# Computational Bodybuilding: Anatomically-based Modeling of Human Bodies

Shunsuke Saito*
University of Pennsylvania, Waseda University

Zi-Ye Zhou†
University of Pennsylvania

Ladislav Kavan‡
University of Pennsylvania
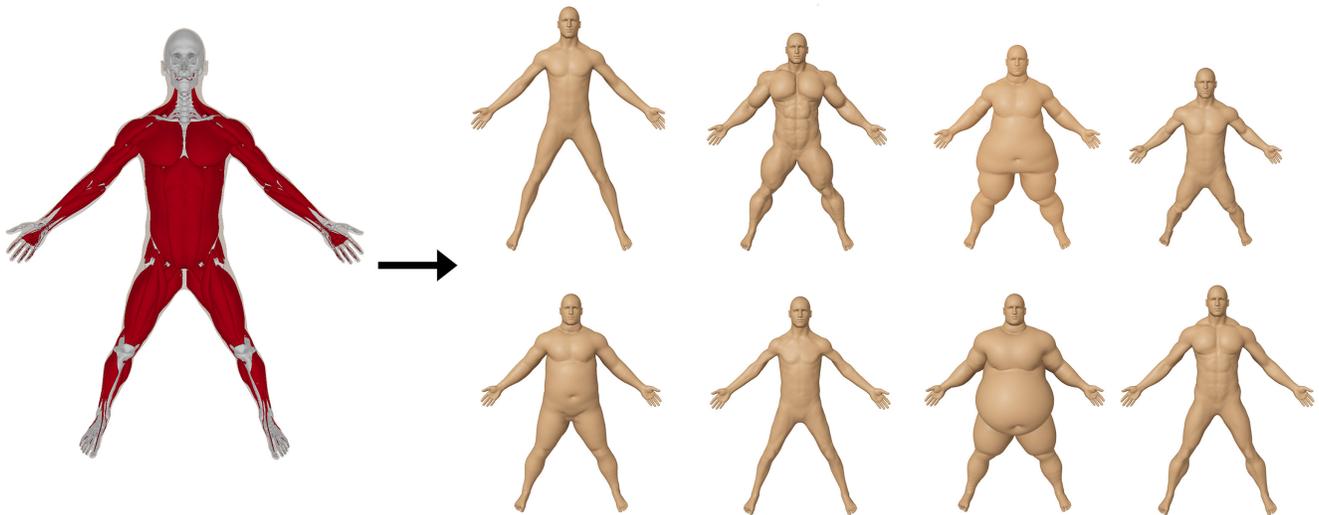
**Figure 1:** *Given an input 3D anatomy template, we propose a system to simulate the effects of muscle, fat, and bone growth. This allows us to create a wide range of human body shapes.*

## Abstract

We propose a method to create a wide range of human body shapes from a single input 3D anatomy template. Our approach is inspired by biological processes responsible for human body growth. In particular, we simulate growth of skeletal muscles and subcutaneous fat using physics-based models which combine growth and elasticity. Together with a tool to edit proportions of the bones, our method allows us to achieve a desired shape of the human body by directly controlling hypertrophy (or atrophy) of every muscle and enlargement of fat tissues. We achieve near-interactive run times by utilizing a special quasi-statics solver (Projective Dynamics) and by crafting a volumetric discretization which results in accurate deformations without an excessive number of degrees of freedom. Our system is intuitive to use and the resulting human body models are ready for simulation using existing physics-based animation methods, because we deform not only the surface, but also the entire volumetric model.

**CR Categories:** I.3.7 [Computer Graphics]: Three-Dimensional Graphics—Animation

**Keywords:** human body modeling, 3D anatomy, growth models, physics-based animation.

*shunsuke.saito16@gmail.com
†ziyezhou.jerry@gmail.com
‡ladislav.kavan@gmail.com

## 1 Introduction

Human bodies exhibit large variations in size and shape due to factors such as height, muscularity, or adiposity. Although modern animations systems such as Weta Tissue can bring realistic digital humans to life by combining physics-based simulation with detailed models of 3D anatomy, creating simulation-ready anatomical models is expensive and typically involves a team of specialized digital artists experienced in 3D modeling, simulation, and anatomy. This *de facto* limits the applicability of realistic human body simulations to high-budget productions.

In this paper, we propose a method to generate a wide range of human body shapes which are ready to be simulated using existing physics-based methods. Our approach is intuitive even to inexperienced users, because it is motivated by natural biological processes. Inspired by growth laws studied in biomechanics [Taber 1995; Taber 1998; Jones and Chapman 2012; Wisdom et al. 2015], we propose mathematical models of hypertrophy and atrophy of skeletal muscles. We devise another model for growth of fat tissues. Our growth models are tightly coupled with a physics-based simulation system, which accounts for elasticity of the individual organs, e.g., as the biceps muscle grows due to hypertrophy, it exerts forces on the adjacent tissues, pushing them out of the way. We combine our models of soft tissue growth with a geometric shape deformation technique to edit the rest pose of the body, allowing us to produce human bodies with varied anthropometry, such as heights and bone lengths.

Our goal is fundamentally different from existing simulators [Lee et al. 2009] which focus on modeling the *motion* of the human body such as walking or lifting an object. Even though our method shares similar building blocks, e.g., volumetric elasticity and finite element methods, we simulate changes of the human body over the *long term*. Growth processes such as hypertrophy do not occur instantaneously and lead to quite different shapes than, e.g., muscle

(a) Original      (b) Muscle Contraction      (c) Muscle Hypertrophy
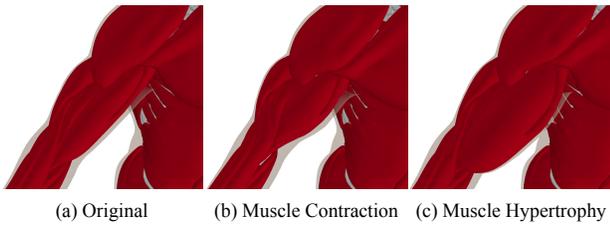
**Figure 2:** *Voluntary muscle contractions (muscle flexing) occur over the short-term and preserve volume (b). Muscle hypertrophy (growth) increases the volume and leads to a different shape (c).*

contractions (Figure 2). On a mathematical level, our growth models are analogous to plasticity, i.e., permanent changes to the rest pose. Specifically, for muscle hypertrophy (or atrophy), we use an anisotropic model which expands the muscle perpendicularly to the direction of muscle fibers. Because individual muscle fibers may not be modeled in our 3D anatomy template, we provide a method to estimate the fiber directions by simulating "flow" between tendons, which amounts to solving a Laplace's equation with carefully chosen boundary conditions. We propose a different model for fat hypertrophy, because the growth of fat tissues is fundamentally different from muscles. Fat behaves as a semi-fluid material and grows in arbitrary directions in order to minimize stretching of the skin. Our fat growth model is inspired by the elastic behavior of inflated rubber balloons which delivers surprisingly realistic obese body shapes.

Our system aims to provide an interactive experience, because immediate visual feedback is highly advantageous to users who are designing characters for movies, games, medical visualizations, or workplace design applications. However, achieving fast computations is challenging, because volumetric body models require many degrees of freedom to resolve the intricate anatomical structures, which makes traditional physics-based simulation methods slow. We achieve a near-interactive performance by 1) employing a slightly modified "Projective Dynamics" solver [Bouaziz et al. 2014] and 2) carefully crafting our volumetric discretization. We found it critical to design our tetrahedral mesh so that it contains a clean interface between the muscles and fat tissue. Using this "muscle envelope" structure, we were able to reduce our meshes to only 76k vertices without compromising quality – even after extreme growth – achieving a near-interactive run time experience.

Our resulting body shapes are created by physics-based simulation which takes into account volume conservation, i.e., after growth there are no more changes to the volume. This corresponds to the fact that most soft tissues in the human body can be considered incompressible due to their high water content. This also prevents artifacts such as self-intersections, ensuring that our output models are ready to be simulated using existing physics-based animation tools and/or displayed using volumetric rendering techniques, e.g., when visualizing individual organs. Our system is intuitive and its use requires no experience in 3D modeling or knowledge of human anatomy. In addition to film, games, and visual effects, our method can be also very useful in educational or training applications to visualize the effects of muscle training or obesity. To our knowledge, our system is to first to propose physics-based modeling of human body growth in computer graphics.

## 2 Related Work

The laws of physics play a key role in determining the structure and shape of living organisms [Thompson 1942]. The utility of anatomically-based modeling in computer graphics was recognized early on [Wilhelms and Van Gelder 1997; Scheepers et al. 1997]. A recent survey [Lee et al. 2010] focuses on modeling of skeletal muscles, but provides also a good overview of general anatomically-based modeling in computer graphics.

Data-driven techniques represent a well-explored approach to creating a wide range of human bodies, assuming a database of input body shapes is available [Allen et al. 2003; Seo and Magnenat-Thalmann 2003]. The SCAPE model [Anguelov et al. 2005] accounts for changes due to both subject shape and pose. Statistical shape models continue to steadily improve [Hasler et al. 2009; Chen et al. 2013; Neumann et al. 2013b; Loper et al. 2014]. Recent methods also capture effects such as isometric muscle contractions [Neumann et al. 2013a] and soft tissue motion [Loper et al. 2014], i.e., changes of shape which cannot be explained by pose alone. Statistical shape models have proven useful in image- and video-based editing of human bodies [Zhou et al. 2010; Jain et al. 2010]. While popular and powerful, data-driven methods have their limitations. First, the results are only as good as the input data. It may be difficult to capture a sufficient number of less common body shapes, such as bodybuilders or extremely obese individuals. Second, typical databases contain only surface measurements and lack information about the internal organs. This means the models are not ready for physics-based simulation and it is hard or even impossible to request specific anatomically-based changes, e.g., increase the size of the biceps or the amount of subcutaneous fat in the belly.

