Multi-layer skin simulation with adaptive constraints

Pengbo Li

Master of Science

School of Computer Science

McGill University

Montreal,Quebec

2014-11-11

A thesis submitted to McGill University in partial fulfillment
of the requirements of the degree of Master of Science

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to my supervisor Prof. Kry for the continuous support of my research and writing of this thesis. I thank my fellow labmates in Computer Animation and Interaction Capture Lab for the stimulating discussions and all kind help I have received in the last one year, especially for translating my abstract to French by Charles Bouchard and polishing figures by Marc Jarvis. I am grateful to my parents Weimin Li and Aiqin Li, for giving birth to me at the first place and supporting me throughout my life. I would like to thank financial support from NSERC, CFI, and GRAND NCE.

ABSTRACT

We present an approach for physics based simulation of the wrinkling of multi-layer skin with heterogeneous material properties. Each layer of skin is simulated with an adaptive mesh, with the different layers coupled via constraints that only permit wrinkle deformation at wavelengths that match the physical properties of the multi-layer model. We use texture maps to define varying elasticity and thickness of the skin layers, and design our constraints as continuous functions, which we discretize at run time to match the changing adaptive mesh topology. In our examples, we use blend shapes to drive the bottom layer, and we present a variety of examples of simulations that demonstrate small wrinkles on top of larger wrinkles, which is a typical pattern seen on human skin. Finally, we show that our physics-based wrinkles can be used in the automatic creation of wrinkle maps, allowing the visual details of our high resolution simulations to be produced at real time speeds.

ABRÉGÉ

Nous présentons une approche pour la simulation basée sur la physique des rides formées dans une peau aux propriétés hétérogènes comportant plusieurs couches. Chaque couche de peau est simulée séparément par un maillage adaptatif. Les différentes couches sont ensuite reliées par des contraintes qui permettent uniquement aux rides dont les longueurs d'ondes du tissu sont identiques aux propriétés physiques du modéle de se déformer. Nous utilisons un mappage de textures pour définir l'élasticité et l'épaisseur des couches, puis définissons les contraintes à l'aide de fonctions continues que nous discrétisons selon les variations du maillage lors de la simulation. Dans nos exemples, nous utilisons une interpolation entre les sommets du maillage pour animer la couche du fond, et nous prsentons une variété d'exemples qui simulent des petites rides recouvrant des rides plus profondes - une composition courante de la peau humaine. Finalement, nous dmontrons que notre approche de simulation des rides basée sur la physique peut être utilisée pour la création automatique ou simultanée d'un mappage des rides, permettant la simulation en temps réel de rides détaillées à haute résolution.

TABLE OF CONTENTS

LIST OF TABLES

Table page

LIST OF FIGURES

CHAPTER 1
Introduction

Current methods for producing computer animation of wrinkled skin and clothing can be categorized into three groups. Artists can draw wrinkle patterns in texture maps to vary the surface normal during shading calculations [5, 17], and can design wrinkle curves to modify mesh geometry in an art-directed manner [10]. Alternatively, procedural approaches can produce mesh deformations that closely resemble the physical phenomenon [26]. In the third category are techniques that rely entirely on physics-based simulation to animate the surface mesh according to a set of assigned material properties [25, 9]. Regardless the technique used, wrinkles are important visual details used in the design of realistic and expressive characters, for clothing, faces, and hands.

In this paper we present a physics based approach for simulating wrinkles, and we focus specifically on skin as opposed to clothing. Whereas cloth can be simulated as a single thin shell that interacts with other shells or surfaces through contact, skin consists of layers of different materials that are *attached* to one another. Skin wrinkling occurs when a thin stiff layer buckles instead of compressing to accommodate deformation of the underlying volume; the elastic energy of the wrinkled shape is lower than when there is compression alone. With skin composed of multiple layers of thin material attached to a deformable volume, it is possible to have fine wrinkles on top of larger ones, as can be seen on human skin undergoing deformation. Figure 6–4 shows an example of this phenomenon. Physics based simulation automates the creation of these fine geometric details, thus the artist or animator need only design the underlying blend shapes and the material properties of the different skin layers. Because of the cost of simulating such fine details, we compute wrinkle maps from our multi later simulation, which permits physically plausible fine wrinkles to be added to interactive applications using a normal-map fragment shader.

We take inspiration from the embedded thin shells approach of Remillard and Kry [25], which has the benefit of using only a small number of degrees of freedom in comparison to a full volumetric simulation of a thin stiff layer on top of a softer volume. In our work, we similarly construct constraints that attach the different layers, but we also deal with multiple layers and modify the constraints to account for the distance between each layer. Because we drive the underlying volume with blend shapes, the multi-layer skin contains all the degrees of freedom. While this simplifies the simulation, there is still the major challenge of simulating the very high mesh resolutions necessary for fine wrinkles. Therefore, we employ the adaptive mesh technique of Narain et al. [23] to add degrees of freedom to the simulation only where they are needed. But adaptive meshes also require a careful modification of the constraints. Our solution is to define the constraints as continuous functions and use an adaptive sampling technique that ensures that the constraints remain satisfied after each remeshing. Furthermore, we address the simulation of heterogeneous materials, such as varying skin thickness and stiffness, both during construction of the constraints, and with a method for resampling material properties during remeshing.

Our contributions can be summarized as follows. We introduce a *multi-layer model* to approximate the layered structure of human skin. Our model takes into account *heterogeneous material properties* in the construction of the constraints between layers that allow the formation of wrinkles at frequencies predicted by the material properties. We present a method for *resampling constraints* such that they remain valid when different layers are adaptively remeshed to produce fine details. We similarly *resample material properties* for the simulation to accommodate remeshing of the different layers. Finally, we automate the creation of *physics-based wrinkle maps* for blend-shape driven simulation, which can be used to enrich real time animations with fine geometric details.

CHAPTER 2
Related work

In this chapter, we list and discuss the most relevant research in the context of our work. As this work builds on ideas in a broad range of study fields, a wide survey of the related work will be provided. These areas are divided into two major groups: cloth simulation and human skin simulation.

2.1 Cloth simulation

Physically based simulation is a critical tool used to produce computer animation that would be otherwise extremely difficult to create by artist. The problem of simulating deformation of multi-layer skin is not too different from cloth simulation, which is a problem which has received a vast amount of attention. Early work on modeling the dynamics of cloth used mass-spring systems [24], while other seminal work demonstrated the use of continuum methods and implicit stepping [3]. While these methods continue to be the widely-used, more recent research has addressed modeling issues [15, 35], collision handling [7, 16], strain limiting [29, 34], and resolving fine details such as wrinkles and folds [6, 9].

The popularity of mass spring systems is due to the ease of implementation and inexpensive computational costs. However, for higher fidelity simulations, continuum-based approaches are typically employed with finite element methods (FEM). Most of the existing FEM approaches are based on the geometrically exact thin shell formulation presented by Simo and Fox [27]. The characteristic folding and buckling behavior of cloth highly depends on bending properties, which plays a central role in the appearance of real textiles. The models of bending forces are typically characterized into two main approaches. One is to use crossover springs that extend the surface [24, 20]. The other one is to build a discrete hinge model and evaluate precisely the angle between adjacent mesh elements, thus creating plausible bending forces [6, 15, 32]. Furthermore, subdivision finite elements and a co-rotational

strain formulation have been used to produce smooth deformations, while likely avoiding the locking artifacts that can occur with linear shape functions [30]. For the simulation of single layer skin, Remillard and Kry [25] also use higher order shape functions for smoothness in the coarsely discretized underlying volume. However, the embedded thin shell uses linear shape function, and is coupled to the underlying volume using constraints designed to only permit deformation and the desired wrinkling frequency. This use of constraints is similar to the method of Bergou et al. [4], which provides a mechanism for a generating fine details in a high resolution simulation that track the general shape and motion of a previously computed coarse simulation. Our work uses a similar constraint based approach, and builds on the embedded thin shell technique, but results in a more realistic multi-layer human skin model.

Wrinkles and folds are critical visual features of deformable surface. The simulation of high resolution meshes is able to capture fine details, but it is computationally expensive. Therefore, a number of recent approaches prefer to add visual details during a post-processing step on the simulate result of a coarser model. Techniques to generate additional wrinkles on the reduced model can be divided into four main categories: procedural generation [2, 10], learning from high-resolution simulation data [33, 18], sampling from examples of real cloth [39], and physics simulation with simplified models [20, 26]. An alternative approach prefers to animate cloth through adaptive refinement, allowing local control of the resolution of meshes to conform to the complex shapes that arise during simulation [23, 37, 22]. This is beneficial in making a balanced trade-off between level of detail and computational cost, and thus we implement our technique using Narain's provided code for adaptive mesh simulation.

2.2 Human skin simulation

Human skin is composed of several layers, each with an unique property and function. Many existing computational skin models simulate several of its complex mechanical characteristics. Earlier work by Thalmann et al. [19] establishes layered skin models based on the physical structure of epidermis and dermis, and explores the formation of folds and wrinkles

on the skin. A three-layer numerical model of skin is presented by Flynn et al. [11] to simulate wrinkling, with plausible results comparing with results of *in vivo* wrinkling experiments performed on the volar forearm. Many multi-layer FE-based skin models are previously reported. Mark et al. [36] present a physics-based approach involving a multi-layer voxel-based model with an anatomical muscle contraction model for efficiently producing realistic-looking animations of facial movement. Yin et al. [38] focus on a specific example that fingertips often wrinkle after extended exposure to water. A full anatomical model is provided for analyzing the effect of a multilayered skin structure, numerically investigating relationship between material parameters and wrinkle wavelength and amplitude.

