness and parameters of skin layers and soft-tissue structures, and constrained nodes. It is therefore likely that much more realistic models and animations could be produced by using models generated from more accurate surface meshes (e.g. from CT and MRI data), rather than manually created surface meshes, as well as more accurate muscle parameters.

6 Conclusion

This work has presented a fully physically-based approach for efficiently producing

[1] http://www.facegen.com/

realistic-looking animations of facial movement, including animation of expressive wrinkles. Focussing on the forehead, this involves simulating non-conforming multi-layered hexahedral soft-tissue models using an optimised GPU-based non-linear TLED FE solver. The simulation process includes an anatomical muscle contraction model that generates active and transversely-isotropic passive muscle stresses, as well as procedures to constrain nodes. Animation examples have demonstrated the ability of our process to produce animation of realistic large and fine-scale soft-tissue behaviour, as well as the flexibility to deform different soft-body structures.

However, various improvements could be made to the simulation process. Multi-resolution models could be used, for example, to enable a higher resolution to be employed along the outer skin surface where wrinkles are produced. To model the thickness of the epidermal layer more accurately, 2D shell elements could also be attached to the outer skin surface. Future work will focus on experimenting with the models and parameters to produce more realistic animations of different-aged forehead movement, including incorporating more complex anisotropic viscoelastic material models into our system. The realism of such animations could also be increased by considering plasticity effects to model the increased appearance of Langer lines on older skin.

Acknowledgements

This research was supported by the EPSRC, and the presentation thereof by the Rabin Ezra Scholarship Trust.

References

[1] D. Terzopoulos and K. Waters. Physically-Based Facial Modeling, Analysis, and Animation. *J. Vis. Comput. Animat.*, 1(2):73–80, 1990.

[2] K. Kähler, J. Haber, and H.-P. Seidel. Geometry-based Muscle Modeling for Facial Animation. In *Proc. GI 2001*, pages 37–46.

[3] C. Flynn and B. A. O. McCormack. Finite element modelling of forearm skin wrinkling. *Skin Res. Technol.*, 14(3):261–269, 2008.

[4] O. Kuwazuru, J. Saothong, and N. Yoshikawa. Mechanical approach to aging and wrinkling of human facial skin based on the multistage buckling theory. *Med. Eng. & Phys.*, 30(4):516–522, 2008.

[5] R. M. Koch, S. H. M. Roth, M. H. Gross, A. P. Zimmermann, and H. F. Sailer. A Framework for Facial Surgery Simulation. In *Proc. SCCG 2002*, pages 33–42.

[6] S. Zachow, H.-C. Hege, and P. Deuflhard. Computer-Assisted Planning in Cranio-Maxillofacial Surgery. *J. Comp. Inf. Technol.*, 14(1):53–64, 2006.

[7] Y. Zhang, T. Sim, and C. L. Tan. Simulating Wrinkles in Facial Expressions on an Anatomy-Based Face. In *Proc. ICCS 2005*, pages 207–215.

[8] E. Sifakis, I. Neverov, and R. Fedkiw. Automatic Determination of Facial Muscle Activations from Sparse Motion Capture Marker Data. *ACM Trans. Graph.*, 24(3):417–425, 2005.

[9] M. Fratarcangeli. Physically Based Synthesis of Animatable Face Models. In *Proc. VRIPHYS 2005*, pages 32–39.

[10] C. Chen and E. C. Prakash. Physically based facial expression synthesizer with performance analysis and GPU-aided simulation. In *Proc. CyberGames 2006*, pages 171–176.

[11] M. Fratarcangeli. Position-based facial animation synthesis. *Comput. Animat. Virtual Worlds*, 23(3-4):457–466, 2012.

[12] K. Mithraratne, A. Hung, M. Sagar, and P. J. Hunter. An Efficient Heterogeneous Continuum Model to Simulate Active Contraction of Facial Soft Tissue Structures. In *Proc. WCB 2010*, pages 1024–1027.

[13] G. Barbarino, M. Jabareen, J. Trzewik, and E. Mazza. Physically Based Finite Element Model of the Face. In *Proc. ISBMS 2008*, pages 1–10.

