

# Data-Driven Physics for Human Soft Tissue Animation

MEEKYOUNG KIM, Korea Advanced Institute of Science and Technology

GERARD PONS-MOLL, Max Planck Institute for Intelligent Systems

SERGI PUJADES, Max Planck Institute for Intelligent Systems

SEUNGBAE BANG, Korea Advanced Institute of Science and Technology

JINWOOK KIM, Korea Institute of Science and Technology

MICHAEL J. BLACK, Max Planck Institute for Intelligent Systems

SUNG-HEE LEE, Korea Advanced Institute of Science and Technology

Data driven models of human poses and soft-tissue deformations can produce very realistic results, but they only model the visible surface of the human body and cannot create skin deformation due to interactions with the environment. Physical simulations can generalize to external forces, but their parameters are difficult to control. In this paper, we present a layered volumetric human body model learned from data. Our model is composed of a data-driven inner layer and a physics-based external layer. The inner layer is driven with a volumetric statistical body model (VSMPL). The soft tissue layer consists of a tetrahedral mesh that is driven using the finite element method (FEM). Model parameters, namely the segmentation of the body into layers and the soft tissue elasticity, are learned directly from 4D registrations of humans exhibiting soft tissue deformations. The learned two layer model is a realistic full-body avatar that generalizes to novel motions and external forces. Experiments show that the resulting avatars produce realistic results on held out sequences and react to external forces. Moreover, the model supports the retargeting of physical properties from one avatar when they share the same topology.

CCS Concepts: • **Computing methodologies** → **Physical simulation**;

Additional Key Words and Phrases: character animation, finite element method, statistical human shape, parameter estimation

## ACM Reference format:

Meekyoung Kim, Gerard Pons-Moll, Sergi Pujades, Seungbae Bang, Jinwook Kim, Michael J. Black, and Sung-Hee Lee. 2017. Data-Driven Physics for Human Soft Tissue Animation. *ACM Trans. Graph.* 36, 4, Article 54 (July 2017), 12 pages.

DOI: <http://dx.doi.org/10.1145/3072959.3073685>

## 1 INTRODUCTION

By learning the space of human shapes, recent statistical approaches have been able to successfully create realistic human shapes of various identities and poses. The capabilities of these approaches have evolved from only creating static poses [Anguelov et al. 2005] to synthesizing dynamic skin deformations, such as jiggling, due to

---

The first author performed part of this work during an internship at Max Planck Institute under the supervision of Gerard Pons-Moll.

This work was supported in part by the Seoul Olympic Sports Promotion Foundation, Ministry of Culture, Sports and Tourism, Republic of Korea.

Author's e-mail addresses: M. Kim, S. Bang, and S.-H. Lee, {koms1701, be2848, sunghee.lee}@kaist.ac.kr; G. Pons-Moll, S. Pujades, and M. J. Black, {gerard.pons.moll, spujades, black}@tue.mpg.de; J. Kim, jwkim@imrc.kist.re.kr.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2017 Copyright held by the owner/author(s). 0730-0301/2017/7-ART54 \$15.00

DOI: <http://dx.doi.org/10.1145/3072959.3073685>

vigorous pose changes [Pons-Moll et al. 2015]. Data driven methods are easy to control, are typically fast and produce very realistic animations. However, current data-driven methods can only produce geometric shape deformations driven by skeletal motions similar to those seen during training. In particular, they currently do not generalize to previously unseen deformations such as local deformations due to pushing and pulling of tissue, or to deformations due to changes in physical conditions (e.g., different gravity, under water, etc.).

The physics-based approach is a principled way to create realistic deformations under novel conditions. For soft tissue deformation, researchers have developed a number of methods based on finite elements methods (FEM) that solve continuum mechanics numerically for a discretized volumetric mesh representing a solid. For example, the nodes of the volumetric mesh corresponding to the bones move rigidly with pose changes and the movement of the remaining nodes are determined by solving the dynamics equation [Kim and Pollard 2011; Lee et al. 2009; Liu et al. 2013]. Physics-based simulation such as FEM has the advantage that it can create realistic deformations, not only due to pose changes, but also due to interactions with external objects. However, detailed simulation usually requires a high resolution mesh, which significantly increases the computational burden and the number of physical parameters; this makes it difficult to find the right parameters to create desirable deformation behaviors.

In this paper, we investigate a novel method that combines the advantages of data driven models with physics based methods. As an intuitive approach for this goal, we developed a layered model for character simulation with an inner layer driven by a statistical body model and an external layer driven by physics simulation. The parameters of both layers are learned from real scans of people and hence our approach is physics based and driven by data.

Since soft tissue deformations located deep in the body occur mostly due to pose changes they can be well represented by statistical approaches instead of physics simulation. Soft tissue deformation located closer to the surface occurs due to pose and also due to external forces applied on the body; such deformations can be better modelled using physics simulation.

Therefore, our strategy is to model a human character with two kinds of volumetric layers: a VSMPL layer, which is mostly located internally and deformed by a geometric, statistical model, and a FEM layer, which is located externally and deformed by FEM simulation. VSMPL is a volumetric extension of SMPL [Loper et al. 2015], a statistical human shape model that leverages the efficiency of linear



Fig. 1. Soft-tissue simulation of our layered model with physical parameters learned from captured data. The learned dynamic model generalizes not only to novel poses, but also to external forces as can be seen from a pulling example (right).

blend skinning (LBS). During animation the internal layer deforms with pose and provides a reliable internal shape that pushes and pulls the external FEM elements. Since VSMPL is incomparably faster than FEM simulation, we can also increase the computational performance significantly if a large portion of the internal nodes can be simulated by VSMPL without creating visual artifacts in the skin deformation. To define the layers, we developed a novel animator-friendly method: we define the relative distance field (RDF), which is a harmonic function that assigns a smoothly interpolated value between bones and skin surface values to each internal node. Consequently, the division of VSMPL and FEM layers can be controlled with the RDF values of the skin surface.