Surface wrinkling of a stiff layer resting on a soft substrate is widely observed, from dried fruit to mountain formation [14]. Human skin made up of epidermis, dermis and subcutaneous tissue with varying thickness and elasticity is one of the typical examples. Cerda et al. [8] builds a theory of thin elastic sheets offering insight into mechanism of wrinkling in human skin. We build on this model in order to identify the critical wrinkling frequency of different layers in order to build constraint functions that couple our adaptively meshed thin shell models. More recently, there have been significant efforts devoted to measuring the elastic properties and thickness of each layer [13, 12], which is critical to compute the wavelength of skin wrinkles.

CHAPTER 3
Multi-layer skin

Human skin is made up of multiple layers. Figure 3–1 shows an illustration of a cross section of human skin (by Don Bliss, National Cancer Institute). The epidermis consists of many sub-layers. Likewise, while the dermis is composed primarily of collagen and elastic fibers, it also contains a variety of other components, such as blood vessels, mechanoreceptors, and sweat glands.

Despite the variation of composition, we make the common assumption that a continuum mechanics model can be suitably defined for each layer, and furthermore we use a thin shell model. Figure 3–2 is simple diagram to illustrate the parts of mutli-layer model. Figure 6–3 provides a nice comparison between single-layer model and multi-layer model. In addition, we note that dermis and epidermis layers can be quite different across the body (e.g., eye lids in comparison to hands). Using the thin shell model, we can set the mechanical properties to correspond to the local properties, and we likewise accommodate the heterogeneous properties of the skin by allowing the parameters to vary across our layer models.

Note that we assume that the deformation of the subcutaneous layer is provided as a blend shape, thus we only simulate a two layer model based on the dermis and epidermis. We couple neighbor layers by using local constraints with certain frequency according to their mechanical properties. This follows a construction similar to the embedded thin shells work of Remillard and Kry [25], except that we use different model for predicting wrinkle frequencies for different layers which takes into account the mutli-layer model, as described in the following subsection.

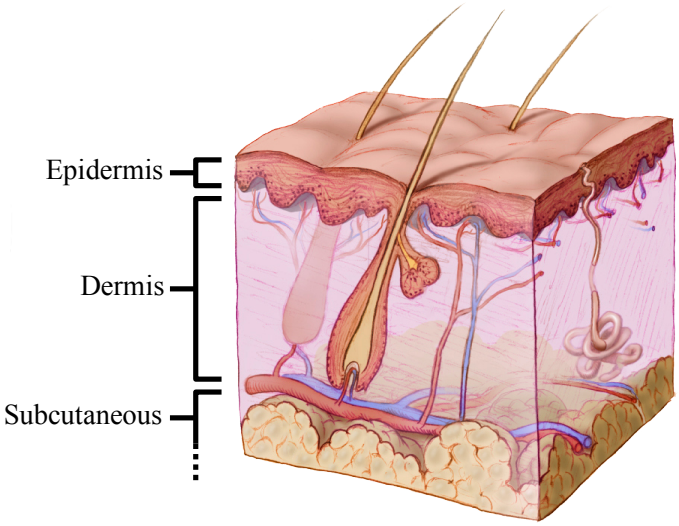

Figure 3–1: Illustration of skin layers cross-section.

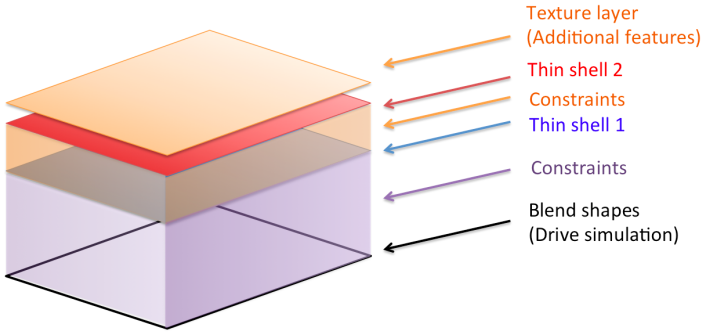

Figure 3–2: The diagram of multi-layer skin model. It clearly shows the composition of our model. From bottom to top, blend shapes drive simulation, thin shell 1 represents dermis to capture larger wrinkles, thin shell 2 represents epidermis to capture fine wrinkles, constraints couple neighbour layers together in way of winkle-frequency aware glue, only allowing wrinkling to happen at desired wavelength. In addition, we have a texture layer for adding additional visual features.

3.1 Critical wavelength model

The previous work on embedded thin shells applied a simple model that has been used to describe the mechanical behavior of a thin film resting on top of a soft elastic foundation [31], which tells us the critical wavelength of top layer can be computed by the skin's thickness and elastic properties. In other words, once these parameters have been identified, the wavelength of wrinkles can be predicted.

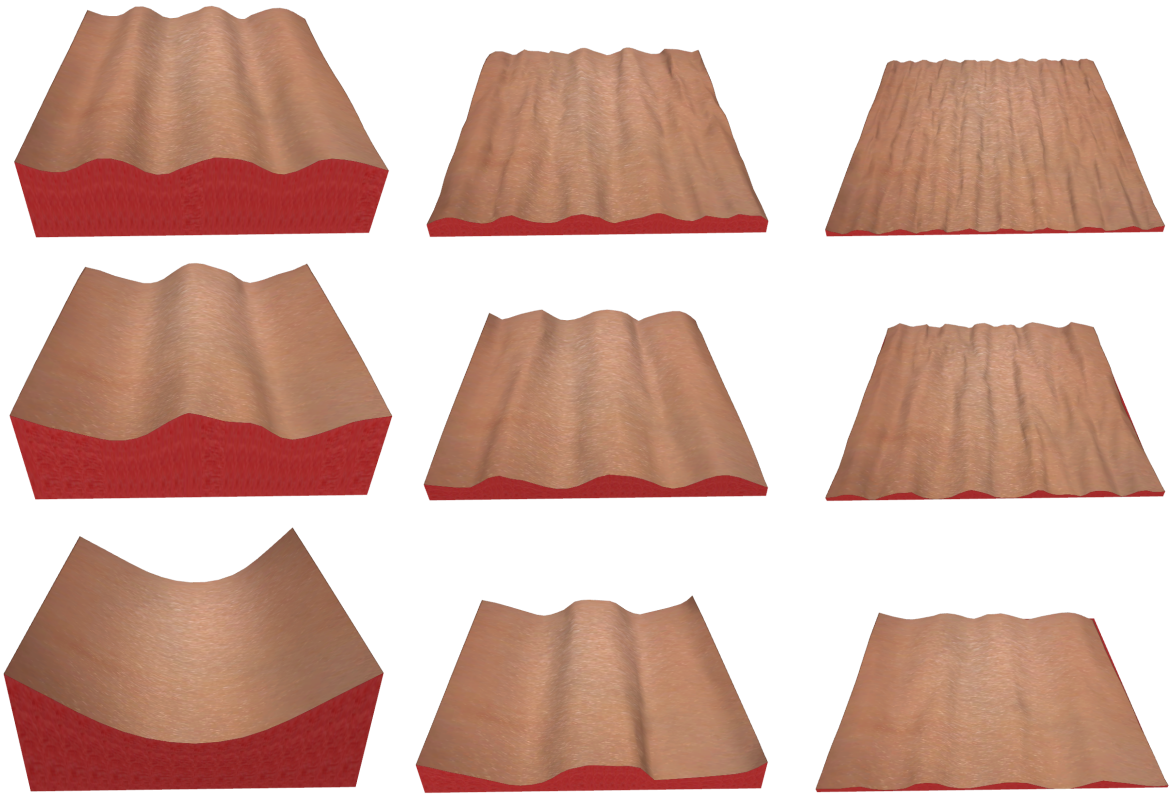

Figure 3–3: One layer examples with varying wavelength and thickness as a validation of Equation 3.1. From top to bottom, the Young's Modulus of skin layer increases by 64 times, resulting in the double of critical wavelength. From left to right, the thickness was quartered, halving the critical wavelength.

Human skin can likewise be approximated as a multi-layer composite material, where a thin and relatively stiff epidermis is attached to a soft dermis. In this case, the wrinkling frequencies can be described by using a simple physical model, though it differs slightly from the single layer model. A general form of the wrinkle periodicity is given by Cerda and Mahadevan et al. [8]. Given the thickness of the upper layer h_k and lower layer h_{k-1}, and the Young's modulus of each layer (E_k and E_{k-1} respectively), the critical wavelength is given by

$$\lambda_k \sim (h_k h_{k-1})^{1/2} \left(\frac{E_k}{E_{k-1}} \right)^{1/6}. \tag{3.1}$$

Once the critical wrinkling wavelength is known, we can determine a set of spatially distributed local constraints so that our multi-layer thin shell model produces wrinkles with the desired wavelengths. Note that when the skin layers have varying elasticity and thickness,

this approximation of the critical wavelength will be local and we must take this into consideration when constructing constraints that allow for the appropriate variation of wrinkle wavelengths.