Physics-based simulation of the human body is another area which received considerable attention. Robust methods to simulate skeletal muscles and related anatomical structures have been developed, producing impressive visual results including muscle activations and bulging [Teran et al. 2003; Sifakis et al. 2005; Teran et al. 2005b; Teran et al. 2005a]. More recent work considered a comprehensive biomechanical upper body model including control [Lee et al. 2009], efficient numerical techniques for fast elasticity computations [Patterson et al. 2012], and muscle-actuated full body control for various types of gait [Geijtenbeek et al. 2013; Lee et al. 2014] and swimming [Si et al. 2015]. It is important to distinguish muscle hypertrophy from bulging due to activations (such as when lifting a heavy object): when a muscle is activated, the fibers contract and bulge due to incompressibility, i.e., the overall volume of the muscle is preserved [Fan et al. 2014]. In our system, we simulate hypertrophy, a long term growth process which changes the volume of the muscle (Figure 2). Even though our model shares similar building blocks, such as volumetric elasticity and finite element methods, modeling of growth introduces new challenges, such as finding suitable mathematical growth models. In addition to muscles, we simulate also the growth of fat. An additional challenge specific to growth simulation is that significant changes of volume are more sensitive to the underlying volumetric discretization.

Even though most previous work uses traditional Lagrangian approaches to elasticity, recently introduced Eulerian methods to simulate elastic solids show great promise [Pai et al. 2014]. In particular, the combination of Eulerian and Lagrangian viewpoints seems to be particularly well suited for simulation of musculoskeletal systems [Fan et al. 2014]. In our current system, we continue to rely on the classical Lagrangian approach, in particular because it allows us to directly apply a recently introduced fast physics solver (Projective Dynamics) [Bouaziz et al. 2014]. However, we believe that exploring Eulerian approaches to growth simulation could be a fruitful direction in the future.

Growth of living tissues plays an important factor in nature [Fung 1990]. Our growth models are based on related work in biomechanics [Rodriguez et al. 1994; Taber 1995; Taber 1998; Jones and Chapman 2012; Wisdom et al. 2015], however, it is important to
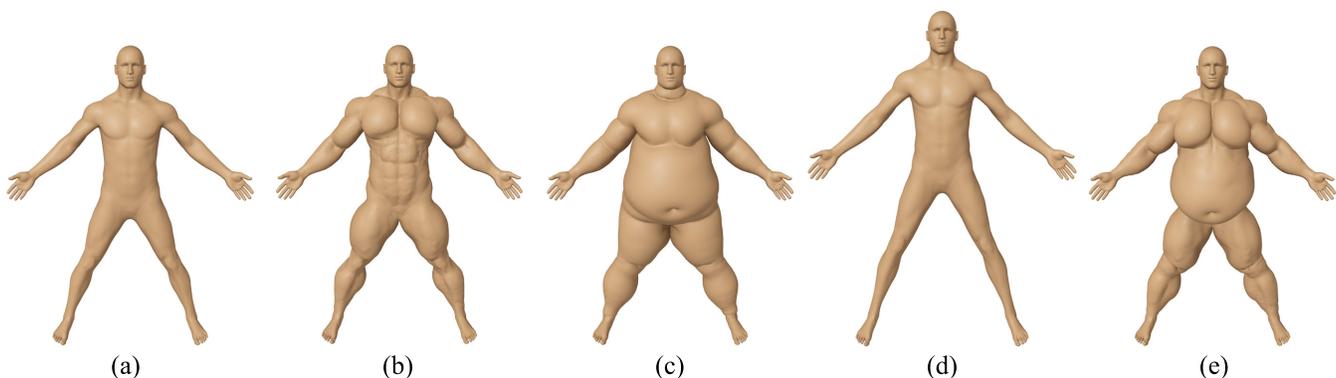
**Figure 3:** *Overview of our system: starting with a template anatomy model (a), we propose physics-based models for muscle (b) and fat growth (c). Complemented with a geometric shape editing tool to change the lengths of the limbs (d), we are able to generate a wide variety of human body shapes (e).*

note that their goals are quite different from ours. Biomechanics aims to explain the growth processes as accurately as possible, often on the microscopic level and focusing only on certain types of tissue. Skeletal muscle growth has been also studied in the context of bodybuilding [D'Antona et al. 2006]. We developed our growth laws specifically for applications in computer graphics, with primary design decisions being visual quality, ease of use, and computational efficiency.

Modeling tools commonly used in computer graphics are typically oriented on modeling of polygonal surfaces [Sorkine et al. 2004], even though some methods also support volumetric deformations [Botsch et al. 2007; Chao et al. 2010]. Non-uniform stretching was studied in the context of man-made shapes [Kraevoy et al. 2008] and, more generally, skinned models [Jacobson and Sorkine 2011]. One of the challenges in human body deformations is the fact that it combines materials ranging from almost rigid (bones) to very supple (fat), mandating specialized shape deformation techniques [Popa et al. 2006]. The above mentioned modeling tools assume the user controls the deformation directly, typically by manipulating a set of handles. Even though expert users can use these techniques to create realistic human body shapes, this assumes the users are knowledgeable of anatomy, because they rely solely on their imagination when manipulating the control handles; it is easy to create unrealistic shapes. Instead, our system provides physics-based growth laws, which are intuitive to understand because they mimic natural growth processes well-known from the real world.

A special type of human body deformations was explored by Reinert et al. [2012], who proposed a method to create a Homunculus – a human figure where the size of each body part is proportional to an input distribution, e.g., neural density. However, internal anatomical structures are not modeled. The Anatomy Transfer system [Ali-Hamadi et al. 2013] can deform an input anatomical template (similar to the one we are using) to conform to an arbitrary target surface. However, Anatomy Transfer assumes that the target body shape is already known, including its full correspondences with the template skin. Instead, our system allows us to create brand new body shapes.

## 3 Method

The input to our system are 3D polygonal models representing bones, major muscles, and skin. In our experiments we use the commercially available Zygote body with 111 muscles and 204 bones, see Figure 1. The main component of our system is physics-
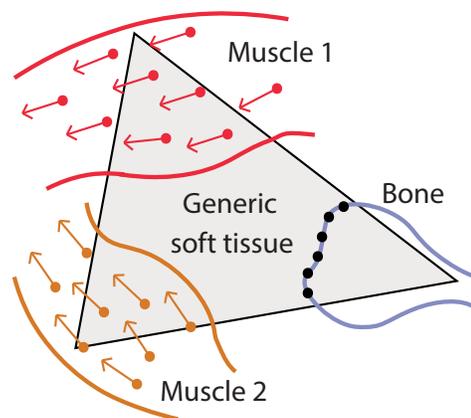


**Figure 4:** *Our simulation mesh allows a tetrahedron to be occupied by multiple materials. The resulting deformation of the tetrahedron averages the contributions of the individual elements.*

based simulation of the growth of soft tissues. We differentiate three types of volumetric soft tissues: 1) muscles, 2) fat, and 3) generic soft tissues which represent the remaining parts of the body, such as the internal organs or the blood vessels. We devise growth laws only for the muscles and subcutaneous fat, the remaining soft tissues are deformed as non-growing elastic solids. During our growth simulation, the bones are treated as static rigid bodies. We use a geometric shape deformation process in order to change the lengths and proportions of individual body parts (Figure 3(d)). This "bone editing" process changes the lengths and shapes of the bones and deforms the rest of the anatomical template accordingly, producing a new rest pose for the subsequent physics-based simulation of soft tissue growth.

Somewhat counter-intuitively, we do not take gravity into account during our physics-based growth simulation, as if we were simulating growth in a zero-gravity environment. We assume that our resulting body model will be used as input of traditional simulation of musculoskeletal motion [Lee et al. 2009] which will introduce the effects of gravity. We implement a light-weight version of this simulator to allow users to preview our resulting body shapes after adding gravity (Figure 3(e)).