Our work also provides a method to estimate the parameters of a subject specific model from dynamic scans of the person. This includes the estimation of optimal layers and the estimation of physical parameters of the subject. Specifically, we devise an algorithm that iteratively estimates the layers and physical parameters that best reproduce the dynamic surface of the captured training mesh sequences from [Pons-Moll et al. 2015]. Figure 1 shows snapshots of our layered model that exhibits dynamic skin deformation similar to captured data and generates responsive soft tissue deformation due to external forces.

To summarize, the contributions are as follows: first, we introduce a two layer model consisting of a statistical volumetric body model and a FEM layer. As a second contribution, we show how the parameters of the layered model, namely layer segmentation and physical parameters can be estimated from captured dynamic scan sequences. Since the model has fewer finite elements it is computationally more efficient than pure physics based models. In addition, once learned, the model can be driven by pose and is reactive to external forces like pushing and pulling. In the experiments, we show that the model generalizes to novel pose sequences and produces realistic deformations due to external forces. We show deformations of the model due to poking, rotation and increased gravity values. While the deformations produced by our model are very realistic, animators might want to produce special effects. To that end, we present an efficient scheme to manually edit the layered model by controlling RDF values on the surface. This allows animators to exaggerate or attenuate the deformations.

## 2 RELATED WORK

Character deformation is a widely studied topic in computer graphics. Our work is closely related to the geometric skin deformation methods and the approaches for reducing the complexity of FEM simulation.

*Geometric Skin Deformation.* Closed form skinning techniques, such as LBS [Magnenat-Thalmann et al. 1988] or Dual Quaternion Skinning [Kavan et al. 2008] are widely used due to their computational efficiency and simplicity. In this approach, the skinning quality heavily depends on the skinning weights, and researchers have developed techniques for computing the weights either automatically [Baran and Popović 2007; Dionne and de Lasa 2013; Jacobson et al. 2012, 2011; Kavan and Sorkine 2012] or from examples [Wang and Phillips 2002]. However, closed form skinning techniques have difficulties in representing realistic deformation due to their simplistic mapping from a pose to the deformation. Recent studies have improved the deformation quality by allowing per-vertex center of rotation [Le and Hodgins 2016]. In contrast to the skinning methods, cage-based methods, such as Joshi et al. [2007] and Lipman et al. [2008], define shape deformation due to the control vertices defined outside a mesh.

Example-based skinning methods better reproduce the character's shapes by interpolating the observed or hand-sculpted examples as a function of the pose [Angelidis and Singh 2007; Kry et al. 2002; Lewis et al. 2000; Mohr and Gleicher 2003; Sloan et al. 2001]. James and Twigg [2005] developed a skinning method that extends to pseudo-articulated motions by automatically determining optimal bone transformations and vertex weight values. To express the space of human shape variation, methods based on principal component analysis have been developed [Allen et al. 2006; Seo et al. 2003]. SCAPE [Anguelov et al. 2005] represents the body shape and pose-dependent shape in terms of triangle deformations. Dense capturing of the human body allows for expressing the secondary movement of soft tissues. Park and Hodgins [2006; 2008] captured the dynamics of a subject with hundreds of markers and modeled soft tissue motions. The Dyna model [Pons-Moll et al. 2015] extends the SCAPE model to approximate dynamic surface deformations of soft tissue. SMPL [Loper et al. 2015] learns the pose and identity-dependent

blendshapes and LBS weights. Despite their realistic results, and due to the nature of the data-driven method, example-based methods cannot create skin deformation due to physical interactions with the environment.

*Reduced Physics Simulation.* Physics simulation of the deformable bodies is capable of creating realistic secondary effects and interacting physically with the environment [Si et al. 2014], but the computational cost is generally too high for interactive applications. Many techniques have been proposed to overcome this problem [McAdams et al. 2011].

Larboulette et al. [2005] used a simple blend between static skinning and dynamic deformation, and Mukai and Kuriyama [2016] added helper bones that add dynamic behaviors to LBS. The dimensionality reduction is an effective and principled approach for efficient simulation of deformable bodies. Since Pentland and Williams [1989] pioneered the use of reduced deformable models using modal analysis, this approach has seen continued development [An et al. 2008; Barbič and James 2005; James and Pai 2002; Li et al. 2014; Xu and Barbič 2016]. The reduction of the deformation dimensions can be performed efficiently for a non-homogeneous object by dividing the object into multiple domains, with each domain reduced differently per their deformation characteristics [Kim and James 2011; Wu et al. 2015; Yang et al. 2013]. Gao et al. [2014] developed a method to simulate only the skin surface while obtaining a similar effect as the corotational volumetric elasticity. Hahn et al. [2012] reduced the soft tissue dynamics to the rig space. Our work gains speed by removing some of the internal nodes that will instead have their movement fully constrained by joint motions through VSMPL.

Many studies have modeled human anatomy in multiple layers. Lee et al. [1995] used a mass spring lattice with three layers for facial animation. Recently, multi-layer systems have been proposed to represent the physical properties of each anatomic layer, such as muscle, fat, and skin [Deul and Bender 2013; Iwamoto et al. 2015]. The purpose of these multi-layer approaches is to enhance the simulation quality by modeling the human anatomy at a higher resolution whereas our work has a largely opposite goal to simplify the simulation by using the predictive power of statistical methods. Thus, our layer model does not respect the anatomical structure.