CHAPTER 4
Constraints

Using a scheme similar to embedded thin shells, we build constraints to couple the thin shell simulations in different layers. A set of constraints must be defined between each pair of adjacent layers. The constraints are local and overlap in a manner that ensures that the higher layer will follow the lower layer at spatial frequencies lower than its critical wrinkling wavelength. We couple the bottom layer with the blend shape through position constraints, which constrains this first layer shell to the embedded positions in the blend shape.

Using $k \in \{1, 2, \cdots\}$ as a layer index, consider an upper layer defined by positions $\mathbf{x}_{k-1} \in \mathbb{R}^m$ and a lower layer with positions $\mathbf{x}_k \in \mathbb{R}^n$. In general, the number of vertices in the upper layer will be higher than the lower layer, and the positions will correspond to a different discretization of material space. We define constraints which include a normal-based term to account for the layer thickness,

$$H_k \mathbf{x}_k = H_{k-1}(\mathbf{x}_{k-1} + t_k \mathbf{n}_{k-1}), \tag{4.1}$$

where $\mathbf{n}_{k-1}$ provide the lower layer vertex normals, t_k is the layer thickness, and $H_k : \mathbb{R}^n \to \mathbb{R}^c$ and $H_{k-1} : \mathbb{R}^m \to \mathbb{R}^c$ form a sparse constraint matrix across all degrees of freedom. The weighted combination of vertex positions defined by the constraints must be computed to address different layer mesh discretizations that arise in using adaptive meshes for the layers and we address this below in Sections 4.2 and 4.2.1. For heterogeneous materials, the thickness of a layer is non-constant, and we replace the scalar t_k with a diagonal matrix to account for the thickness at different vertices. The thickness must be sampled from a property map based on the current layer discretizations (see Section 5.1), and similarly the normals must be computed (we use an area weighted average of adjacent face normals).

Below, we first address the construction of the constraints, and then describe our approach to consistently sampling the constraint function when the adaptive layer mesh changes.

4.1 Construction by clustering

Simulating skin layers with heterogeneous material properties requires that we establish non-uniform constraints according to the critical wavelength of each vertex. Each vertex can be assigned a wavelength based on its material space position and a material space property map. By using the critical wavelength, we produce a weighted clustering where the clusters have different sizes. Small clusters form in regions of low wavelength, and large clusters form in regions of high wavelength.

We use a greedy approach to create clusters. We approximate the distance of a vertex to a cluster center by the shortest path following mesh edges, similar to Remillard and Kry [25], except that we divide the mesh edge lengths by the average critical wavelength of the adjacent vertices. This effectively encourages regions where we expect high frequency wrinkles to form smaller clusters because the edge traversal costs in these regions will be larger. We are also able subdivide to get a fine mesh or use fast marching method, which are both useful for computing more accurate geodesic distances.

We perform the clustering on a fine meshing of the material space so as to best approximate the variation of wrinkle wavelengths across the continuous regions of material space. We start by choosing a random vertex as the first center, then compute the shortest weighted edge traversal distance to all other vertices using Dijkstra's algorithm. Given the critical wavelength at the randomly selected vertex, we do not want to include another vertex cluster within radius r equal to $2/\pi$ the critical wavelength (see Remillard and Kry [25] but here r is not constant) Then, among the vertices that are sufficiently far from other existing cluster centers, we choose the farthest vertex as the next center. The process repeats until all vertices are within half a wavelength of a cluster center.

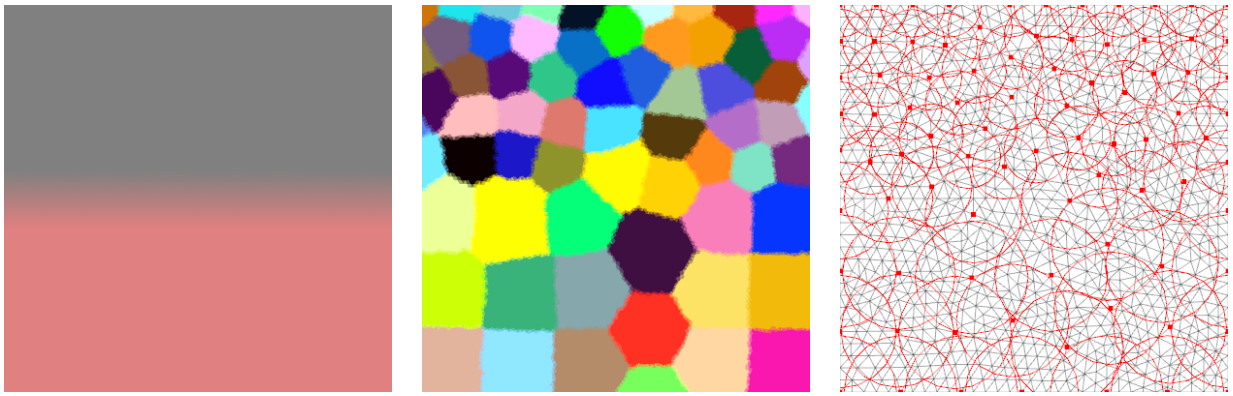

Figure 4–1: Constraint construction visualized in material space. Left shows the varying property map of the skin. Middle shows the clusters formed based on the expected wrinkle properties with smaller areas higher in material space to accommodate smaller wrinkles. Right shows the continuous truncated Gaussian supports overlaid with an adaptive meshing of material space.

Figure 4–2: These are examples for running our clustering algorithm on non-flat meshes. From left to right, they are curved surface, hand model and human face.

Our clustering algorithm applies to both flat and non-flat meshes (see Figure 4–1 and Figure 4–2). Regarding flat meshes, we can perform clustering on a fine meshing of the material space instead of world space so as to construct a continuous constraints function field across the regions of material space based on the Euclidean distances between the vertex and the center of cluster, allowing to accurately sample weights of vertices after remeshing. For non-flat meshes with high distortion in material space, Euclidean distances between vertices become meaningless. We will discuss this in the Section 6.2. An example of the clustering on a flat mesh can be seen in Figure 4–1. Given uniform elastic properties, the

brightness of red color indicates the thickness of skin layer. The darker region is thinner than the bright region, resulting in a smaller critical wavelength and thus smaller clusters and a smaller radius of the continuous constraint function. The extent of the truncated Gaussian constraint functions can be seen as circles in the right most image of the figure.

4.2 Area weighted constraint sampling

Because the layers do not have a fixed meshing, we must sample the continuous functions to create constraints for a given discretization of the layers.

Within a cluster of vertices belonging to the upper layer, each constraint requires the weighted position average of those vertices to match a corresponding weighted position average of vertices in the lower layer. For a given constraint function and a given material space mesh for each layer, the set of vertices influenced by the constraint is defined by the radius of the continuous constraint function, which is equal to half the critical wavelength. The constraint in Equation 4.1 with the normal term omitted can be written in an expanded form as

$$\sum_{i \in C_k} \omega_i x_i^k = \sum_{j \in C_{k-1}} \omega_j x_j^{k-1}, \tag{4.2}$$

where C_k and C_{k-1} are the sets of vertices involved at each layer, and ω_i are weights given by a truncated Gaussian function,

$$\omega_i = \begin{cases} \sigma_c a_i e^{-\frac{d_i^2}{2\sigma^2}} & \text{if } d_i < 2\sigma, \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

Here σ_c normalizes the sum of weights to one ensuring an affine combination, a_i is the area corresponding to vertex i, which equals the average of one third the area of the adjacent faces, $\sigma = \frac{1}{2}r$, and d_i is the distance from vertex i to the constraint's center. Note again that the truncation at distance 2σ determines the vertex presence in the cluster, namely, it is within the radius of constraint center.