Our soft-tissue simulation solves quasi-statics elasticity to account for the fact that grown tissues compete for space, have discrete

boundaries, and push each other out of the way. We are aiming for interactive run time speeds, which is challenging because non-linear elasticity is computationally expensive. We found that carefully crafted volumetric discretization is critical for achieving fast and visually pleasing results. Specifically, we use irregular tetrahedral meshes conforming to the input skin. We also experimented with tet-meshes conforming to the shape of the internal organs, but their resolution is prohibitively high, even for coarsely tesselated bones and muscles. Instead, we allow each tetrahedron to be occupied by multiple materials and we average their contributions (Figure 4). Previous work has shown this approach to be effective [Sifakis et al. 2005; Lee et al. 2009]. Unfortunately, without further precautions this leads to visible artifacts due to the non-smooth boundaries between individual organs (Section 3.2). These artifacts are analogous to aliasing, well-known in rendering. We found that the key to obtaining nice results is to introduce degrees of freedom to *smoothly* separate muscles from fat. We call the interface between muscles and fat the "muscle envelope," which is a polygon mesh similar to shrunken skin. By creating a tet-mesh conforming to both the skin and the muscle envelope, we successfully eliminated the aliasing artifacts without introducing too many elements: our final tet-mesh has only 76,087 vertices, which facilitates near-interactive growth simulation.
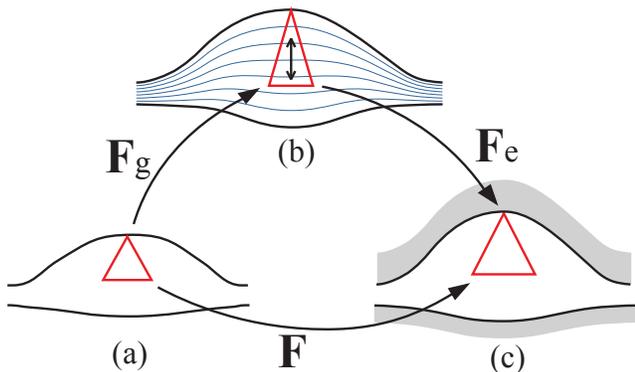


**Figure 5:** *Our skeletal muscle growth model decomposes the deformation gradient $\mathbf{F}$ in two parts: $\mathbf{F}_g$ which corresponds to pure growth and $\mathbf{F}_e$ which accounts for the effects of elasticity. Muscle growth occurs perpendicularly to the muscle fiber direction.*

### 3.1 Muscle Growth Model

Our muscle growth model comprises plastic and elastic components. Even though plasticity is typically applied to simulate non-reversible changes in material structure due to deformation [Bargteil et al. 2007], the same mathematical model has also been used in biomechanics to simulate soft tissue growth [Rodriguez et al. 1994; Ben Amar and Goriely 2005; Jones and Chapman 2012]. Our method applies the same principle even though it does not aspire to predictive simulations or biomechanical accuracy. We use linear finite elements defined on tetrahedra [Sifakis and Barbic 2012]. The deformation gradient $\mathbf{F} \in \mathbb{R}^{3 \times 3}$ is a linear transformation between the rest-pose state of a tetrahedron and its current (deformed) configuration. We decompose the deformation gradient $\mathbf{F}$ into $\mathbf{F} = \mathbf{F}_e \mathbf{F}_g$, where $\mathbf{F}_g \in \mathbb{R}^{3 \times 3}$ is the growth component and $\mathbf{F}_e \in \mathbb{R}^{3 \times 3}$ is the elastic component (Figure 5). Our method first determines $\mathbf{F}_g$ using our growth model, which simulates the effect of hypertrophy or atrophy of a single muscle without considering the elastic response of the adjacent soft tissues, such as other muscles which may also be growing. To account for this elastic be-

havior, $\mathbf{F}_e$ is defined as $\mathbf{F}_e = \mathbf{F}\mathbf{F}_g^{-1}$, assuming that $\mathbf{F}_g$ is constant. In the following we define an elastic energy as a function of $\mathbf{F}_e$ and find its minimum, i.e., a quasi-static solution. An important component of this elastic energy is quasi-incompressibility, requiring that $\det(\mathbf{F}_e)$ is close to 1. This ensures that the volume of the muscles is not changed in the quasi-static phase and is entirely controlled by the growth model.

We assume that each muscle has been segmented into areas denoted as the muscle belly and tendon regions. The key components of the muscle belly are fibers that induce muscle contraction, producing forces that generate motion of the bones. It is important to distinguish between muscle activations (flexion) and muscle hypertrophy (growth). When the muscles are activated, the fibers contract and muscles bulge due to incompressibility. Because muscle tissue has a high water content, the volume does not change during motion, e.g., bicep flexion. However, when muscles grow due to hypertrophy, the volume does change – an effect coveted by bodybuilders. Specifically, muscle hypertrophy (or atrophy) changes muscle volume in a plane perpendicular to muscle fiber directions [Taber 1998]. This makes sense intuitively, because muscles cannot change their length if they are to remain functional. For now, let us now assume that a vector field of fiber directions is known for all muscles (we describe our fiber direction estimation process in Section 3.1.1). Let $\mathbf{D} \in SO(3)$ be a rotation that aligns the x-axis with the fiber direction. Our muscle growth model can be then expressed as

$$\mathbf{F}_g(\alpha) = \mathbf{D}^\mathsf{T} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha \end{bmatrix} \mathbf{D} \qquad (1)$$

where $\alpha$ is a parameter controlling hypertrophy ($\alpha > 1$) or atrophy ($\alpha < 1$) of a given muscle. This achieves the desired effect of expanding (or shrinking) muscle tissues perpendicularly to the fiber direction. The growth is only applied to the belly of the muscle, the tendon does not grow.

To achieve interactive run times, the tetrahedra of our volumetric discretization do not conform to the geometry of the muscles. Consequently, each tetrahedron can be occupied by more than one muscle and we blend their individual growth contributions. To compute the growth deformation gradient $\mathbf{F}_g$ for a tetrahedron, we first compute a deformation gradient $\mathbf{F}_g^m$ which corresponds to growth only due to muscle $m$ for each of the contained muscles $m$. Second, we blend the deformation gradients $\mathbf{F}_g^m$ with weights corresponding to the fraction of volume the muscle $m$ occupies in the tetrahedron (Figure 4). Note that an element does not have to be occupied entirely by muscles; the fraction corresponding to generic soft tissues or bones does not grow and therefore is represented by an identity matrix. We do not need to account for blending with fat because our discretization cleanly separates muscles from subcutaneous fat (Section 3.2).

We estimate the fraction of muscle volume in a tetrahedron using Monte-Carlo sampling. Instead of the straightforward linear blending of matrices $\mathbf{F}_g^m$, we found it is more accurate to use non-linear matrix interpolation based on polar decomposition [Shoemake and Duff 1992]. In particular, we compute polar decomposition of all input matrices $\mathbf{F}_g^m$, and blend the rotation components using quaternions (blending the quaternions linearly and re-normalizing [Kavan and Zara 2005]).

After computing the growth deformation gradients $\mathbf{F}_g$, the next step is to define an elastic energy and find its minimum, i.e., a quasi-static solution. The classically applied Newton's method is time consuming because it requires us to compute second order derivatives of the energy and solve a new system of linear equations at

(a) Without Volume Preservation  (b) With Volume Preservation
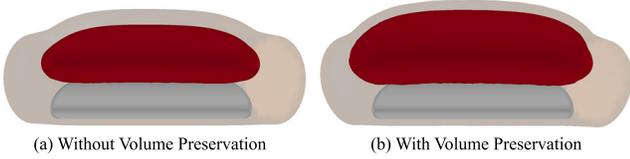
**Figure 6:** *Without volume preservation, corotated elasticity loses 17% of the total volume, canceling a significant fraction of the simulated muscle growth (left). Our method achieves incompressibility up to an error of 1.5% (right).*

each iteration. We obtained a much more responsive simulator by using a "Projective Dynamics" solver, which employs local-global alternation instead of the traditional Newton's method [Bouaziz et al. 2014]. The trade-off of Projective Dynamics is that it requires a special form of elastic energies: assuming deformation gradient $\mathbf{F}$ as the main building block, only energies of the form $||\mathbf{F} - \mathcal{P}(\mathbf{F})||_F^2$ are allowed, where $\mathcal{P} : \mathbb{R}^{3 \times 3} \to \mathbb{R}^{3 \times 3}$ is a projection operator. A wide choice of projection operators is possible as long as we can provide their efficient implementation (derivatives are not necessary). In a nutshell, the Projective Dynamics solver alternates between projections (local step) and solving a constant linear system (global step). We refer the reader to Bouaziz et al. [2014] for more details.

In the following, we formulate our elastic energies in a form suitable for Projective Dynamics. As discussed above, we use only the elastic component of the deformation gradient $\mathbf{F}_e = \mathbf{F}\mathbf{F}_g^{-1}$. Because we assume that $\mathbf{F}_g$ is fixed, $\mathbf{F}_e$ is a linear function of the current state $\mathbf{x} \in \mathbb{R}^{3n \times 1}$, where $n$ is the number of vertices of our volumetric mesh. We start with corotational elasticity:

$$E_{\text{rot}}(\mathbf{x}) = ||\mathbf{F}_e - \mathcal{P}_{SO(3)}(\mathbf{F}_e)||_F^2 \qquad (2)$$

where $\mathcal{P}_{SO(3)}$ is an operator which projects on the closest rotation. We accomplish this by computing the signed Singular Value Decomposition which guarantees that $\mathbf{U}, \mathbf{V} \in SO(3)$ [McAdams et al. 2011]. This leads to $\mathbf{F}_e = \mathbf{U}\mathbf{S}\mathbf{V}^\mathsf{T}$ and we clamp all of the singular values to one, resulting in $\mathcal{P}_{SO(3)}(\mathbf{F}_e) = \mathbf{U}\mathbf{V}^\mathsf{T}$. We do not use any Poisson term because corotational elasticity does not ensure volume preservation for large deformations [Patterson et al. 2012]. Instead, we use the following volume preservation term:

$$E_{\text{vol}}(\mathbf{x}) = ||\mathbf{F}_e - \mathcal{P}_{SL(3)}(\mathbf{F}_e)||_F^2 \qquad (3)$$

where $\mathcal{P}_{SL(3)}$ is an operator which projects on $SL(3)$, i.e., a group of matrices with determinant one. This projection is slightly more involved; if $\mathbf{S}$ are the signed singular values as before, we need to solve:

$$\mathcal{P}_{SL(3)}(\mathbf{F}_e) = \underset{\mathbf{P}}{\text{argmin}} \ ||\mathbf{S} - \mathbf{P}||_F^2 \ \text{ subj. to } \ P_{11}P_{22}P_{33} = 1 \quad (4)$$

where $\mathbf{P} \in \mathbb{R}^{3 \times 3}$ is a diagonal matrix with elements $P_{11}, P_{22}, P_{33}$. Even though the constraint $P_{11}P_{22}P_{33} = 1$ is non-convex, an effective strategy is to linearize it and solve the resulting convex approximation of Equation (4) iteratively [Boyd and Vandenberghe 2009], where each iteration amounts to a small linear system solve. In fact, we found that in our case only one iteration is sufficient, because the global step does not require an exact solution; a more accurate result will be achieved during the course of multiple local-global iterations.