In physics-based deformation simulation, researchers have developed many methods to couple multiple simulation units, such as combining heterogeneous materials and simulating the material using overlapping meshes with different resolutions. The readers can refer to Malgat et al. [2015] for a detailed review of these approaches. In contrast to these approaches, we are interested in the creation of a practical heterogeneous simulation system for a human character.

*Material Property Estimation.* Dynamic effects of materials are mainly governed by their physical properties. Recently, there is a growing interest in the data-driven learning of physical parameters by capturing the dynamic behavior of real materials in the computer graphics field. Miguel et al. [2012] directly optimized a captured cloth with nonlinear stress-strain curves based on measurements with force and deformed configurations. Wang et al. [2015] captured deformation with an automatically calculated rest pose and force displacement. They also introduced a reduced order material

optimization with just a few control points. Xu et al. [2015] solved material properties in low dimensions with a model reduction technique.

Recent studies have also focused on constructing subject-specific models for some [Murai et al. 2016; Zhu et al. 2015] or whole [Ali-Hamadi et al. 2013] anatomical structures. The most similar work to ours is Kadleček et al. [2016], who constructed anatomical models for physical simulation using a set of 3D surface data. Their work is impressive in that a plausible physics-based anatomical model was obtained after a large scale optimization. Our work is different in that our method can create realistic soft tissue deformation under highly dynamic movements whereas the simulation results reported in Kadleček et al. [2016] are mostly quasi-static. Kozlov et al. [2017] define a layered model of the head with underlying rigid components covered by novel blendvolumes. They manually define the physical parameters of these blendvolumes to add dynamics to standard blendshape facial animation.

### 3 SURFACE BODY MODEL

SMPL [Loper et al. 2015] is a body surface model that uses a learned rigged template  $\bar{S}_\mu$  with  $N = 6890$  vertices. The vertex positions of SMPL are adapted based on the identity-dependent shape parameters and the skeleton pose. The skeletal structure of the human body is modeled with a kinematic chain consisting of rigid bone segments linked by  $K = 24$  joints. Each joint is modeled as a ball joint with 3 rotational degrees of freedom (DoF), parameterized with exponential coordinates, see [Pons-Moll and Rosenhahn 2011] for more details on kinematic chains and exponential maps. Including translation, the pose  $\theta$  is determined by a pose vector of  $3 \times 24 + 3 = 75$  parameters.

To model the shape and pose dependent deformations, SMPL modifies the template in an additive way and predicts the joint locations from the deformed template. The surface body model  $S(\beta, \theta)$  is then

$$S(\beta, \theta) = W(\bar{S}(\beta, \theta), J(\beta), \theta, \mathbf{W}) \quad (1)$$

$$\bar{S}(\beta, \theta) = \bar{S}_\mu + S_s(\beta) + S_p(\theta) \quad (2)$$

where  $W(\bar{S}, J, \theta, \mathbf{W}) : \mathbb{R}^{3N} \times \mathbb{R}^{3K} \times \mathbb{R}^{|\theta|} \times \mathbb{R}^{N \times K} \mapsto \mathbb{R}^{3N}$  is a linear blend skinning function [Magnenat-Thalmann et al. 1988] that takes skin vertices in the rest pose  $\bar{S}$ , joint locations  $J$ , a pose  $\theta$ , and the blend weights  $\mathbf{W}$ , and returns the posed skin vertices. The parameters  $\beta$  control the shape of the template mesh. The learned functions  $S_s(\beta)$  and  $S_p(\theta)$  output vectors of vertex offsets from the template. We refer to these as shape and pose blend shapes respectively. Pose blend shapes deform the shape in a rest pose to compensate for skinning artifacts and to model pose dependent deformations such as muscle bulging. Shape blend shapes model changes due to identity. Both functions are linear combinations of learned blend shapes. See Loper et al. [2015] for more details.

### 4 LAYERED MODEL

Our proposed layered model is a volumetric mesh consisting of two layers: the VSMPL layer, which is driven by a learned statistical body model, and a FEM layer that is driven using physics based simulation. Statistical body models such as SMPL [Loper et al. 2015]

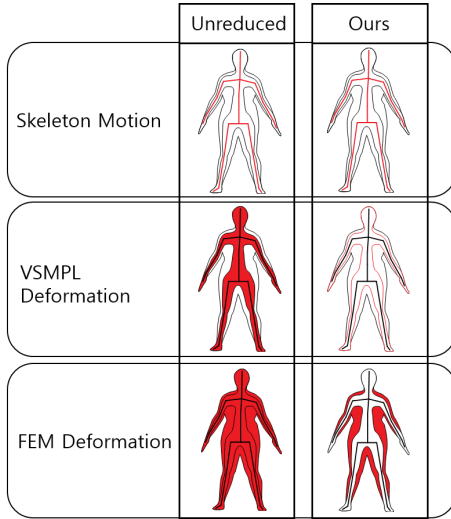


Fig. 2. Overall animation process. Left: unreduced model. Right: our model. Red lines and areas are where actual computation is performed.

only model deformation at the surface of the body. Hence, we first adapt SMPL to work with volumetric meshes; we call the resulting model VSMPL (*Volumetric SMPL*). Secondary motion effects are achieved using the FEM layer. The segmentation of the volumetric mesh into VSMPL and FEM layers and the physical parameters of the FEM layer are optimized directly using 4D dynamic scans of real people and described in Section 5.