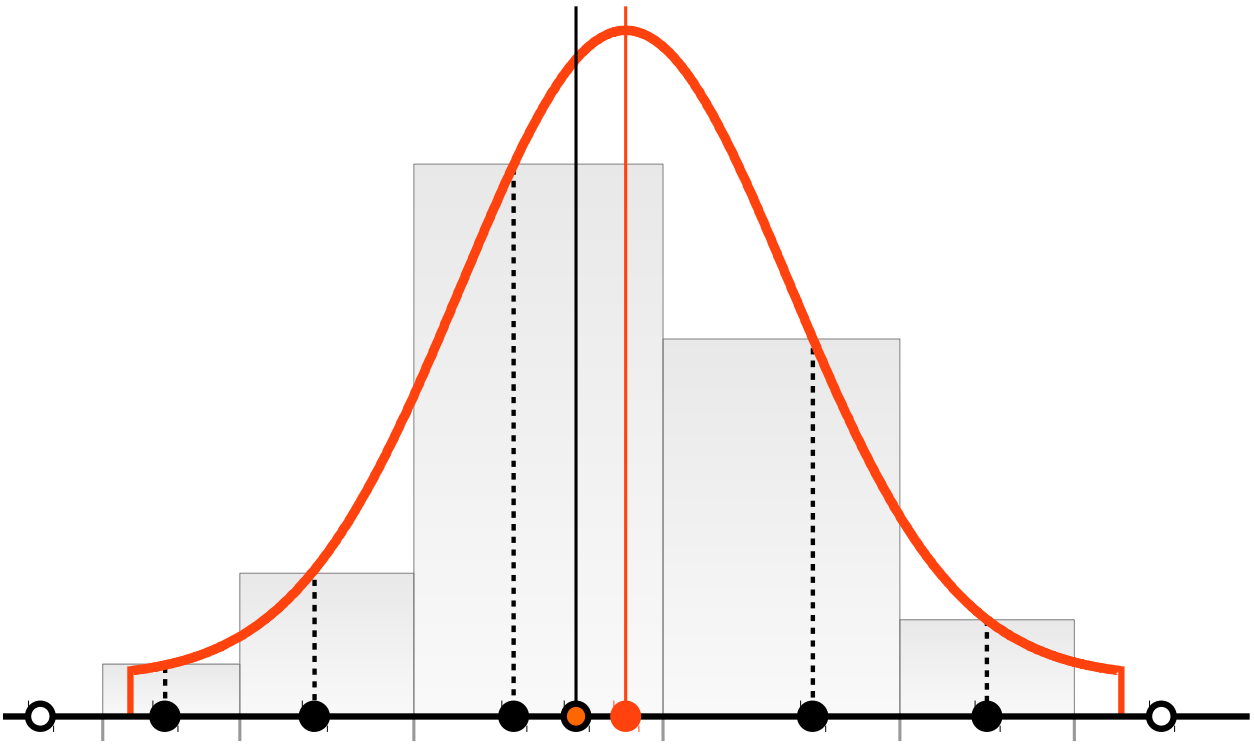

Figure 4–3: This 2D illustration explains the first step in our weighted sampling of continuous constraint functions. The constraint function is a truncated gaussian, and each sample is weighted differently given the distance to its neighbours (small ticks on the horizontal axis denote the midpoints between samples). Note that we must renormalize these weighted samples to produce an affine combination, and without an additional modification, the affine combination of the sample locations does not correspond to the centre of the constraint (compare the orange and black vertical lines).

Figure 4–3 shows a 2D representation of the constraint sampling. Using area weights is important because the adaptive mesh may create scenarios where there are small triangles on one side of a constraint and larger ones on the other side. But even with this area weighted sampling, the constraint's affine combination of material point sample locations will not generally correspond to the center of the constraint function. We address this with an adjustment as described in the next section.

4.2.1 Constraint Adjustment

The weights of vertices in a constraint are determined by its area and distance to the center of the cluster, and the constraint coefficients for vertices are resampled each time a layer is remeshed. Anisotropic adaptive remeshing leads to changes in the sampled constraint value, and produces unstable simulations if these changes are not addressed. Therefore, we propose a method to adjust the weights of each vertex such that the weighted average positions of the material space coordinates remains at the exact center of its constraint after adaptive remeshing.

Given a vertex with material space coordinates $u^i = (u^i_x, u^i_y)$, we compute an adjusted weight ω^*_i as

$$\omega^*_i = \omega_i + \alpha_1 m_1(u^i_x) + \alpha_2 m_2(u^i_y), \tag{4.4}$$

where α_1 and α_2 are coefficients of functions that alter the weight depending on the material space location. We use simple linear functions for m_1 and m_2, which evaluate zero at the cluster center,

$$m_1(u^i_x) = u^i_x - u^*_{c_x}, \tag{4.5}$$

$$m_2(u^i_y) = u^i_y - u^*_{c_y}. \tag{4.6}$$

where u^*_c is the coordinates of the center of the cluster. To solve for α_1 and α_2, we note that the sum of material space coordinates with adjusted weights should equal the cluster center, that is,

$$\sum_{i \in C} (\omega_i + \alpha_1 m_1(u^i_x) + \alpha_2 m_2(u^i_y)) u^i = u^*_c. \tag{4.7}$$

Bringing the sum of old weighted positions to the right hand side, we can create and solve a 2×2 matrix system as $A\alpha = b$ where

$$A = \sum_{i \in C} [m_1(u^i_x) \;\; m_2(u^i_y)]^T u^i, \quad b = u^*_c - \sum_{i \in C} \omega_i u^i. \tag{4.8}$$

While the system solve is trivial, there is a cost to assembling A and b proportional to the number of vertices in the cluster. However, the cost is small relative to the initial area weighted constraint sampling and normalization.

CHAPTER 5
Simulation

We start this chapter with briefly describing our implementation process, which shows the heritage from previous research and new modifications we made. We then have three subsections to describe how we sample property parameters, design blend shapes and automate the generation of wrinkle maps in details.

5.1 Implementation overview

Our implementation is based on the ARCSim code developed by Narain et al. [23]. We naturally inherit many methods from their work for deformable surface simulation. We use the piecewise linear model described by Wang et al. [35] as the constitutive model, the co-rotational finite element method [21] for in-plane stretching, and discrete hinge model [6] for bending energy. In addition, a bounding volume hierarchy [28] is used to detect collisions and non-rigid impact zones [16] is used to resolve collisions. The linear system is solved using direct solver in the TAUCS library. [1]

At each simulation step, we add a sampling step after the adaptive remeshing to resample the material property and weights of each vertex with respect to the new discretization. Furthermore, we modify the solver step as follows. There is a step to build the constraint matrix, and we've changed the framework to solve the constrained system with an implicit Euler method. In addition, we must update the position of vertices in blend shapes at the beginning of each frame.

We employed Lagrangian constrained mechanics to have implemented both a statics position based solver and dynamic velocity based solver. We formalized position based

[1] http://www.tau.ac.il/∼stoledo/taucs/

solver as following

$$\begin{bmatrix} \frac{\partial^2 E_x}{\partial x^2} \mid_{x_k} & \nabla g(x_k)^T \\ \nabla g(x_k) & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \lambda \end{bmatrix} = \begin{bmatrix} -F(x_k) \\ g(x_k) \end{bmatrix}, \tag{5.1}$$

where $F(x_k)$ is all forces acting on the shell at configuration x_k and $g(x_k)$ is generic expression for constraints function, in our cases, which is formulated as Equation 4.1. This solver computes a quasi-static equilibrium of each layer from the configuration of lower layer, without considering the mass of skin and velocity damping. Bulking in this case happens suddenly once the skin reaches the critical strain. In contrast, taking inertia into consideration, we implemented the velocity based solver as

$$\begin{bmatrix} M - h^2 J_F(x_k + h\dot{x_k}) & h\nabla g(x_k)^T \\ h\nabla g(x_k) & 0 \end{bmatrix} \begin{bmatrix} \Delta\dot{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} hf(x_k) \\ -g(x_k + h\dot{x_k}) \end{bmatrix}, \tag{5.2}$$

where M is mass physical mass matrix, $J_F(x_k + h\dot{x_k})$ is force Jacobian of forces evaluated at $x_k + h\dot{x_k}$ and $f(x_k)$ is external forces. We solve this system for $(\Delta\dot{x}, \lambda)$ to forwardly integrate positions. After choosing sufficient damping, we can observe smooth transitions between energy minima. The comparison between position based solver and velocity based solver can be seen in the supplementary video.

5.2 Property map sampling

We assign a texture map to the material space to allow varying layer material properties across the surface. Properties are encoded in the different color channels of the texture map, allowing an artist to simply paint varying thickness or stiffness. The red channel represents the critical wavelength, the green channel is for Young's modulus, and the blue channel indicates thickness. For each model, having a reference material parameter with default thickness and Young's modulus, we convert the sampled RGB values to corresponding relative values, then scale the reference material to real value. Our two-layer skin model has a reference material of Young's modulus $E = 4$ MPa and thickness $h = 1.3$ mm for the first

layer, and a reference material with $E = 600$ MPa and $h = 0.12$ mm for the second layer. For the soft substructure we use $E = 0.5$ MPa and $h = 15$ mm.

We sample RGB color for each vertex with respect to the texture coordinates and the bilinear interpolation among related pixels is naturally applied, then we are able to assign average values of related vertices to faces and edges. Considering the applicable range of 8-bit color, thickness and Young's modulus are both stored as relative value. Each model has reference material parameters with default thickness and Young's modulus. Once we sampled relative value, we scale the parameters of reference material to real value for physical simulation.

When the a skin layer is adaptively remeshed, we set the elastic properties of the new discretization by sampling the property map. For instance, a new edge will have its bending stiffness computed from values obtained from the property map at the midpoint of the edge. Note that this assumes that heterogeneous properties in the material map are smooth and effectively constant at the resolution of mesh. This is discussed in Section 6.2.

5.3 Blend shapes design

Blend shapes is a technique that a series of deformed version of a mesh is stored, then the vertices are interpolated between these stored positions of key poses as animation is approaching. Because artists are able to define the individual positions of the vertices without any constraints, blend shapes is widely used to animate cloth and skin for its flexibility of making desired deformations.

We design blend shapes to approximate external influence or muscle activations as the bottom layer to drive physical simulation. The positions of vertices are updated by changing the weights of linear interpolation between initial shape and target shapes. In addition, we are able to create a complicated shapes through blending multiple simple shapes. Figure 5–1 shows examples we created for human skin simulation using deformation tools in Maya.