Finally, we need to account for positional constraints due to bones. As illustrated in Figure 4, we uniformly sample the surface of each bone. At each sample point, we compute a vector of barycentric

coordinates $\mathbf{b}_i \in \mathbb{R}^{n \times 1}$ such that $(\mathbf{b}_i^\mathsf{T} \otimes \mathbf{I}_3)\mathbf{x}$ retrieves the coordinates of this point ($\otimes$ denotes Kronecker product and $\mathbf{I}_3 \in \mathbb{R}^{3 \times 3}$ is identity). We attach each of these points to a fixed location by zero rest-length springs, leading to the following energy term:

$$E_{\text{bone}}(\mathbf{x}) = \sum_i ||(\mathbf{b}_i^\mathsf{T} \otimes \mathbf{I}_3)\mathbf{x} - \mathbf{t}_i||^2 \qquad (5)$$

where $\mathbf{t}_i \in \mathbb{R}^3$ are the target positions. This formulation accounts for the fact that due to limited resolution, bones cannot be resolved as hard boundary conditions.

Our final potential energy of one element has the following form:

$$E_{\text{muscle}}(\mathbf{x}) = w_{\text{rot}}E_{\text{rot}}(\mathbf{x}) + w_{\text{vol}}E_{\text{vol}}(\mathbf{x}) + w_{\text{bone}}E_{\text{bone}}(\mathbf{x}) \quad (6)$$

where $w_{rot}$, $w_{vol}$, and $w_{bone}$ are the weights for each term. In our examples, we use $w_{rot} = 100$ for muscles and generic soft tissues, $w_{rot} = 1000$ for tendons, and $w_{rot} = 30$ for fat. We set $w_{bone} = 100$. To achieve sufficient incompressibility, we use $w_{vol} = 1000$. We sum volume-weighted $E_{\text{muscle}}(\mathbf{x})$ over all tetrahedrons and minimize the resulting energy using Projective Dynamics, starting from the rest pose as an initial guess.

The most time-consuming part of the Projective Dynamics solver is computing sparse Cholesky factorization of the Hessian used in the global step. Even though this Hessian does not depend on the current deformed configuration $\mathbf{x}$ (which is the main advantage of Projective Dynamics), it does, unfortunately, depend on the growth deformation gradients $\mathbf{F}_g$. This means that we would have to recompute the sparse Cholesky factorization every time we change the growth parameters, hampering interactivity. To avoid this problem, we first observe the global step of Projective Dynamics can be interpreted as one iteration of Newton's method, which is sufficient to produce a perfect solution, because the global step is a convex quadratic problem. Next, we replace the Newton step with a quasi-Newton one, i.e., proposing an "approximate global step." Instead of the correct growth-dependent Hessian, we simply use the Hessian of the rest pose (without any growth). This way, we avoid the expensive sparse Cholesky factorization when changing the growth parameters. The trade-off is that this quasi-Newton step no longer solves the global step perfectly. This is not a big problem, because the subsequent iterations of the local/global process will correct this error at a cost of extra iterations. In our experiments, we observed that using this approximate global step increases the number of iterations only slightly, while the time savings by avoiding repeated Cholesky factorizations are quite significant, see Section 4.

### 3.1.1 Fiber Estimation
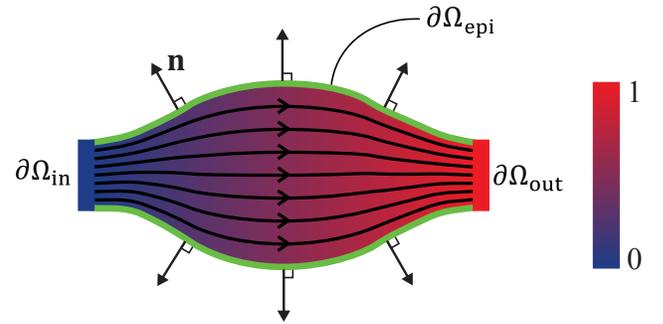


**Figure 7:** *We solve for fiber directions by modeling a hypothetical flow from one tendon ($\partial\Omega_{in}$) to another ($\partial\Omega_{out}$). There is no flow through the epimysium ($\partial\Omega_{epi}$), which forms the muscle boundary.*
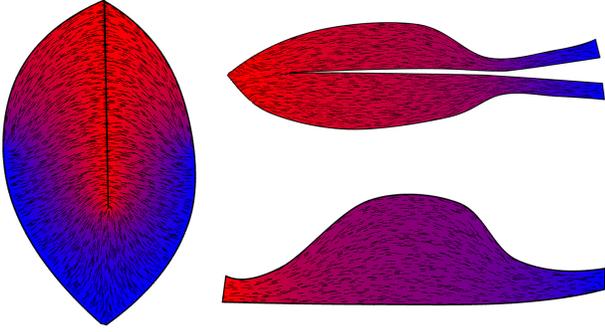
**Figure 8:** *Our Laplacian-based fiber estimation method is applicable not only to simple parallel muscles (right-bottom), but also to fusiform (right-top) and bipennate muscles (left).*



| 402,136 | 1,013,564 | 375,187 |
| tetrahedrons | tetrahedrons | tetrahedrons |
| | | (muscle envelope) |

**Figure 9:** *Volumetric discretization non-conforming to muscles results on aliasing-like artifacts on the skin (left). Increasing the resolution helps, but does not completely remove the problem, even at the cost of much slower simulation (middle). Our "muscle envelope" removes the artifacts without using too many tetrahedrons.*

Fiber directions play a key role in both muscle contractions as well as muscle growth. However, typical 3D anatomical models usually lack fiber information. Accurate fiber directions can be reconstructed using medical imaging [Levin et al. 2011]. An alternative proposed for computer graphics applications is B-spline interpolation [Agur et al. 2003; Teran et al. 2005a]. However, this requires tuning interpolation parameters which vary from muscle to muscle. We propose a simple yet effective method to estimate fiber directions inspired by recent work in biomechanics [Choi and Blemker 2013]. The key observation is that fibers seem to point in the same directions as gradients of a hypothetical flow from one tendon to another, i.e., from one end of the muscle to the other, see Figure 7. We can find this "flow" by solving a Laplace's equation with mixed boundary conditions. Specifically, at the outer surface of muscle belly (epimysium), we require the normal derivative of our scalar field $\phi$ to vanish, i.e., there is no flow through the epimysium. This amounts to Neumann boundary conditions. At the input tendon, we impose Dirichlet boundary condition fixing $\phi$ to zero and at the output tendon, we fix $\phi$ to one. Formally, we solve:

$$\Delta\phi(\mathbf{p}) = 0, \quad \forall \mathbf{p} \in \text{int}(\Omega)$$

subject to

$$\phi(\mathbf{p}) = 0, \quad \forall \mathbf{p} \in \partial\Omega_{\text{in}}$$
$$\phi(\mathbf{p}) = 1, \quad \forall \mathbf{p} \in \partial\Omega_{\text{out}}$$
$$\frac{\partial\phi}{\partial\mathbf{n}}(\mathbf{p}) = 0, \quad \forall \mathbf{p} \in \partial\Omega_{\text{epi}}$$

where $\Omega_{\text{in}}$ denotes the input tendon, $\Omega_{\text{out}}$ denotes the output tendon, and $\Omega_{\text{epi}}$ denotes the epimysium, as shown in Figure 7. The choice of input and output tendons is arbitrary because we only need the gradient of $\phi$ (the constants 0 and 1 used as Dirichlet boundary conditions are also arbitrary). We define our fiber directions as the normalized gradient of $\phi$, i.e., $\nabla\phi/||\nabla\phi||$. Note that we do not require $\partial\Omega_{\text{in}}$ or $\partial\Omega_{\text{out}}$ to be contiguous, the same formulation works also for more complex muscles as shown in Figure 8. In our implementation, we tet-mesh each muscle individually to precompute the fiber directions. These individual meshes are then discarded, and we store only the average fiber direction for each tetrahedron.