*Volumetric mesh.* Using the SMPL surface template  $\bar{S}$  and its skeletal structure we create a tetrahedral mesh with nodes  $\mathbf{X}$  ( $|\mathbf{X}| = M$ ) using the Tetgen software package [Si 2015]. Let  $\mathbf{B}_j$  ( $j = 1, \dots, K$ ) denote a set of vertices distributed in equidistant intervals in the  $j$ -th bone (a line segment). Then a tetrahedral mesh is constructed so that its nodes  $\mathbf{X}$  include  $\mathbf{B} = \mathbf{B}_1 \cup \dots \cup \mathbf{B}_K$ , and the external boundary of the tetrahedral mesh coincides with the SMPL surface mesh  $\mathbf{S}$ . This operation is done only once for the template mesh that is watertight and has no self-intersections. Novel volumetric shapes are obtained by extending the SMPL function to volumetric meshes, as shown in Section 4.1. For a given subject, each node in  $\mathbf{X}$  is then assigned to one simulation method, either VSMPL or FEM. The nodes that move near-rigidly with the neighboring bones are simulated by VSMPL, and hence are called VSMPL nodes  $\mathbf{X}^V \subset \mathbf{X}$ . The other nodes that should express dynamic responses against vigorous body movements and external forces are simulated by FEM, and called FEM nodes  $\mathbf{X}^F \subset \mathbf{X}$ . We additionally include into  $\mathbf{X}^F$  the VSMPL nodes that are connected to the FEM nodes by edges. Therefore,  $\mathbf{X}^V \cap \mathbf{X}^F$  are the set of nodes on the boundary between the VSMPL and FEM layers, and  $\mathbf{X} = \mathbf{X}^V \cup \mathbf{X}^F$ . Section 4.3 explains how we automatically label each node. For later use, we will also define  $\mathbf{X}^S \subset \mathbf{X}$  as the surface nodes.

Given a new pose, VSMPL updates the positions of  $\mathbf{X}^V$  (Section 4.1), which then act as constraints that generate internal forces due to the stress-strain relationship for FEM simulation. The positions of  $\mathbf{X}^F$  are updated by the FEM simulation and time integration.

Figure 2 (left) shows these overall steps of soft tissue deformation. Note that if the whole volumetric mesh is fed to the FEM simulator with  $\mathbf{X}^V$  given as constraints, the performance gain due to the layered model is not satisfying. Thus, part of the volumetric mesh needs to be removed, as will be discussed in Section 4.2.

#### 4.1 VSMPL: Volumetric - SMPL

The SMPL model is a function that takes as input a set of shape and pose parameters and outputs a set of deformed  $N$  surface vertices  $\mathbf{X}^S$ , i.e.,  $S(\boldsymbol{\beta}, \boldsymbol{\theta}) : \mathbb{R}^{|\boldsymbol{\beta}| \times |\boldsymbol{\theta}|} \mapsto \mathbb{R}^{3N}$ . VSMPL is an extension of SMPL to volumetric meshes. Consequently, VSMPL takes as input a shape and pose parameters  $X(\boldsymbol{\beta}, \boldsymbol{\theta}) : \mathbb{R}^{|\boldsymbol{\beta}| \times |\boldsymbol{\theta}|} \mapsto \mathbb{R}^{3M}$ , and outputs a set of deformed nodes in the volumetric mesh. A naive adaptation of SMPL to volumetric meshes would use SMPL to construct a deformed surface and then apply Tetgen to obtain the volumetric mesh. This, however, is impractical and problematic. Novel shapes typically contain self-intersections, which make Tetgen fail. Furthermore, a model with a consistent volumetric topology is desirable to make animation, learning, and retargeting easy.

Hence, for a given shape and pose  $\boldsymbol{\beta}, \boldsymbol{\theta}$  we interpolate the SMPL surface function to the entire volumetric mesh using variants of the Laplace equation. Without loss of generality, let  $F : \mathbf{X}^S \mapsto \mathbb{R}$  be a function returning a scalar value on the mesh surface nodes  $\mathbf{x}_i \in \mathbf{X}^S$ . Then, we find an interpolant function  $G : \mathbf{X} \mapsto \mathbb{R}$  at the rest of the nodes  $\mathbf{x}_i \in \mathbf{X}$  by solving the following equation:

$$\Delta G(\mathbf{x}_i) = 0, \quad \mathbf{x}_i \in \mathbf{X} \quad \text{subject to} \quad (3)$$

$$G(\mathbf{x}_i |_{\mathbf{B}}) = 0, \quad G(\mathbf{x}_i |_{\mathbf{X}^S}) = F(\mathbf{x}_i) \quad (4)$$

that forces interpolation using the boundary conditions at the surface  $G(\mathbf{x}_i |_{\mathbf{X}^S}) = F(\mathbf{x}_i)$  and at the bone nodes  $G(\mathbf{x}_i |_{\mathbf{B}}) = 0$ . We denote the interpolant as  $\mathcal{L}$ . We use variants of Eq. (3) to interpolate the pose and shape blend shape functions. To interpolate blendweights, we use bounded biharmonic weights (BBW) [Jacobson et al. 2011], which minimize the Laplacian energy with additional constraints.