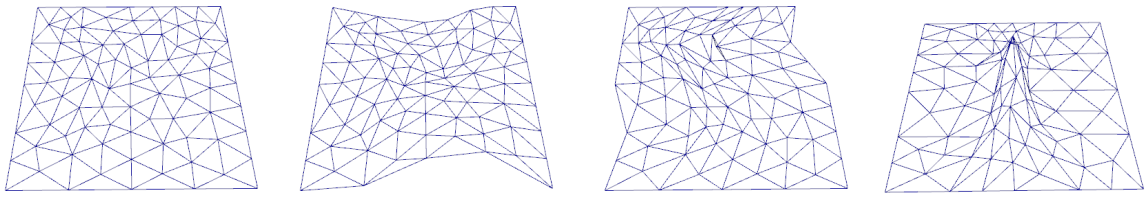

Figure 5–1: This figure shows the blend shapes of our human skin example (see Fig 6–4). The most left is rest shape, then from left to right, they are full blend shapes of motion patterns during compressing, twisting and pinching.

5.4 Wrinkle map construction

The simulation of very fine wrinkles with our multi-layer model can be quite costly, but we need not compute the shapes every time if we are only using a small number of blend shapes. Instead, we see our model as an interesting means for automating the creation of wrinkle maps. The creation of a wrinkle map is straightforward as we only need to drawing the material space mesh with the vertex color set to the normal direction. We map each components of the normal to the $[0, 1]$ interval and store the result in an 8 bit texture map (we find that quantization errors are not noticeable in our application). Figure 5–2 shows an example of the wrinkle maps produced.

We create wrinkle maps for a few key frames that capture the different wrinkling modes, and then use a texture shader to interpolate the normal maps based on the parameter of the blend shape model. This produces plausible results that resemble those of the physics-based simulation, as can be seen in an example of a two-layer skin model in the supplementary video. While we demonstrate this process for a blend shape model, we note that it can also be combined with skinning methods where the angle of the joint would provide the interpolation parameter. Besides, in the future the idea of coupling physics simulation to texture maps is able to have wide applicability, including rendering muscles and tendons, color changes of blood flow [1], bruises and cuts.

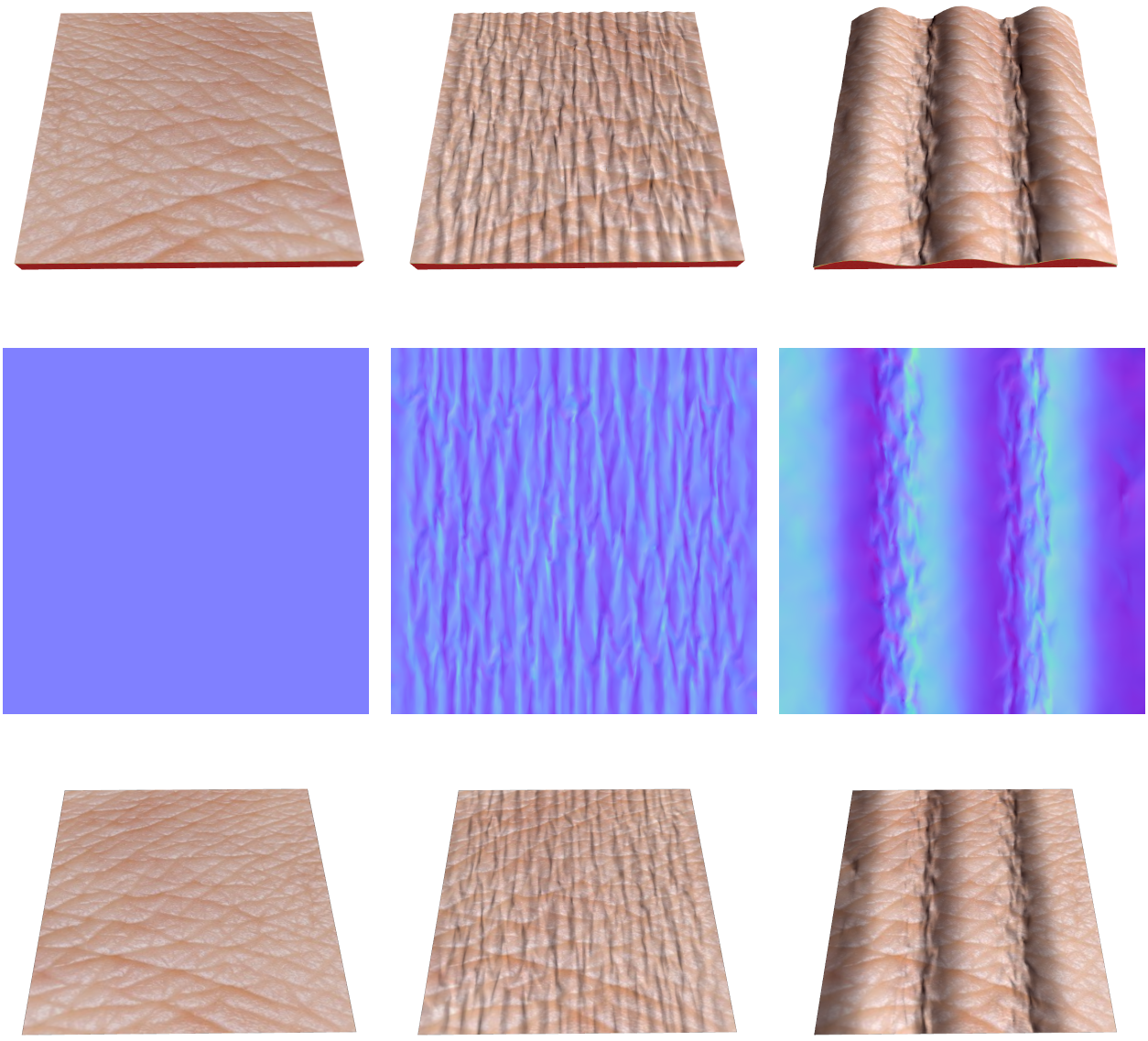

Figure 5–2: Examples of skin rendered with a wrinkle-map shader computed from a two layer skin simulation. Top and middle row show examples of wrinkle maps constructed from simulation of different blend shape deformations using a two layer skin model. At the bottom row, left to right shows the interpolation points in our wrinkle-map model, with left being the skin for the rest blend shape, middle at a blend shape of 0.5, and right the full blend shape.

CHAPTER 6
Discussion

In this chapter, we present a variety of examples and collect simulation time in Table 6.1, as an important aspect to evaluate the practicability. We will describe these examples and make comparisons helping to show the success of our technique. Finally we also discuss the limitations to provide a guideline for future exploration.

6.1 Results

Table 6.1 shows a breakdown of computation time for the different examples in this paper. As we use direct solver instead of minres iterative solver, the simulation is relatively time-consuming. But we do not need to worry about preconditioning to facilitate the solver progression. Note that we set the minimum allowable triangle size of adaptive remeshing very small, which produces a large number of degree of freedom and slower solves.

According to the model of Equation 3.1, we compute critical wavelength and construct proper constraints on a one layer model. Figure 3–3 shows that our simulation results well match the prediction that the changes of wavelength is in accordance with the varying thickness and stiffness.

Figure 6–3 is a good examples that helps justify our multi-layer concept. We can easily observe small wrinkles standing on larger wrinkles in the two-layer model, while the single layer model does not produce this effect. Attaching this second layer increases the realism of skin simulation. Meanwhile, the figures of meshes in material space shows the superiority of adaptive meshing, which focuses the resolution on the regions where small wrinkles appear. The example simulation accounting for heterogeneous material properties can be seen in Figure 6–1.

Aiming to produce realistic animation, we design three blend shape models to mimic typical human skin deformations. This includes compressing, twisting, and a pinch while