### 3.2 Volumetric Discretization

Ideally, our tetrahedral mesh would conform to the input polygonal models of bones, muscles, and skin. We tried this approach initially but found that it results in a prohibitive number of degrees of freedom, even with a relatively coarse tessellation of the input meshes. Instead, we use volumetric discretization which conforms only to the skin. This implies that tetrahedra can be occupied by multiple organs, which we resolve using material blending (Section 3.1). Unfortunately, in spite of the blending, growth simulation with tetrahedral meshes conforming only to the skin produces artifacts, see Figure 9 (left). The problem is that during muscle growth, the interface between the growing and non-growing tissues is not smooth, because the tet-mesh generator is unaware of the internal structures. This results in artifacts analogous to aliasing. Increasing the resolution helps, however, even with one million tetrahedra, some artifacts are still noticeable (Figure 9, middle).
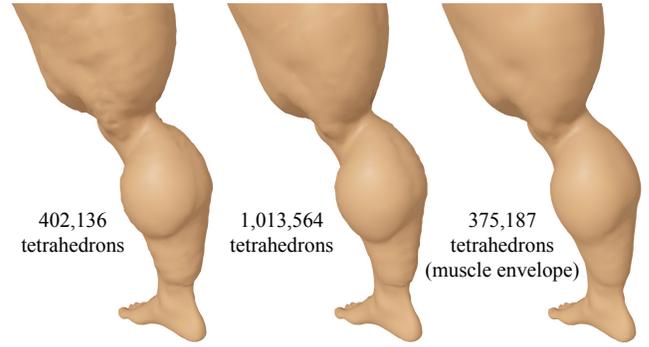
We propose a method to avoid these problems without using excessively many degrees of freedom. The key idea is to create a "muscle envelope" — a well-behaved polygonal mesh that separates the two conforming tetrahedral meshes for the subcutaneous fat and the rest of the body. We use constrained Delauney tetrahedralization [Si 2011] to ensure the resulting discretization conforms to both the skin and the muscle envelope. The resulting tet-mesh cleanly separates muscles from the subcutaneous fat without prohibitively many elements. Specifically, our final mesh has 76,087 vertices and 375,187 tetrahedra and produces visually pleasing results even after significant growth (Figure 9, right).

We create the muscle envelope by progressively shrinking the skin mesh until it reaches the muscles or bones. To implement this skin shrinking process, we re-use some of the components of our muscle growth system (see Section 3.1). Specifically, we use our volumetric discretization conforming to the skin and we craft an elastic energy corresponding to uniform shrinking by setting each growth deformation gradient $\mathbf{F}_g = \beta\mathbf{I}$, where $\beta < 1$ is the shrinking parameter and $\mathbf{I} \in \mathbb{R}^{3\times 3}$ is the identity. Defining $\mathbf{F}_e = \mathbf{F}\mathbf{F}_g^{-1}$ as before, we use the corotated elastic energy (Equation (2)), augmented with a Tikhonov regularization term which guarantees a unique solution despite the translation and rotation invariance of the corotated elasticity. In our system, we use $w_{rot} = 100.0$ and $w_{reg} = 10.0$ and we minimize the elastic energy using Projective Dynamics (without any modifications as speed is not critical in this pre-processing phase). Once the quasi-static solution has been computed, we check whether any vertex of the shrunken skin collided with a muscle or a bone. If a collision has been detected, we constrain the offending vertex to the point of its first intersection and keep this constraint active for all subsequent shrinking iterations. We also experimented with sliding, but we found it is typically undesirable because we do not want the muscle envelope to deviate too much from the skin. At every iteration, we use $\beta = 0.9$ so that the muscle envelop
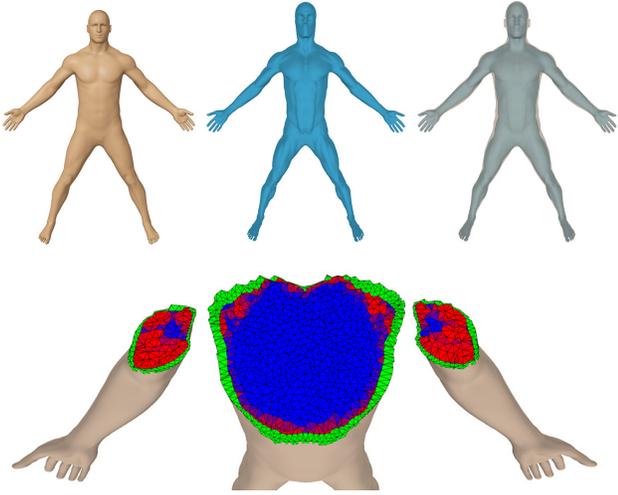
**Figure 10:** *Our "muscle envelope" (top-middle) is created by shrinking the input skin (top-left) until it reaches the muscles and the bones. We assume the space between the muscle envelope and the skin corresponds to subcutaneous fat (top-right). As a result, the envelope cleanly separates subcutaneous fat (green) from muscles (red) and generic soft tissues (blue).*

shrinks gradually. We stop the process after 10 iterations. The resulting muscle envelope is a well-behaved mesh tightly wrapping the muscles and bones; embedding it in our volumetric discretization creates a clean interface between subcutaneous fat and muscles (Figure 10).

### 3.3 Fat Growth

In this section we describe our model of fat hypertrophy. The first step is to segment out the volumetric region corresponding to subcutaneous fat. The muscle envelope discussed in the previous section facilitates this process, because we can simply select the tetrahedra which are outside of the muscle envelope. Note that our model grows only this region, corresponding to subcutaneous fat. Other types of adipose tissues are treated as generic soft tissues and are not subject to growth.

Inspired by the success of the shrinking process used to establish the muscle envelope, we initially tried to employ an analogous model for fat growth. Specifically, we set the growth deformation gradients corresponding to the fat tetrahedra to $\mathbf{F}_g = \beta\mathbf{I}$, with $\beta > 1$, and we find the quasi-static solution of a corotated elastic energy, as in Section 3.2. Even though this process successfully increases the volume of our subcutaneous fat region, the results do not look very realistic (see Figure 11 left). The reason is that the mechanical properties of fat are fundamentally different from muscles, because fat is very supple and exhibits semi-fluid behavior [Wolfram-Gabel et al. 1996]. Intuitively, when a subject is gaining weight, the fat fills any areas that are available in order to minimize stretching of the skin. The fact that subcutaneous fat "smooths out" muscle definitions is well-known to bodybuilders who strive to reduce it as much as possible.

We achieve the desired semi-fluid behavior of fat by a physics-based model inspired by the elastic behavior of pressured rubber balloons [Skouras et al. 2012]. Effectively, we treat the fat as fluid, but we introduce a regularization term preventing large displacements. Recall that in our system, we simulate growth without the presence of gravity and, therefore, the increased fat mass does not
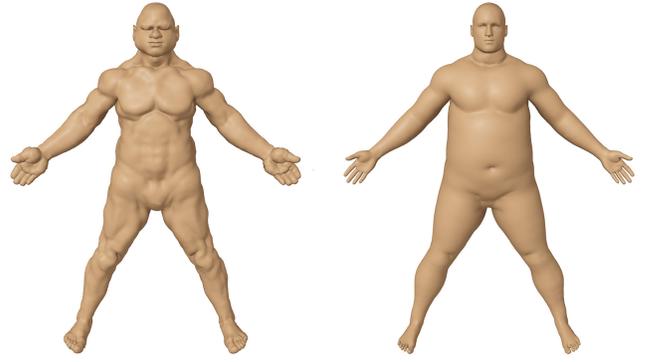


**Figure 11:** *Modeling fat as a solid elastic material does not yield the desired effect (left). Our system models fat as semi-fluid, achieving much more natural results (right).*

have any impact on the adjacent soft tissues. We can thus assume that during fat growth, the volume of the body under the muscle envelope is fixed. Our fat growth model is based on the fact that an increase of fat volume introduces pressure on the skin. The skin resists this pressure with a restoring force which prevents it from stretching too much. Specifically, given a fat growth parameter $\gamma > 0$, we define the pressure potential on vertex $\mathbf{x}_i, i = 1, \ldots, n$ as:

$$p_i(\mathbf{x}_i) = \gamma(\mathbf{x}_i - \tilde{\mathbf{x}}_i)^\top \tilde{\mathbf{n}}_i \quad (7)$$

where $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{n}}_i$ is the rest pose position and normal of vertex $i$. We use area-weighted normals. The stretching resistance of the skin is modeled using 2D corotated elasticity, applied to each skin triangle $t$:

$$E_{\text{skin},t}(\mathbf{x}) = \|\mathbf{F}_t - \mathcal{P}_{SO(2)}(\mathbf{F}_t)\|_F^2 \quad (8)$$

where $\mathbf{F}_t \in \mathbb{R}^{2 \times 2}$ is the in-plane deformation gradient. For each vertex we add a regularization term analogous to a zero rest-length spring, $\|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2$, which prevents each vertex from departing too far from its rest pose. We define our final energy as:

$$w_{\text{stretch}} \sum_t E_{\text{skin},t}(\mathbf{x}) + \sum_i (w_{\text{press}} p_i(\mathbf{x}_i) + w_{\text{reg}}\|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2) \quad (9)$$

where $t$ sums over all triangles and $i$ over all vertices. We solve for quasi-static equilibrium using (unmodified) Projective Dynamics, which gives us the positions of the skin vertices. Subsequently, we interpolate interior vertices of the subcutaneous fat region using harmonic coordinates [Joshi et al. 2007], treating the skin and the muscle envelope as boundary conditions. We found the following parameters work well: $w_{\text{stretch}} = 15.0$, $w_{\text{press}} = 1.0$, $w_{\text{reg}} = 10.0/\gamma$, producing very interesting *uniform* fat growth results.