*Volumetric pose blend shapes.* The pose blend shape function takes a pose as input and outputs a vector field of 3D offsets at surface vertices. The function is a linear combination of blend shapes  $\mathbf{P}_n$ , where the weights are a function of rotation matrix elements  $R_n(\boldsymbol{\theta})$ , see [Loper et al. 2015]:

$$S_p(\boldsymbol{\theta}) = \sum_{n=1}^{9(K-1)} (R_n(\boldsymbol{\theta}) - \mathbf{I}) \mathbf{P}_n. \quad (5)$$

To obtain the volumetric analog, one could apply Laplacian interpolation to the output field for every new pose. However, that would be computationally too expensive. Hence, we approximate it as

$$\begin{aligned} X_p(\boldsymbol{\theta}) = \mathcal{L}(S_p(\boldsymbol{\theta})) &= \mathcal{L} \left( \sum_{n=1}^{9(K-1)} (R_n(\boldsymbol{\theta}) - \mathbf{I}) \mathbf{P}_n \right) & (6) \\ &\approx \sum_{n=1}^{9K} (R_n(\boldsymbol{\theta}) - \mathbf{I}) \mathcal{L}(\mathbf{P}_n). & (7) \end{aligned}$$

We are effectively interpolating every pose blend shape to obtain volumetric pose blend shapes  $\mathbf{P}_{V,n} = \mathcal{L}(\mathbf{P}_n)$ . If each of the volumetric pose blend shapes  $\mathbf{P}_{V,n}$  satisfies the boundary conditions and null

Laplacian everywhere (Eq. (3)), it is easy to show that the resulting linear combination  $X_p(\theta)$  also satisfies the boundary conditions and has zero Laplacian everywhere. An interpolant with zero Laplacian everywhere might not be possible to find and hence Eq. (7) is an approximation. More importantly, we can obtain smooth interpolants that satisfy the boundary conditions at the surface. In this way, we can precompute a set of volumetric pose blend shapes only once and use them for animation as it is done with the SMPL model.

*Volumetric shape blend shapes.* To obtain a volumetric mesh for a novel surface shape  $S_{\text{new}}$  we deform the template volumetric mesh  $X$ . However, here we do not impose the interpolant  $\mathcal{L}$  to vanish at bone locations because the changes in shape also affect the bone structure. Instead, we force the interpolant to match the joints of the new shape  $G(\mathbf{x}_i|_B) = J_{\text{new}}$ . As before, we also have the surface boundary condition  $G(\mathbf{x}_i|_{X^S}) = S_{\text{new}}$ . This way we obtain a novel volumetric mesh  $X_{\text{new}}$ . One can apply this strategy to each shape blend shape to obtain a volumetric shape blend shape function  $X_s(\beta)$ . To obtain more detail, novel shapes can directly be represented as a personalized surface  $S_{\text{new}}$  and not with the shape coefficients. In this case, SMPL Eq. (2) reduces to  $\tilde{S}(\theta) = S_{\text{new}} + S_p(\theta)$  and VSMPL to  $\tilde{X}(\theta) = X_{\text{new}} + X_p(\theta)$ .

*Volumetric blendweights.* After applying the shape and pose blend shapes, SMPL transforms a point on the surface using linear blend skinning (LBS). Specifically, SMPL determines the position of the skin vertices  $S(\beta, \theta)$  using a linear combination bone transformation matrices  $G_j$  with weights  $w_{ij}$ . The weight  $w_{ij}$  controls the importance of the  $j$ -th bone to the  $i$ -th vertex  $\mathbf{x}_i$

$$\mathbf{x}_i = \sum_{j=1}^K w_{ij} G_j \bar{G}_j^{-1} \bar{\mathbf{x}}_i, \quad \bar{\mathbf{x}}_i \in \tilde{S}(\beta, \theta), \quad (8)$$

where  $\bar{\mathbf{x}}_i$  and  $\mathbf{x}_i$  are the rest and deformed positions of the  $i$ -th vertex, respectively, and  $\bar{G}_j$  is the transformation of the  $j$ -th bone in the rest pose.

While LBS is generally applied to the surface vertices, VSMPL uses it to deform the volumetric mesh; i.e., we extend  $\tilde{S}(\beta, \theta)$  to  $\tilde{X}(\beta, \theta)$  in Eq. (8). Thus the blendweights  $\mathbf{W}$  of SMPL are extended to the volume, denoted by  $\mathbf{W}_V$ . For this, we use bounded biharmonic weights (BBW) [Jacobson et al. 2011] for the nodes in the volumetric mesh. Specifically, we find the function  $w_j : \Omega \rightarrow \mathbb{R}$  ( $j = 1, \dots, K$ ), where  $\Omega$  is a volumetric domain that includes node  $X$  and  $w_{ij} = w_j(\mathbf{x}_i)$  is an element of  $\mathbf{W}_V$ , so as to satisfy the following conditions:

$$\begin{aligned} & \arg \min_{w_j, j=1, \dots, K} \sum_{j=1}^K \frac{1}{2} \int_{\Omega} \|\Delta w_j\|^2 dV \quad \text{subject to} \\ & \sum_{j=1}^K w_j(\mathbf{x}_i) = 1, \quad w_j|_{B_k} = \delta_{jk}, \quad 0 \leq w_j(\mathbf{x}_i) \leq 1, \end{aligned} \quad (9)$$

where  $\delta_{jk} = 1$  if  $j = k$ , and 0 otherwise. We add an additional constraint to set surface weights to the original SMPL blendweights; i.e.,  $w|_S = \mathbf{W}$ .

*Final model.* VSMPL deforms a volumetric mesh using the volumetric pose and shape blend shape functions  $X_p(\theta)$ ,  $X_s(\beta)$  and

volumetric blendweights  $\mathbf{W}_V$

$$X(\beta, \theta) = W(\tilde{X}(\beta, \theta), J(\beta), \theta, \mathbf{W}_V) \quad (10)$$

$$\tilde{X}(\beta, \theta) = \tilde{X}_\mu + X_s(\beta) + X_p(\theta). \quad (11)$$

Note that VSMPL is analogous to the original SMPL model from Eq. (1) and is therefore also linear, easy to use, and can be animated.