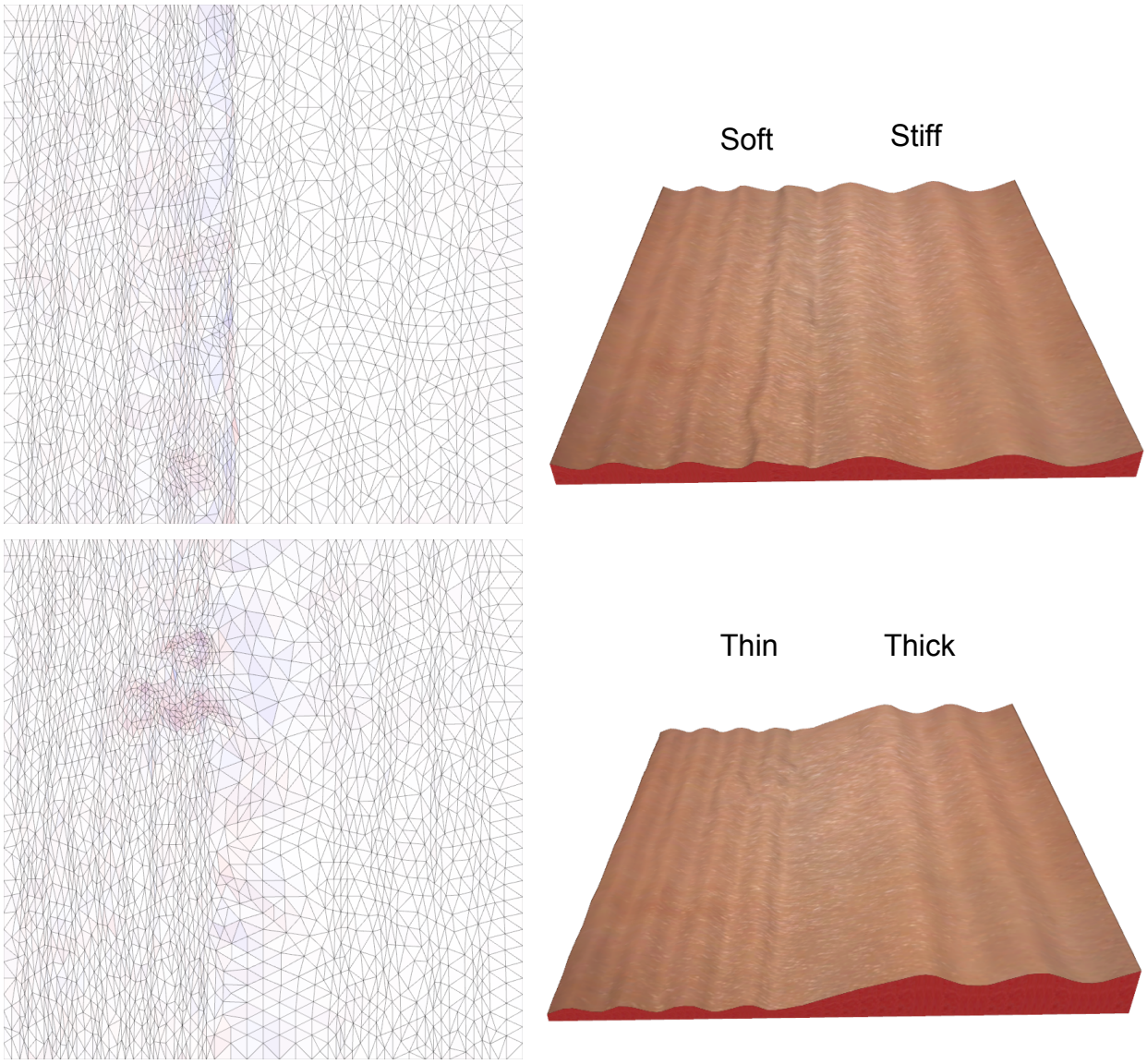

Figure 6–1: Examples of one and two layer skin models under compression with different heterogeneous material properties. From top to bottom, single layer with varying stiffness, single layer with varying thickness, attaching second layer on one layer model with varying stiffness.

pulling up. Figure 6–4 shows the plausible results for wrinkled human skin. We use a diffuse skin texture to render this example in order to provide increasing visual realism (see Figure 6–2).

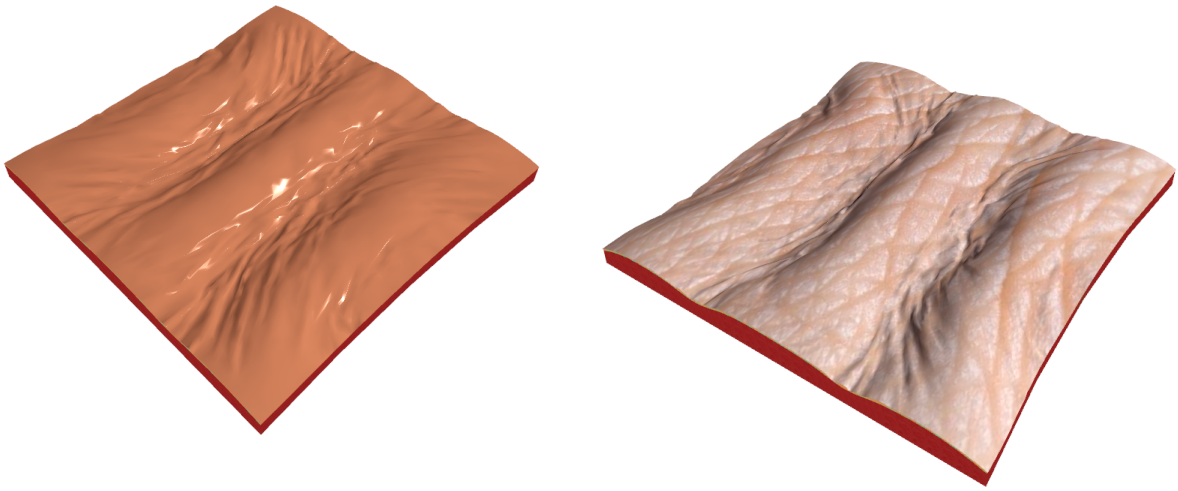

Figure 6–2: We have tried typical Phong shading (left) and texture shading (right) to render our human skin examples. As we can see, the highlights on the Phong shading model give us good clues to tell geometrical wrinkling deformation. But the model with skin texture is more realistic.

				Avg. time/frame			
Name	N_c	Avg. N_v	t_c	t_I	t_R	t_S	Total
Figure 6–3	436	7693	1.05	107.4	0.242	1.44	109.082
Figure 6–1 middle	189	3104	0.08	76.04	0.051	0.80	76.891
Figure 6–1 top	139	2696	0.05	71.81	0.056	0.63	72.496
Figure 6–4 top	421	11529	1.01	269.8	0.651	3.42	273.871
Figure 6–4 middle	421	6770	1.01	85.57	0.191	1.06	86.821
Figure 6–4 bottom	401	5294	0.98	81.45	0.218	1.21	82.878

Table 6–1: Performance measurements for our examples. The number of constraints is represented by N_c, N_v is the number of vertices, t_c is the time of constructing constraints, t_I is the time of integration for solving and updating system, t_R is the time for remeshing and t_S is the time for sampling, including the time of material property sampling and constrains adjustment. All time data were collected on a machine with 2.4 GHz Intel Core i7 CPU. All times are in seconds.

6.2 Limitations

Sampling of material properties assumes that they only change smoothly. Note that it would be an interesting challenge to compute appropriate material properties for a very coarse mesh such that the mechanical behavior matched the finer version. Because our base

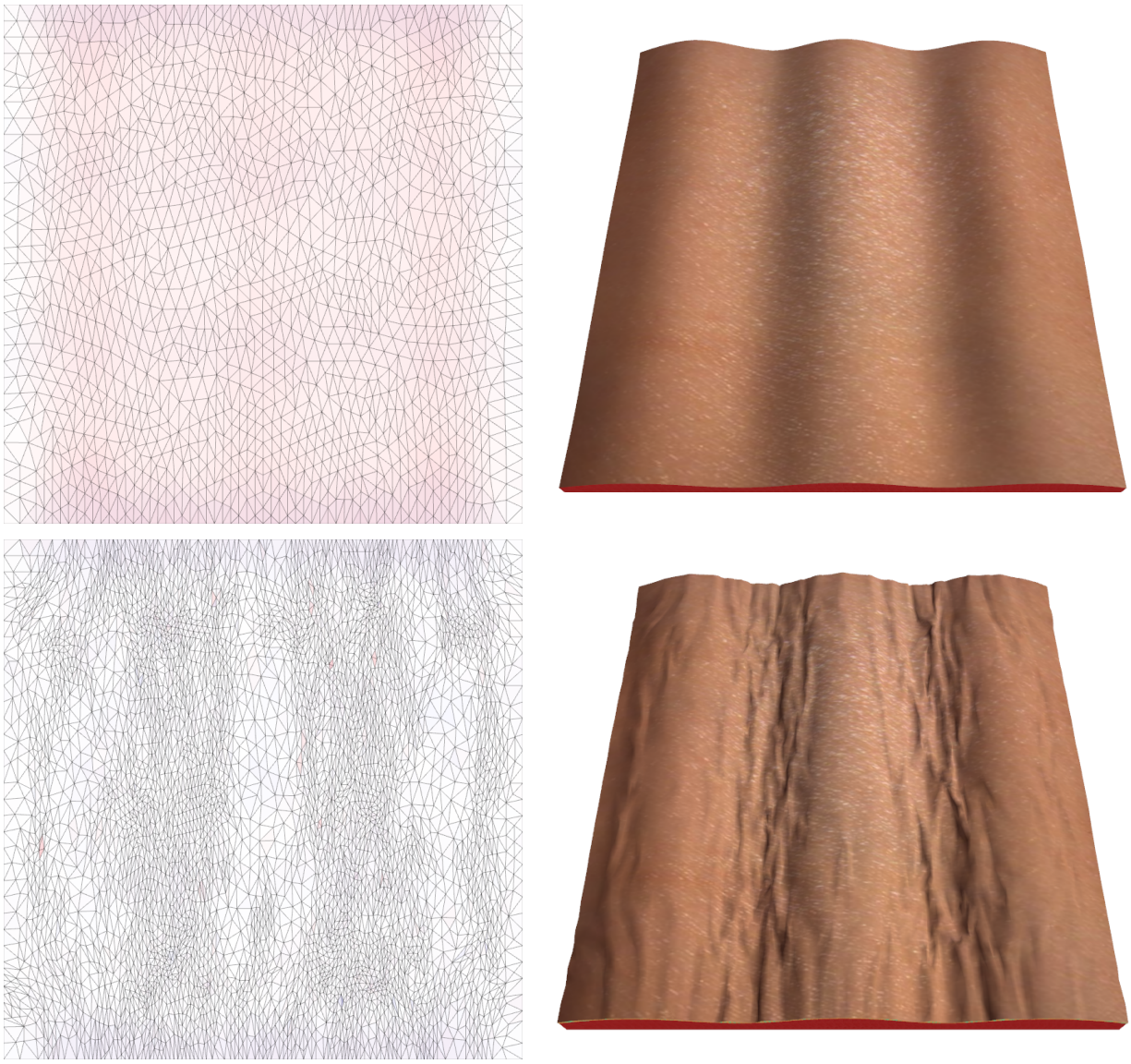

Figure 6–3: Comparison between one layer simulation (top) and two layer simulation (bottom) under the same compression.

mesh resolution is high enough to accurately model the variation on the material property maps we assume that we do not have artifacts due to numerical coarsening. Alternatively, if sharp material boundaries were to be introduced in the material property map, these could be handled by constraining the adaptive mesh such that it always places edges along the discontinuity.