To obtain more realistic *non-uniform* results, we need to account for the distribution of fat cells inside the human body, which varies largely between individuals and explains why different bodies have different preferential locations of fat storage, e.g., some people more easily gain fat in the abdomen, while others in the hip or chest areas. Common 3D anatomy models, including ours, correspond to lean individuals and lack information about the distribution of fat cells. Therefore, we add this information using a scalar function $f$ defined on the skin. For vertex $i$, $f_i \in [0, 1]$ tells us how much fat growth occurs near this vertex, which is, roughly speaking, proportional to the number of fat cells on a ray originating at $\tilde{\mathbf{x}}_i$ with direction $\tilde{\mathbf{n}}_i$. We use the value $f_i$ to modulate the pressure and regularization terms in Equation (9), effectively creating skin regions which are more or less prone to fat growth. In our system, the scalar function $f_i$ is designed by the user, who can use it to control
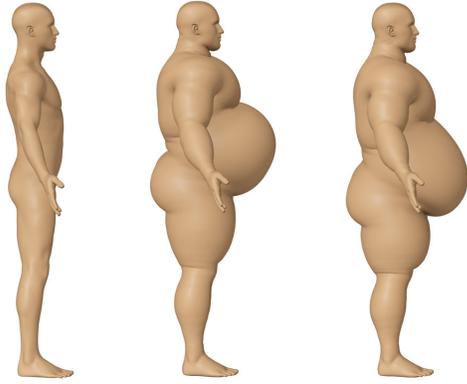
**Figure 12:** *Our system simulates fat growth without the presence of gravity (middle, the input model is on the left). We allow the user the preview the body after adding gravity (right).*



**Figure 14:** *Bone growth can be used to adjust the height of the body and the lengths of the limbs.*



**Figure 13:** *We change the shape of bones using bounded biharmonic weights with point handles placed on each bone head (a). Small bones use only one handle and are grouped together (b). The resulting deformations do not change the shape of the joints (c).*

the desired fat distribution. The scalar parameter $\gamma$ (Equation (7)) controls the *amount* of fat growth.

Even though this model is a gross simplification of real fat growth processes in the human body, it produces surprisingly realistic visual results, especially after we apply gravity in the motion-simulation phase, which allows the fat to sag in a natural way, see Figure 12. We also experimented with fat growth *with* gravity, adding gravitational potential to Equation (9). This approach is better justified physiologically, because human bodies naturally grow in a gravitational field. One complication is that we would ultimately have to cancel the effect of gravity before simulation [Chen et al. 2014], because simulation reintroduces gravity along with other effects such as inertia. We prefer to avoid these complications by decoupling the effects of growth and gravity.

## 3.4 Bone Growth

The process of bone growth is substantially different from soft-tissue growth, which we have discussed in the previous sections. While the volume of muscles and subcutaneous fat can change significantly during the life of an adult subject, bone growth occurs

only during childhood and adolescence. Even though simulating the growth of a child would be an extremely interesting research direction, the goal of this paper is a body shape modeling tool for computer graphics. Therefore, we base our bone growth model on geometric shape deformation techniques. We start by modeling the desired skeleton size and shape, ignoring the soft tissues. Subsequently, we fix the re-shaped skeleton and deform the adjacent soft tissues to obtain a full volumetric model.

We can use any shape deformation technique to edit the shape of the bones, such as applying an affine transformation to each bone [Ali-Hamadi et al. 2013]. However, to obtain more realistic and animatable results, we need to avoid non-uniform scaling of the bones, which would lead to unrealistically deformed bone heads. Also, we must preserve the structure of the joints, i.e., the bone heads meeting at a joint must be deformed in a coherent way, otherwise we may obtain abnormal joint shapes or even self-intersecting bones. We found that these constraints can be met by linear blend skinning, if we introduce point handles at each bone head, see Figure 13(b). To deform the bones, we define a transformation $\mathbf{T}_j$ at each handle $j$, consisting only of translation and uniform scaling. We transform each rest pose bone vertex $\tilde{\mathbf{v}}$ by linear blend skinning, i.e., $\mathbf{v} = \sum_j w_j \mathbf{T}_j \tilde{\mathbf{v}}$, where the skinning weights $w_j$ are computed using bounded biharmonic weights [Jacobson et al. 2011], see Figure 13(a). This way, we preserve the structure of the joints because the blending of transformations occurs only in the middle parts of the bones, see Figure 13(c). For extremities such as the hands and feet, we create only one point handle at each small bone, transforming the entire part uniformly. Even though we could use the same method to change the length of the fingers and toes, we focus on the whole body and defer detailed modeling of hand, feet, and the head to future work.

Having obtained the desired shape and size of the skeleton, we need to propagate these changes to the adjacent soft tissues. We achieve this by minimizing a corotated elastic energy (Equation (2)) subject to position constraints (zero rest-length springs) corresponding to the new bones. We use the same uniform sampling of the bones as in Section 3.1, where we set the target positions $\mathbf{t}_i$ in Equation (5) according to the new bones. We do not include any volume preservation terms, because the volume of the soft tissues is expected to
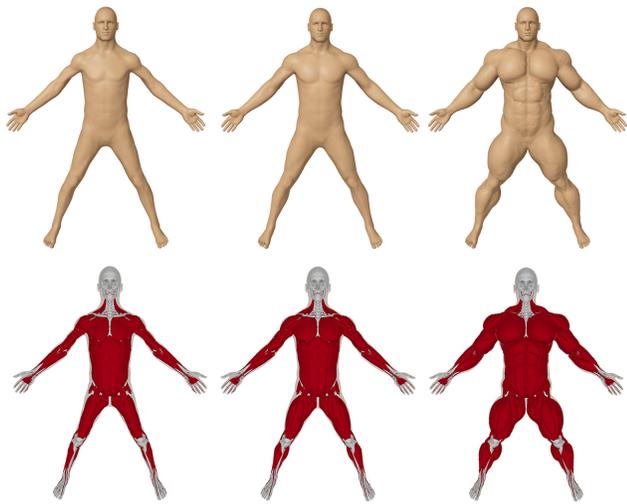
**Figure 15:** *Muscle growth can be used to simulate both atrophy (left) as well as hypertrophy (right). Compare with our input model in the middle.*
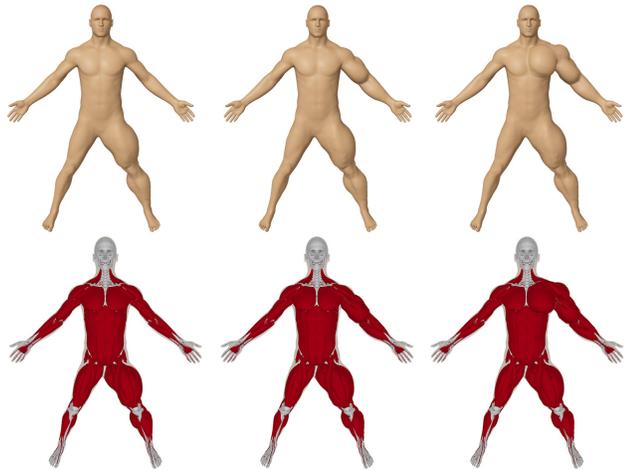


**Figure 16:** *The user can control the growth parameters of every muscle individually. Here, we grow only one part of the body to exaggerate this effect.*



**Figure 17:** *Our system allows the user to provide a function which specifies how much each part of the body is prone to fat accumulation (left), achieving different types of obese bodies (right).*

change. We minimize our energy using our modified Projective Dynamics (see Section 3.1), producing a new rest pose model which is ready for simulation of muscle and fat growth. Even though the corotated elastic energy does not guarantee non-inverted tetrahedra [Schüller et al. 2013], we did not obtain any self-intersections even when creating very tall or short subjects, see Figure 14.

## 4   Results

Our muscle growth model can simulate the effects of both hypertrophy and atrophy. This is controlled by changing the parameter $\alpha$ in Equation (1), see Figure 15. In all of our results, we produce not only the skin of the target character, but also its full volumetric deformations. To visualize the muscles, as in Figure 15, we barycentrically embed our polygonal muscle models in the tetrahedral mesh. An important practical consequence is that our resulting body shapes are ready for motion simulation using standard physics-based techniques [Lee et al. 2009].

Excessive parameter tuning is not necessary. In fact, we obtain visually plausible results even if we set all muscle growth parameters $\alpha$ to the same value. However, our system allows the user to control every muscle individually, which enables us to design almost arbitrary body shapes, see Figure 16. To achieve a reasonable compromise between flexibility and the number of parameters, in our experiments we divided the muscles into 12 groups, consisting of major muscle groups (e.g., quadriceps, abdomen, forearm) and important individual muscles (e.g., biceps and deltoid). Tuning the resulting 12 parameters is quite entertaining and even without any anatomy or bodybuilding expertise, the process is very fast. The longest example (Figure 15 right) took us about 5 minutes.