## 4.2 Layer for FEM simulation

We employ the co-rotated linear model [Müller and Gross 2004] that retains the benefit of the computational efficiency of the linear model while alleviating the artifact due to large rotation. For fast simulation, we use the implicit Euler time integration scheme that allows for a larger time step. The dynamics equation of the FEM layer is

$$\mathbf{M}\ddot{\mathbf{x}} + c\dot{\mathbf{x}} + \mathbf{K}(\mathbf{x} - \bar{\mathbf{x}}) = \mathbf{f}_{\text{ext}} \quad (12)$$

where  $\mathbf{x} \in \mathbb{R}^{3|X^F|}$  is the vector of the positions of FEM nodes  $X^F$  and  $\bar{\mathbf{x}}$  is their rest position.  $\mathbf{M}$  and  $c$  denote the mass matrix and damping parameter, respectively. We simply employ the artificial damping that reduces all nodal velocities by a constant factor  $c$ . The vector  $\mathbf{f}_{\text{ext}}$  is the external force term and  $\mathbf{K}(\mathbf{x} - \bar{\mathbf{x}})$  is the internal force due to the elasticity of the material. In the co-rotational linear model, the element-wise internal force is modeled as  $-\mathbf{R}_e \mathbf{K}_e (\mathbf{R}_e^T \mathbf{x}_i - \bar{\mathbf{x}}_i)$ , where  $\mathbf{R}_e$  is the rotational component in the deformation of the element  $e$ . We construct  $\mathbf{K}' = \sum_e \mathbf{R}_e \mathbf{K}_e \mathbf{R}_e^T$  by assembling the internal forces of every element, where  $\sum_e$  denotes the addition of the element-wise stiffness terms to the slots in the matrix corresponding to the associated node. Combined with the implicit time integration, Eq. (12) turns into:

$$(\mathbf{M} + c\Delta t \mathbf{I} + \Delta t^2 \mathbf{K}') \mathbf{v}^{t+1} = \mathbf{M} \mathbf{v}^t - \Delta t (\mathbf{K}' \mathbf{x}^t - \mathbf{f}_0 - \mathbf{f}_{\text{ext}}) \quad (13)$$

where  $\mathbf{v}^{t+1}$  is the velocity vector at the next time step  $t + 1$  and  $\mathbf{f}_0 = \sum_e \mathbf{R}_e \mathbf{K}_e \bar{\mathbf{x}}$ . The nodal positions  $\mathbf{x}^{t+1}$  are obtained by integrating  $\mathbf{v}^{t+1}$  in time. We use the conjugate gradient method to solve Eq. (13). We refer the readers to Müller and Gross [2004] regarding the method for computing  $\mathbf{K}_e$ .

*Efficient Computation.* We increased the computational speed to compute  $\mathbf{K}'$  and  $\mathbf{f}_0$  by parallelizing the assembly processes by dividing elements into clusters that can be processed concurrently. The dynamics equations, Eq. (12) and Eq. (13) are constructed to cover only the nodes in  $X^F$ , as shown in Figure 2 (right). In addition, among the nodes in VSMPL, only the nodes at the intersection  $X^V \cap X^F$  need to be updated by VSMPL as they define boundary condition to the FEM nodes. Therefore, we remove the internal nodes  $X^V - X^F$  from VSMPL to further simplify the simulation process, as shown in our model of Fig. 2.

## 4.3 Segmentation into Layers with Relative Distance Field

Now we explain how to label each node as VSMPL or FEM. Since the total number of labeling cases is  $2^{|X|}$ , a brute force approach to find the optimal labels by testing all possible cases is not feasible. A reasonable approach would be to label the nodes close to the bones as VSMPL and those close to the skin as FEM; to that end we need a suitable method to measure the distance to the bone or skin. Our first trial was to contract the skin mesh towards the bones to obtain a boundary between the VSMPL and FEM nodes. We tried



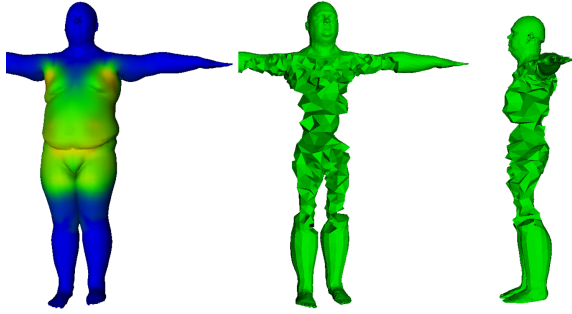


Fig. 3. Color map of RDF values (right) and the corresponding VSMPL layer.

several methods such as contracting the skin mesh along its normal direction or to its nearest bone and applying a skeleton extraction method [Au et al. 2008], but none have produced satisfactory results, especially in regions close to multiple bones.

Our proposed solution is to define the relative distance field (RDF)  $\mathbf{d} : \mathbf{X} \rightarrow \mathbb{R}$  that assigns to each node a smoothly interpolating value from the bones to the skin, and set a threshold  $\sigma$  (set to 1 in our work) that divides nodes into either VSMPL or FEM. Nodes with RDF values less than  $\sigma$  are labeled as VSMPL, otherwise as FEM. To this end, we again use the Laplacian interpolation scheme Eq. (3)

$$\Delta \mathbf{d}(\mathbf{x}_i) = 0, \quad \mathbf{x}_i \in \mathbf{X} \quad \text{subject to} \quad (14)$$

$$\mathbf{d}(\mathbf{x}_i | \mathbf{B}) = 0, \quad \mathbf{d}(\mathbf{x}_i | \mathbf{X}^s) = d_i. \quad (15)$$

In this way, the segmentation into layers is controlled by the values  $d_i$  at skin nodes. A skin region with  $d_i$  values set much higher than  $\sigma$  has a thick FEM layer, and thus can produce a jiggling effect on the abdomen and buttocks of an obese person. In contrast, a skin region with  $d_i$  values close to  $\sigma$  has a thin FEM layer, suitable for modeling rigid parts or a thin person.