[14] S. Xu, X. P. Liu, H. Zhang, and L. Hu. A Nonlinear Viscoelastic Tensor-Mass Visual Model for Surgery Simulation. *IEEE Trans. Instrum. Meas.*, 60(1):14–20, 2011.

[15] Z. A. Taylor, M. Cheng, , and S. Ourselin. High-Speed Nonlinear Finite Element Analysis for Surgical Simulation Using Graphics Processing Units. *IEEE Trans. Med. Imaging*, 27(5):650–663, 2008.

[16] Y. Lee, D. Terzopoulos, and K. Waters. Realistic Modeling for Facial Animation. In *Proc. SIGGRAPH '95*, pages 55–62.

[17] K. Waters. A muscle model for animation three-dimensional facial expression. In *Proc. SIGGRAPH '87*, pages 17–24.

[18] O. Röhrle and A. J. Pullan. Three-dimensional finite element modelling of muscle forces during mastication. *J. Biomech.*, 40(15):3363–3372, 2007.

[19] C. Y. Tang, G. Zhang, and C. P. Tsui. A 3D skeletal muscle model coupled with active contraction of muscle fibres and hyperelastic behaviour. *J. Biomech.*, 42:865–872, 2009.

[20] A. Duysak and J. J. Zhang. Fast Simulation of Facial Tissue Deformations Using Mass-Spring Chain Algorithm. In *TPCGUK '05*, pages 139–145.

[21] C. D. G. Reis, H. Bagatelo, and J. M. Martino. Real-time Simulation of Wrinkles. In *Proc. WSCG 2008*, pages 109–116.

[22] X. S. Yang and J. J. Zhang. Modelling and Animating Hand Wrinkles. In *Proc. ICCS 2005*, pages 199–206.

[23] C. Larboulette and M.-P. Cani. Real-Time Dynamic Wrinkles. In *Proc. CGI 2004*, pages 522–525.

[24] L. Dutreve, A. Meyer, and S. Bouakaz. Real-Time Dynamic Wrinkles of Face for Animated Skinned Mesh. In *Proc. ISVC 2009*, pages 25–34.

[25] M. Müller and N. Chentanez. Wrinkle Meshes. In *Proc. SCA 2010*, pages 85–92.

[26] B. Bickel, M. Botsch, R. Angst, W. Matusik, M. Otaduy, H. Pfister, and M. Gross. Multi-Scale Capture of Facial Geometry and Motion. *ACM Trans. Graph.*, 26(3), 2007.

[27] Y. Wu, P. Kalra, L. Moccozet, and N. Magnenat-Thalmann. Simulating wrinkles and skin aging. *Vis. Comput.*, 15(4):183–198, 1999.

[28] N. Magnenat-Thalmann, P. Kalra, J. Luc Lêvêque, R. Bazin, D. Batisse, and B. Querleux. A Computational Skin Model: Fold and Wrinkle Formation. *IEEE Trans. Inf. Technol. Biomed.*, 4(6):317–323, 2002.

[29] J. E. Bischoff. Reduced Parameter Formulation for Incorporating Fiber Level Viscoelasticity into Tissue Level Biomechanical Models. *Ann. Biomed. Eng.*, 34(7):1164–1172, 2006.

[30] K. Miller, G. Joldes, D. Lance, and A. Wittek. Total Lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation. *Commun. Numer. Methods Eng.*, 23(2):121–134, 2007.

[31] M. Warburton and S. Maddock. Creating Animatable Non-Conforming Hexahedral Finite Element Facial Soft-Tissue Models for GPU Simulation. In *Proc. WSCG 2012*, pages 317–325.

[32] G. R. Joldes, A. Wittek, and K. Miller. An efficient hourglass control implementation for the uniform strain hexahedron using the Total Lagrangian formulation. *Commun. Numer. Methods Eng.*, 24(11):1315–1323, 2008.

[33] T. Wu, K. Mithraratne, M. Sagar, and P. J. Hunter. Characterizing Facial Tissue Sliding Using Ultrasonography. In *Proc. WCB 2010*, pages 1566–1569.

[34] Q. Avril, V. Gouranton, and B. Arnaldi. Fast Collision Culling in Large-Scale Environments Using GPU Mapping Function. In *Proc. EGPGV 2012*, pages 71–80.