Fat hypertrophy is another important factor contributing to the shape of the human body. Because the distribution of fat varies dramatically between individuals, we allow the user to specify a scalar function on the surface of the body which specifies how much each body part is prone to fat accumulation. We show fat growth results
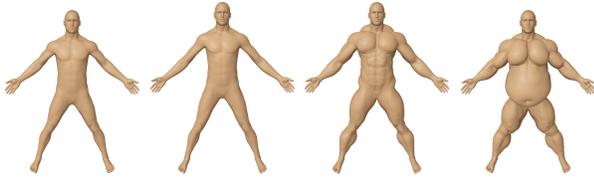
**Figure 18:** *In our pipeline, we first adjust the lengths of the bones and the adjacent soft tissues. Subsequently, we apply muscle growth and fat growth.*
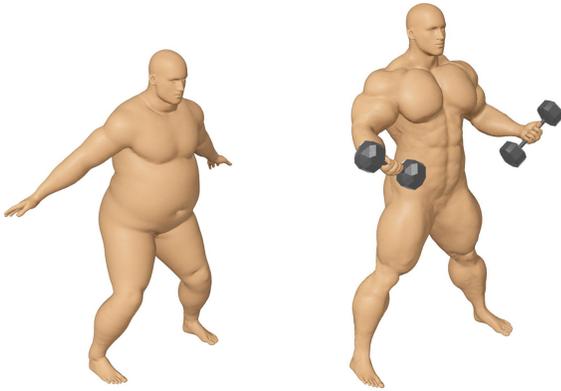


**Figure 19:** *Our system produces musculoskeletal models which are ready for simulation.*



**Figure 20:** *Applying the exact Hessian in our local/global solver (red curve) leads to only slightly faster convergence than our approximate but constant Hessian (blue curve). The latter avoids slow matrix factorization when changing growth parameters.*

for three different types of fat distributions in Figure 17. We created these functions with a simple 3D painting interface. The design of each fat body took us around 15 minutes, with a significant portion of the time spent in previewing the results. We did not experiment with fat atrophy because our input anatomy model is already quite lean.

Our physics-based growth models are combined with a simple yet effective geometric shape deformation technique to adjust the height of the skeleton and the lengths of the individual bones. We start by specifying global uniform scaling for the whole body, followed by non-uniform adjustments to the trunk and the skull. Subsequently, we manipulate the limbs using control points at each bone head, see Figure 13. We group all of the small bones in the hands and the feet and we control them as a whole. The deformations of the skeleton are immediately propagated to the whole body and the user previews the final result as shown in Figure 14. The entire bone modeling session typically takes no more than 5 minutes.

Our complete workflow is illustrated in Figure 18. First, the user adjusts the size of the skeleton and the lengths of the limbs. This produces a rest-pose ready for our physics-based simulation of muscle and fat growth. Our system first simulates muscle growth, followed by fat growth, and finally we add the effect of gravity to produce natural sagging of the fat (the effect of gravity on muscles is barely noticeable). We found that this decoupling also makes sense from the modeling perspective, because it is intuitive to design the human body one layer at a time, starting from the inside. We show that our resulting human body models are ready for physics-based simulation of motion. We experimentally verified this by simulating jumping of an obese subject and weight training of a muscular subject, see Figure 19 and the accompanying video. The main bottleneck of our run time is solving for quasi-static equilibrium using
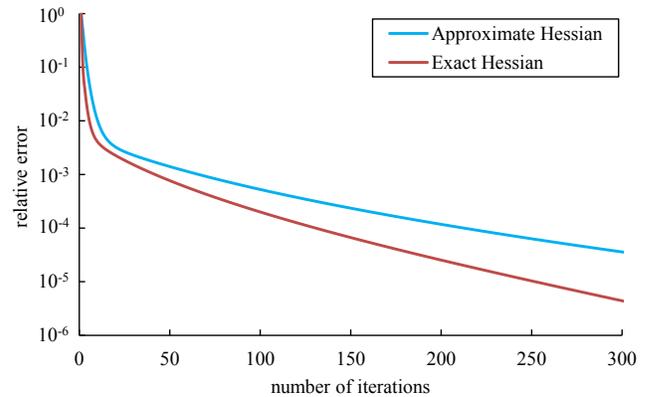
Projective Dynamics. In particular, its global step involves solving linear systems. However, our technique of approximating the true growth-dependent Hessian with a constant rest pose Hessian (see Section 3.1) helps significantly. Even though in theory the approximate Hessian provides inferior descent directions (analogously to quasi-Newton methods), we found that in our case the rest pose Hessian is an excellent approximation of the exact Hessian. This is demonstrated by our experiment in Figure 20, where we compare muscle growth simulation with our approximate Hessian and the exact one.

Sparse Cholesky factorization of the Hessian takes 5.51 seconds on a quad-core Intel Core i7 2.90GHz CPU using the Eigen library. However, this is done only once during initialization. At run time, each iteration of the local-global process takes 0.196 seconds on average – the local steps are quite fast even with our volume preservation terms. The user can preview intermediate results after every iteration. Typically, 10 iterations of the local/global process lead to sufficient visual quality.

The simulation of fat growth operates only on the subcutaneous fat region and therefore each iteration of the local-global process takes only 0.04 seconds. However, we use more iterations (typically 50) to accurately resolve the increase of volume of our obese body shapes. Even though our current prototype is missing many optimization opportunities, we found the performance sufficient to achieve near-interactive user experience. This makes the process of designing different body shapes fast and enjoyable. Please see Figure 1 for several examples of human body models created using our system.

## 5 Limitations and Future Work

Our models relies only on a crude simplification of the biological processes responsible for growth of living tissues. For example, when simulating obesity, we ignore the increase of volume of visceral fat and organs such as the liver. Our model of bone growth does not even attempt to be realistic, because in reality, the bone grows only from a small growth plate close to the bone head. In our system we chose to trade simplicity and fast computations for accuracy. We believe that the choice of our methods is well suited for applications in computer graphics.

Our work opens up many questions in the area of human body growth modeling. In our system, we rely only on a single anatom-

ical template with ad hoc material parameters. One avenue to improve both quality and accuracy of our models would be to use more data acquired e.g. using MRI scanning [Fan et al. 2014]. Combining our models with data-driven techniques would open a lot of possiblities. For example, statistical shape models of internal organs such as liver or kidneys could be incorporated to our models using example-based materials [Martin et al. 2011]. Conversely, our model could be used to generate input for data-driven body modeling methods. Although we focus on growth starting from our lean human body template, simulating weight loss of obese subjects would also be very interesting because skin typically does not shrink back. We did not develop detailed models for growth of the skull, hands, and feet. Growth modeling of the skull is a particularly important direction of future work because it could help us understand and treat pathological conditions such as craniosynostosis. Even though we experimented only with a human body model, we believe that our methods should be directly generalizable to other vertebrates and – with some modifications – to other animals. The main hurdle in these experiments is the availability of 3D anatomical models, even though recent methods such as Anatomy Transfer [Ali-Hamadi et al. 2013] aspire to alleviate this.

## 6 Conclusion

We presented a system to create a variety of human body shapes using a single anatomical model as input. To our knowledge, our work is the first to simulate physics-based growth processes of human tissues in computer graphics. We believe that our system will be instrumental in reducing the often prohibitive costs of human body modeling and will find applications even beyond the traditional realms of computer graphics, such as film, games, and visual effects. For example, we envision visualization applications useful in bodybuilding, ergonomic analysis, or to illustrate the adverse effects of obesity. At a higher level, we hope that our work will inspire new synergies between computer graphics and biomechanics.

## Acknowledgements

## References

AGUR, A. M., NG-THOW-HING, V., BALL, K. A., FIUME, E., AND MCKEE, N. H. 2003. Documentation and three-dimensional modelling of human soleus muscle architecture. *Clinical Anatomy 16*, 4, 285–293.

ALI-HAMADI, D., LIU, T., GILLES, B., KAVAN, L., FAURE, F., PALOMBI, O., AND CANI, M.-P. 2013. Anatomy transfer. *ACM Trans. Graph. 32*, 6, 188.

ALLEN, B., CURLESS, B., AND POPOVIĆ, Z. 2003. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graph. 22*, 3, 587–594.

ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., RODGERS, J., AND DAVIS, J. 2005. Scape: shape completion and animation of people. *ACM Trans. Graph. 24*, 3, 408–416.

BARGTEIL, A. W., WOJTAN, C., HODGINS, J. K., AND TURK, G. 2007. A finite element method for animating large viscoplastic flow. *ACM Trans. Graph. 26*, 3, 16.

BEN AMAR, M., AND GORIELY, A. 2005. Growth and instability in elastic tissues. *Journal of the Mechanics and Physics of Solids 53*, 10, 2284–2319.

BOTSCH, M., PAULY, M., WICKE, M., AND GROSS, M. 2007. Adaptive space deformations based on rigid cells. *Comput. Graph. Forum 26*, 3, 339–347.

BOUAZIZ, S., MARTIN, S., LIU, T., KAVAN, L., AND PAULY, M. 2014. Projective dynamics: fusing constraint projections for fast simulation. *ACM Trans. Graph. 33*, 4, 154.

BOYD, S., AND VANDENBERGHE, L. 2009. *Convex optimization*. Cambridge university press.

CHAO, I., PINKALL, U., SANAN, P., AND SCHRÖDER, P. 2010. A simple geometric model for elastic deformations. *ACM Trans. Graph. 29*, 4, 38.