Figure 3 shows the RDF values of every node colored with color temperatures from blue ( $\sigma + 0$ ) to red ( $\sigma + 1$ ), and the corresponding VSMPL layer. The RDF method greatly reduces the complexity of the problem, but still has the capability to express the different dynamic effects per body part. The coarseness of the tetrahedral mesh results in a somewhat jagged FEM domain. Since our goal is to produce a plausible surface simulation that can be generalized to external forces, and not to have a realistic model of the internal structure, the resulting FEM domain obtains enough visual quality for the simulations. Note that our method obtains a smoother FEM domain for finer tetrahedral meshes, as shown in Figure 4. Next, we explain how to obtain the optimal RDF values for the skin vertices.

## 5 PARAMETER OPTIMIZATION

The resulting skin deformation depends on the true rest shape  $\bar{\mathbf{x}}$  (unaffected by gravity), the segmentation of VSMPL and FEM layers (controlled by the RDF values  $\mathbf{d}$  on the skin surface), and the physical parameters for the FEM layer, including the mass  $\mathbf{M}$ , damping  $c$ , the Young's modulus  $E$  and the Poisson's ratio  $\nu$  that determine  $\mathbf{K}$  in Eq. (12). Notice that the captured rest shape, denoted as  $\bar{\mathbf{x}}_s$  in this paper, is affected by gravity. We set the mass of each node such that every element has equal density. We then estimate the rest shape  $\bar{\mathbf{x}}$ , RDF value  $\mathbf{d}$  of each skin node, and physical parameters  $\mathbf{p} = (c, E, \nu)$

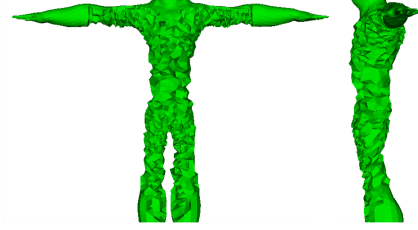


Fig. 4. VSMPL layer obtained for the higher resolution mesh (3.5 times more nodes and 5.4 times more elements).

of each node or element from the captured sequences. Pons-Moll et al. [2015] provide the Dyna registrations of the 4D dynamic sequences. The dataset captures the skin deformation in detail for various human motions, including vigorous dynamic movements such as running, and thus provides a good test case for our experiments on parameter estimation.

Let  $X_i = \{\mathbf{x}_i^f | f = 1, \dots, F\}$  denote the sequence of the  $i$ -th node  $\mathbf{x}_i$  from frame 1 to  $F$ . Sequences  $\hat{X}_i$  and  $X_i(\mathbf{d}, \bar{\mathbf{x}}, \mathbf{p})$  are the captured and simulated sequences given by  $\mathbf{d}$ ,  $\bar{\mathbf{x}}$ , and  $\mathbf{p}$ , respectively. Our purpose is to estimate the parameters that bring  $X_i$  as close as possible to  $\hat{X}_i$ . Thus we define the loss function of  $X_i$  given  $\hat{X}_i$ .

### 5.1 Loss Function: Mean Amplitude Deviation

The physical parameters we aim to estimate are closely related to the dynamic soft-tissue movement of nodes *relative* to the skeleton. In particular, the characteristics of the parameters are clearly observed in the form of vibrational motion with respect to the skeleton when a human performs repetitive dynamic movements such as running. To find such motion relative to the skeleton, we first bring all nodes to a canonical T-pose, we call this step *unposing* which we explain in Sec. 5.1.1. This effectively isolates the motion due to soft-tissue dynamics. Then we compare the node trajectories of the simulation with those of the captured data. To be more robust to phase misalignments we use the average distance of amplitude values as error metric, Sec. 5.1.2.

**5.1.1 Skeletal Motion Extraction and Unposing.** To unpose a sequence of captured nodes  $\hat{\mathbf{x}}_i^f$  we need to estimate the pose at every frame and then invert the SMPL generative process. Adding soft-tissue motion to the SMPL equation (2), the position of a captured node  $\hat{\mathbf{x}}_i$  can be written as

$$\hat{\mathbf{x}}_i^f = \sum_{j=1}^K w_{ij} \mathbf{G}_j(\boldsymbol{\theta}^f) \bar{\mathbf{G}}_j^{-1} (\bar{\mathbf{x}}_i + \mathbf{b}_i(\boldsymbol{\beta}) + \mathbf{b}_i(\boldsymbol{\theta}^f) + \boldsymbol{\xi}^f) \quad (16)$$