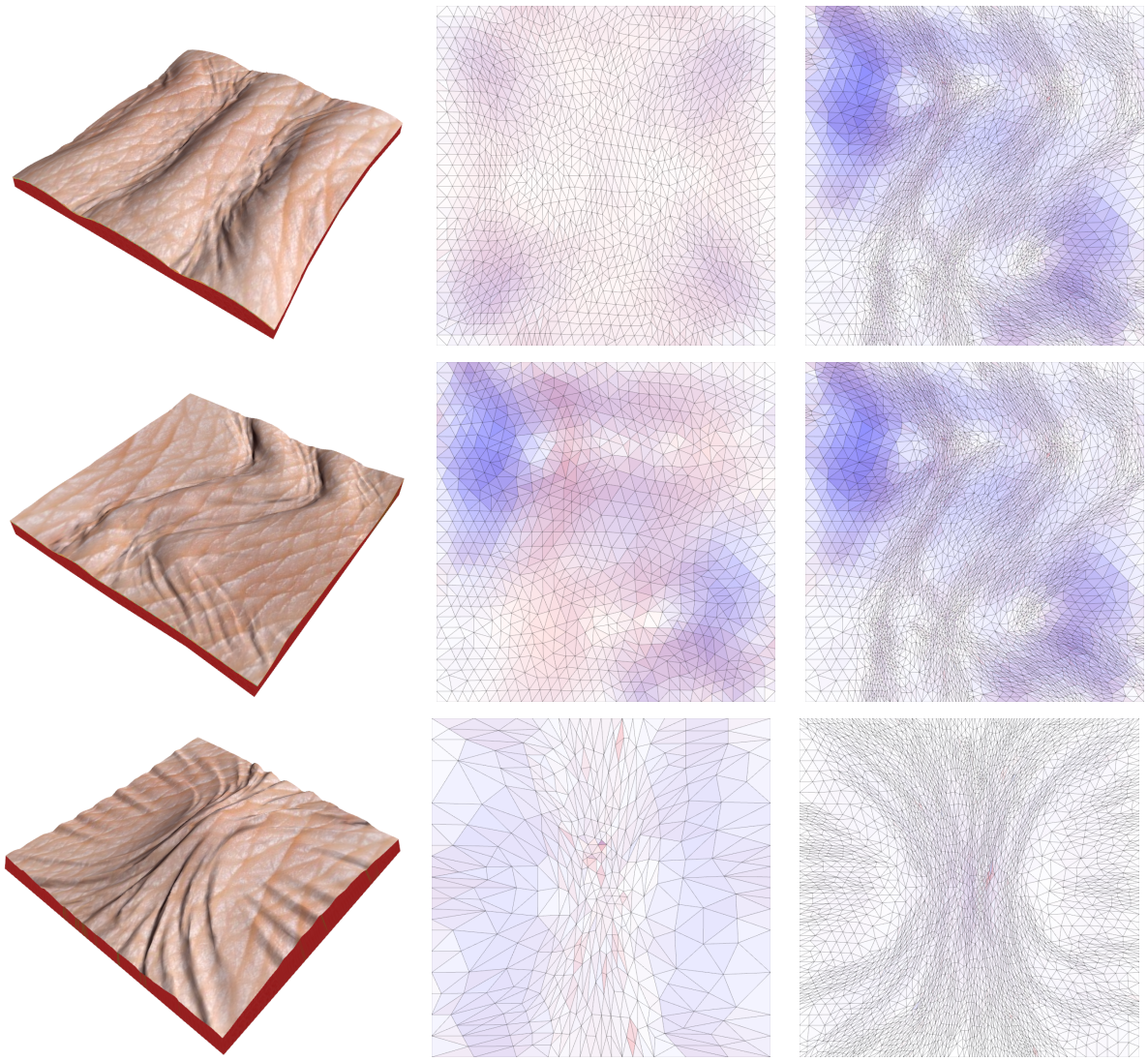

Figure 6–4: Additional examples of multi-layer simulations that mimicking human skin wrinkling patterns during compressing, twisting and pinching. At center and left are the adaptive meshes in material space, where blue indicates stretching and red indicates compression.

We assume that the material space embedding has low distortion, and therefore that we can reasonably define the continuous constraint functions in material space. When there is high distortion in the embedding, it is difficult to assign a meaningful truncation distance and weights. One potential solution might be to build a distance mapping for each cluster in material space based on the geodesic distances we computed in world space, which is similar to the generation of contour lines in topographic maps. Each cluster would have its own continuous distance mapping in material space. We could then estimate a continuous

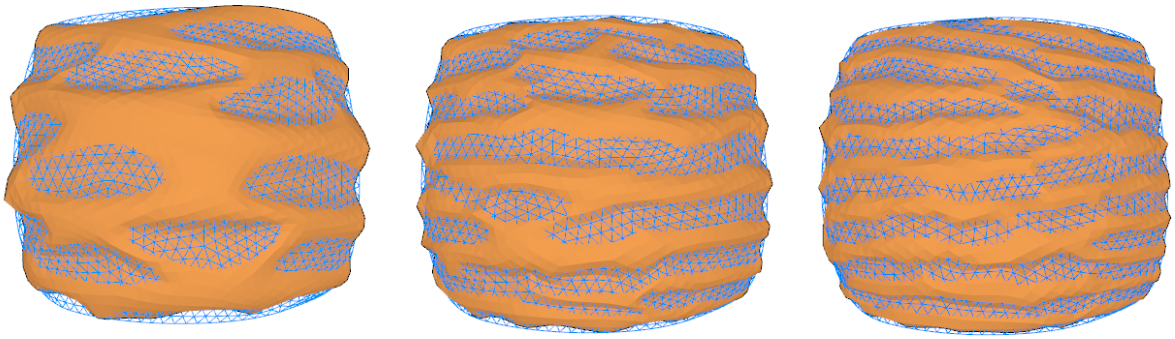

Figure 6–5: This is the preliminary result of one non-flat examples. From left to right, its wrinkle wavelength becomes smaller as its thickness decreasing. The blue meshes are embedded shape with respect to the driving blend shape.

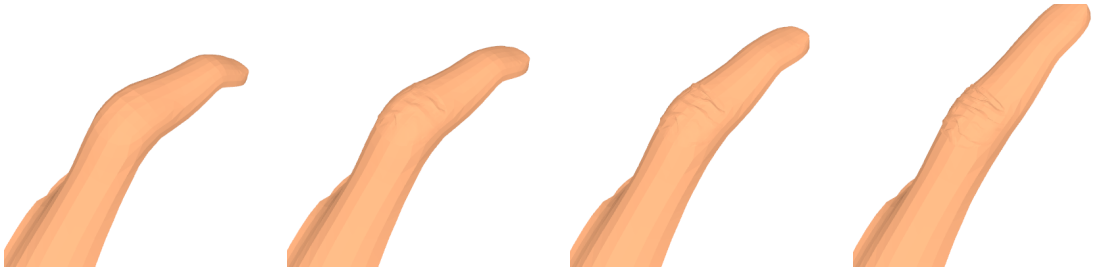

Figure 6–6: This is the preliminary result of a finger example. We can observe the knuckle starts to wrinkling as the finger is straightening. From left to right, figures show the simulation results with frame 00, frame 30, frame 60 and frame 90.

mapping for sampling weights after remeshing. Figure 6–5 is a one layer non-flat example with varying wrinkle wavelength and a finger shape example is shown in Figure 6–6. They both are using constant resolution meshes and constraints are all precomputed, which is reimplementation of previous work by Remillard and Kry [25]. We would try to make non-flat examples with adaptive mesh and adaptive constraints in the future.