CHEN, Y., LIU, Z., AND ZHANG, Z. 2013. Tensor-based human body modeling. In *Proc. CVPR*, 105–112.

CHEN, X., ZHENG, C., XU, W., AND ZHOU, K. 2014. An asymptotic numerical method for inverse elastic shape design. *ACM Trans. Graph. 33*, 4, 95.

CHOI, H. F., AND BLEMKER, S. S. 2013. Skeletal muscle fascicle arrangements can be reconstructed using a laplacian vector field simulation. *PloS one 8*, 10, e77576.

D'ANTONA, G., LANFRANCONI, F., PELLEGRINO, M. A., BROCCA, L., ADAMI, R., ROSSI, R., MORO, G., MIOTTI, D., CANEPARI, M., AND BOTTINELLI, R. 2006. Skeletal muscle hypertrophy and structure and function of skeletal muscle fibres in male body builders. *The Journal of physiology 570*, 3, 611–627.

FAN, Y., LITVEN, J., AND PAI, D. K. 2014. Active volumetric musculoskeletal systems. *ACM Trans. Graph. 33*, 4, 152.

FUNG, Y.-C. 1990. *Biomechanics: motion, flow, stress, and growth*, vol. 990. Springer-Verlag New York.

GEIJTENBEEK, T., VAN DE PANNE, M., AND VAN DER STAPPEN, A. F. 2013. Flexible muscle-based locomotion for bipedal creatures. *ACM Trans. Graph. 32*, 6, 206.

HASLER, N., STOLL, C., SUNKEL, M., ROSENHAHN, B., AND SEIDEL, H.-P. 2009. A statistical model of human pose and body shape. *Comput. Graph. Forum 28*, 2, 337–346.

JACOBSON, A., AND SORKINE, O. 2011. Stretchable and twistable bones for skeletal shape deformation. *ACM Trans. Graph. 30*, 6, 165.

JACOBSON, A., BARAN, I., POPOVIC, J., AND SORKINE, O. 2011. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph. 30*, 4, 78.

JAIN, A., THORMÄHLEN, T., SEIDEL, H.-P., AND THEOBALT, C. 2010. Moviereshape: Tracking and reshaping of humans in videos. *ACM Trans. Graph. 29*, 6, 148.

JONES, G. W., AND CHAPMAN, S. J. 2012. Modeling growth in biological materials. *SIAM Review 54*, 1, 52–118.

JOSHI, P., MEYER, M., DEROSE, T., GREEN, B., AND SANOCKI, T. 2007. Harmonic coordinates for character articulation. *ACM Trans. Graph. 26*, 3, 71.

KAVAN, L., AND ZARA, J. 2005. Spherical blend skinning: A real-time deformation of articulated models. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 9–16.

KRAEVOY, V., SHEFFER, A., SHAMIR, A., AND COHEN-OR, D. 2008. Non-homogeneous resizing of complex models. *ACM Trans. Graph. 27*, 5, 111.

LEE, S.-H., SIFAKIS, E., AND TERZOPOULOS, D. 2009. Comprehensive biomechanical modeling and simulation of the upper body. *ACM Trans. Graph. 28*, 4, 99.

LEE, D., GLUECK, M., KHAN, A., FIUME, E., AND JACKSON, K. 2010. A survey of modeling and simulation of skeletal muscle. *ACM Trans. Graph. 28*, 4.

LEE, Y., PARK, M. S., KWON, T., AND LEE, J. 2014. Locomotion control for many-muscle humanoids. *ACM Trans. Graph. 33*, 6, 218.

LEVIN, D. I., GILLES, B., MÄDLER, B., AND PAI, D. K. 2011. Extracting skeletal muscle fiber fields from noisy diffusion tensor data. *Medical Image Analysis 15*, 3, 340–353.

LOPER, M., MAHMOOD, N., AND BLACK, M. J. 2014. Mosh: motion and shape capture from sparse markers. *ACM Trans. Graph. 33*, 6, 220.

MARTIN, S., THOMASZEWSKI, B., GRINSPUN, E., AND GROSS, M. 2011. Example-based elastic materials. *ACM Trans. Graph. 30*, 4, 72.

MCADAMS, A., ZHU, Y., SELLE, A., EMPEY, M., TAMSTORF, R., TERAN, J., AND SIFAKIS, E. 2011. Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph. 30*, 4, 37.

NEUMANN, T., VARANASI, K., HASLER, N., WACKER, M., MAGNOR, M., AND THEOBALT, C. 2013. Capture and statistical modeling of arm-muscle deformations. *Comput. Graph. Forum 32*, 2pt3, 285–294.

NEUMANN, T., VARANASI, K., WENGER, S., WACKER, M., MAGNOR, M., AND THEOBALT, C. 2013. Sparse localized deformation components. *ACM Trans. Graph. 32*, 6, 179.

PAI, D. K., LEVIN, D. I. W., AND FAN, Y. 2014. Eulerian solids for soft tissue and more. In *ACM SIGGRAPH 2014 Courses*, 22:1–22:151.

PATTERSON, T., MITCHELL, N., AND SIFAKIS, E. 2012. Simulation of complex nonlinear elastic bodies using lattice deformers. *ACM Trans. Graph. 31*, 6, 197.

POPA, T., JULIUS, D., AND SHEFFER, A. 2006. Material-aware mesh deformations. In *Proc. of Shape Modeling International*, 22–22.

REINERT, B., RITSCHEL, T., AND SEIDEL, H.-P. 2012. Homunculus warping: Conveying importance using self-intersection-free non-homogeneous mesh deformation. *Comput. Graph. Forum 31*, 7, 2165–2171.

RODRIGUEZ, E. K., HOGER, A., AND MCCULLOCH, A. D. 1994. Stress-dependent finite growth in soft elastic tissues. *Journal of biomechanics 27*, 4, 455–467.

SCHEEPERS, F., PARENT, R. E., CARLSON, W. E., AND MAY, S. F. 1997. Anatomy-based modeling of the human musculature. In *Proc. SIGGRAPH*, 163–172.

SCHÜLLER, C., KAVAN, L., PANOZZO, D., AND SORKINE-HORNUNG, O. 2013. Locally injective mappings. *Comput. Graph. Forum 32*, 5, 125–135.

SEO, H., AND MAGNENAT-THALMANN, N. 2003. An automatic modeling of human bodies from sizing parameters. In *Proc. I3D*, 19–26.

SHOEMAKE, K., AND DUFF, T. 1992. Matrix animation and polar decomposition. In *Proc. Graphics interface*, vol. 92, 258–264.

SI, W., LEE, S.-H., SIFAKIS, E., AND TERZOPOULOS, D. 2015. Realistic biomechanical simulation and control of human swimming. *ACM Trans. Graph. 34*, 1, 10.

SI, H. 2011. A quality tetrahedral mesh generator and three-dimensional delaunay triangulator.

SIFAKIS, E., AND BARBIC, J. 2012. FEM simulation of 3D deformable solids: a practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, 20.

SIFAKIS, E., NEVEROV, I., AND FEDKIW, R. 2005. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph. 24*, 3, 417–425.

SKOURAS, M., THOMASZEWSKI, B., BICKEL, B., AND GROSS, M. 2012. Computational design of rubber balloons. *Comput. Graph. Forum 31*, 24.

SORKINE, O., COHEN-OR, D., LIPMAN, Y., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. 2004. Laplacian surface editing. In *Proc. SGP*, 175–184.

TABER, L. A. 1995. Biomechanics of growth, remodeling, and morphogenesis. *Applied mechanics reviews 48*, 8, 487–545.

TABER, L. A. 1998. Biomechanical growth laws for muscle tissue. *Journal of theoretical biology 193*, 2, 201–213.

TERAN, J., BLEMKER, S., HING, V., AND FEDKIW, R. 2003. Finite volume methods for the simulation of skeletal muscle. In *Proc. SCA*, 68–74.

TERAN, J., SIFAKIS, E., BLEMKER, S. S., NG-THOW-HING, V., LAU, C., AND FEDKIW, R. 2005. Creating and simulating skeletal muscle from the visible human data set. *IEEE Trans. Vis. Comput. Graphi. 11*, 3, 317–328.

TERAN, J., SIFAKIS, E., IRVING, G., AND FEDKIW, R. 2005. Robust quasistatic finite elements and flesh simulation. In *Proc. SCA*, 181–190.

THOMPSON, D. W. 1942. On growth and form.

WILHELMS, J., AND VAN GELDER, A. 1997. Anatomically based modeling. In *Proc. SIGGRAPH*, 173–180.

WISDOM, K., DELP, S., AND KUHL, E. 2015. Use it or lose it: multiscale skeletal muscle adaptation to mechanical stimuli. *Biomechanics and Modeling in Mechanobiology 14*, 2, 195–215.

WOLFRAM-GABEL, R., BEAUJEUX, R., FABRE, M., KEHRLI, P., DIETEMANN, J., AND BOURJAT, P. 1996. Histologic characteristics of posterior lumbar epidural fatty tissue. *Journal of neuroradiology 23*, 1, 19–25.

ZHOU, S., FU, H., LIU, L., COHEN-OR, D., AND HAN, X. 2010. Parametric reshaping of human bodies in images. *ACM Trans. Graph. 29*, 4, 126.