In the two layer simulation, we sometimes observe regions around the wave crest that are less wrinkled, especially in examples with a large thickness. Looking at Equation 4.1, we can see that the average weighted position of upper layer is constrained in the direction of the weighted normal of lower layer. When the lower layer starts to wrinkle, normals on the two sides of wave crest spread out. This introduces wrinkling in the upper layer which prevents wrinkles from forming. If we could endow thickness constraints with an elastic

property, our solver might be able to produce configurations of skin with minimum energy that still exhibit wrinkles at the top of the wave.

CHAPTER 7
Conclusions

Our multi-layer physics-based skin model improves upon the single layer model for wrinkle simulation. Permitting heterogeneous material properties is important for improving simulation fidelity, and is addressed by our model. Simulating each layer with an adaptive mesh allows for reasonable computation times with respect to the size of the fine details that can be achieved. By designing continuous constraint functions that respect the predicted wrinkling wavelengths, we transform the problem of assigning constraints into a sampling problem. By using blend shapes to drive the bottom later, and using texture maps to define the heterogeneous elastic properties and varying layer thickness, we have a system that can be used to produce fine physical wrinkling details into standard character animation models. While our demonstrations focus on small patches of skin, we plan to apply the technique to the creation of wrinkles on faces and fingers. Finally, because we can easily construct wrinkle maps from our high resolution simulations, we expect that this technique can find use in interactive applications, such as video games.

7.1 Future work

In continuing this work, it would be attractive to expand our simulation examples to non flat and pre-wrinkled geometry models, such as fingers and character faces. It would be necessary to build better constraints that accommodate the high distortion these examples suffer from in a planar-embedding material space. We also want to include anisotropic material properties of skin in our physical simulation, in addition to heterogeneous properties we address in this work. Finally, another approach to construct constraints based on the user-input strokes is worth exploring. Specifically, allowing artists to build desired constraints only with simple sketching would be an interesting interface to designing physically based wrinkles within our framework.

References

[1] Sheldon Andrews, Marc Jarvis, and Paul G. Kry. Data-driven fingertip appearance for interactive hand simulation. In *Proceedings of Motion on Games*, MIG '13, pages 155:177–155:186, New York, NY, USA, 2013. ACM.

[2] Yosuke Bando, Takaaki Kuratate, and Tomoyuki Nishita. A simple method for modeling wrinkles on human skin. In *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, PG '02, pages 166–175, 2002.

[3] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 43–54, 1998.

[4] Miklós Bergou, Saurabh Mathur, Max Wardetzky, and Eitan Grinspun. Tracks: toward directable thin shells. *ACM Trans. Graph.*, 26(3):50:1–50:10, July 2007.

[5] James F. Blinn. Simulation of wrinkled surfaces. *SIGGRAPH Comput. Graph.*, 12(3):286–292, August 1978.

[6] R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '03, pages 28–36, 2003.

[7] Robert Bridson, Ronald Fedkiw, and John Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph.*, 21(3):594–603, July 2002.

[8] E. Cerda and L. Mahadevan. Geometry and physics of wrinkling. *Phys. Rev. Lett.*, 90:074302:1–4, Feb 2003.

[9] Zhili Chen, Renguo Feng, and Huamin Wang. Modeling friction and air effects between cloth and deformable bodies. *ACM Trans. Graph.*, 32(4):88:1–88:8, July 2013.

[10] Lawrence D. Cutler, Reid Gershbein, Xiaohuan Corina Wang, Cassidy Curtis, Erwan Maigret, Luca Prasso, and Peter Farson. An art-directed wrinkle system for CG character clothing and skin. *Graphical Models*, 69(5-6):219–230, 2007. Special Issue on SCA 2005.

[11] Cormac O. Flynn and Brendan A.O. McCormack. A three-layer model of skin and its application in simulating wrinkling. *Computer Methods in Biomechanics and Biomedical Engineering*, 12(2):125–134, 2009.

[12] Marion Geerligs. *Skin layer mechanics*. PhD thesis, Technische Universiteit Eindhoven, 2010.

[13] J. L Gennisson, T. Baldeweck, M. Tanter, S. Catheline, M. Fink, L. Sandrin, C. Cornillon, and B. Querleux. Assessment of elastic parameters of human skin using dynamic elastography. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 51(8):980–989, Aug 2004.

[14] Jan Genzer and Jan Groenewold. Soft matter with hard skin: From skin wrinkles to templating and material characterization. *Soft Matter*, 2:310–323, 2006.

[15] Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '03, pages 62–67, 2003.

[16] David Harmon, Etienne Vouga, Rasmus Tamstorf, and Eitan Grinspun. Robust treatment of simultaneous collisions. *ACM Trans. Graph.*, 27(3):23:1–23:4, August 2008.

[17] Jorge Jimenez, Jose I. Echevarria, Christopher Oat, and Diego Gutierrez. *GPU Pro 2*, chapter Practical and Realistic Facial Wrinkles Animation, pages 15–27. AK Peters Ltd., 2011.

[18] Ladislav Kavan, Dan Gerszewski, Adam W. Bargteil, and Peter-Pike Sloan. Physics-inspired upsampling for cloth simulation in games. *ACM Trans. Graph.*, 30(4):93:1–93:10, July 2011.

[19] N. Magnenat-Thalmann, P. Kalra, J. Luc Leveque, R. Bazin, D. Batisse, and B. Querleux. A computational skin model: fold and wrinkle formation. *IEEE Transactions on Information Technology in Biomedicine*, 6(4):317–323, December 2002.

[20] Matthias Müller and Nuttapong Chentanez. Wrinkle meshes. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '10, pages 85–92, 2010.

[21] Matthias Müller and Markus Gross. Interactive virtual materials. In *Proceedings of Graphics Interface 2004*, pages 239–246, 2004.

[22] Rahul Narain, Tobias Pfaff, and James F. O'Brien. Folding and crumpling adaptive sheets. *ACM Trans. Graph.*, 32(4):51:1–51:8, July 2013.

[23] Rahul Narain, Armin Samii, and James F. O'Brien. Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph.*, 31(6):152:1–152:10, November 2012.

[24] Xavier Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Proceedings of Graphics Interface '95*, pages 147–154, 1995.

[25] Olivier Rémillard and Paul G. Kry. Embedded thin shells for wrinkle simulation. *ACM Trans. Graph.*, 32(4):50:1–50:8, July 2013.

[26] Damien Rohmer, Tiberiu Popa, Marie-Paule Cani, Stefanie Hahmann, and Alla Sheffer. Animation wrinkling: Augmenting coarse cloth simulations with realistic-looking wrinkles. *ACM Trans. Graph.*, 29(6):157:1–157:8, December 2010.

[27] J. C. Simo and D. D. Fox. On stress resultant geometrically exact shell model. Part I: formulation and optimal parametrization. *Comput. Methods Appl. Mech. Eng.*, 72(3):267–304, March 1989.

[28] Min Tang, Dinesh Manocha, and Ruofeng Tong. Fast continuous collision detection using deforming non-penetration filters. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, I3D '10, pages 7–13, 2010.

[29] Bernhard Thomaszewski, Simon Pabst, and Wolfgang Strasser. Continuum-based strain limiting. *Computer Graphics Forum*, 28(2):569–576, 2009.

[30] Bernhard Thomaszewski, Markus Wacker, and Wolfgang Strasser. A consistent bending model for cloth simulation with corotational subdivision finite elements. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '06, pages 107–116, 2006.

[31] Stephen P. Timoshenko and James M. Gere. *Theory of Elastic Stability*. Dover Civil and Mechanical Engineering Series, 2009.

[32] Pascal Volino and Nadia Magnenat-Thalmann. Simple linear bending stiffness in particle systems. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '06, pages 101–105. Eurographics Association, 2006.

[33] Huamin Wang, Florian Hecht, Ravi Ramamoorthi, and James O'Brien. Example-based wrinkle synthesis for clothing animation. *ACM Trans. Graph.*, 29(4):107:1–107:8, July 2010.

[34] Huamin Wang, James F. O'Brien, and Ravi Ramamoorthi. Multi-resolution isotropic strain limiting. *ACM Transactions on Graphics*, 29(6):156:1–10, December 2010. Proceedings of ACM SIGGRAPH Asia 2010, Seoul, South Korea.

[35] Huamin Wang, James F. O'Brien, and Ravi Ramamoorthi. Data-driven elastic models for cloth: modeling and measurement. *ACM Trans. Graph.*, 30(4):71:1–71:12, July 2011.

[36] Mark Warburton and Steve Maddock. Physically-based forehead animation including wrinkles. *Computer Animation and Virtual Worlds*, pages n/a–n/a, 2014.

[37] Martin Wicke, Daniel Ritchie, Bryan M. Klingner, Sebastian Burke, Jonathan R. Shewchuk, and James F. O'Brien. Dynamic local remeshing for elastoplastic simulation. *ACM Transactions on Graphics*, 29(4):49:1–49:11, July 2010.

[38] Jie Yin, Gregory J. Gerling, and Xi Chen. Mechanical modeling of a wrinkled fingertip immersed in water. *Acta Biomaterialia*, 6(4):1487 – 1496, 2010.

[39] Zhenglong Zhou, Bo Shu, Shaojie Zhuo, Xiaoming Deng, Ping Tan, and Stephen Lin. Image-based clothes animation for virtual fitting. In *SIGGRAPH Asia 2012 Technical Briefs*, SA '12, pages 33:1–33:4, 2012.