where  $\mathbf{b}_i(\boldsymbol{\beta})$  is the shape blendshape offset for  $\bar{\mathbf{x}}_i$ ,  $\mathbf{b}_i(\boldsymbol{\theta}^f)$  is the pose blendshape offset at frame  $f$ , and  $\boldsymbol{\xi}^f$  is the offset due to dynamic movements at frame  $f$ . The captured node  $\hat{\mathbf{x}}_i^f$  is an observable but the unposed node  $\tilde{\mathbf{x}}_i^f = \bar{\mathbf{x}}_i + \mathbf{b}_i(\boldsymbol{\beta}) + \boldsymbol{\xi}^f$  needs to be computed. After estimating  $\boldsymbol{\theta}^f$ , we compute  $\sum_{j=1}^K w_{ij} \mathbf{G}_j(\boldsymbol{\theta}^f) \bar{\mathbf{G}}_j^{-1}$  and we apply the inverse transform to the captured nodes  $\hat{\mathbf{x}}_i^f$ . Thereafter, we subtract the effect of pose blendshapes  $\mathbf{b}_i(\boldsymbol{\theta})$  to obtain the unposed node  $\tilde{\mathbf{x}}_i^f$ , which is the nodal position at rest pose after the dynamic effect

is applied. We call this operation unposing. In the following, the superscript  $f$  will be omitted unless it hurts clarity. To estimate the pose we minimize the squared distance between captured vertices and SMPL. To remove the influence of soft-tissue dynamics in the estimation of pose we use part weights  $\alpha_j$ , which are higher in rigid parts (like for instance the lower arms and lower legs). Let  $\mathcal{J}(j)$  denote the set of node indices corresponding to part  $j$ . With this definition we minimize

$$\sum_{j=1}^K \left( \alpha_j \sum_{k \in \mathcal{J}(j)} \|\hat{\mathbf{x}}_k - S_k(\boldsymbol{\theta})\|^2 \right) \quad (17)$$

where  $\hat{\mathbf{x}}_k$  is the captured position and  $S_k(\boldsymbol{\theta})$  is the position of the SMPL model given a pose  $\boldsymbol{\theta}$ . The shape parameter  $\boldsymbol{\beta}$  is omitted as it is constant for a subject. The parameter  $\alpha_j$  is set high for rigid body parts such as the limbs and low for non-rigid parts such as the trunk. While the assumption of existence of near-rigid nodes may not hold for overweight subjects, adjusting  $\alpha_j$  allowed us to obtain reasonable skeletal motions even in such cases. An alternative would be to estimate the pose based on inertial sensors attached at rigid parts as in [von Marcard et al. 2016, 2017]. The optimization is solved using the Levenberg-Marquardt nonlinear least squares algorithm [Lourakis 2004].

Given the estimated poses  $\boldsymbol{\theta}^f$ , we unpose each node sequence and obtain a sequence free from pose variations  $\tilde{X}_i = \{\tilde{\mathbf{x}}_i^f | f = 1, \dots, F\}$ . We then compute the mean amplitude of an unposed node sequence.

**5.1.2 Mean Amplitude.** Our simple implementation of mean amplitude is as follows. We find the local extrema  $\{\tilde{\mathbf{x}}_i^{f_1}, \dots, \tilde{\mathbf{x}}_i^{f_s}\}$  in  $\tilde{X}_i$  (i.e., positions where the dot product of velocity vectors in two consecutive frames is negative), and compute the mean amplitude

$$a(X_i) = \frac{\sum_{k=1}^{s-1} \|\tilde{\mathbf{x}}_i^{f_{k+1}} - \tilde{\mathbf{x}}_i^{f_k}\|}{s-1}. \quad (18)$$

Using the mean amplitude, we define the mean amplitude deviation for the error measure of  $X_i$  given  $\tilde{X}_i$  as

$$E(X_i(\mathbf{d}, \bar{\mathbf{x}}, \mathbf{p})) = a(\hat{X}_i) - a(X_i(\mathbf{d}, \bar{\mathbf{x}}, \mathbf{p})) \quad (19)$$

and subsequently the overall objective function can be minimized:

$$\mathbf{E}(X(\mathbf{d}, \bar{\mathbf{x}}, \mathbf{p})) = \sum_{i=1}^N |E(X_i(\mathbf{d}, \bar{\mathbf{x}}, \mathbf{p}))|. \quad (20)$$

Figure 5 shows the color maps for mean amplitudes  $a(\hat{X}_i)$  of the captured sequence for two different subjects. One can see that a highly deformable region such as the belly fat of subjects exhibits high mean amplitude.

The most basic error function for measuring the difference between two motion sequences would be to evaluate the Euclidean distance between the corresponding nodal positions at every frame. However, we found that this error function was difficult to minimize for highly dynamic cases such as ours, partly because even if the simulation has similar movement behavior to the captured one, the position error may accumulate to a large value if the vibration phase is shifted. Therefore, we adopted the mean amplitude deviation reflecting vibrational characteristics as an error function throughout some time interval.

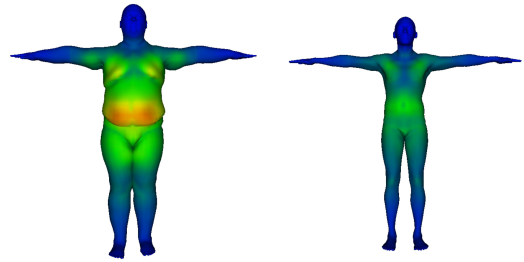


Fig. 5. Color maps for mean amplitudes.

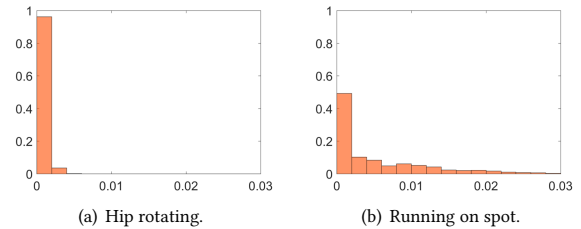


Fig. 6. Histogram of mean amplitudes of the left subject in Figure 5.

**Training sequence selection.** For efficiency we reduce the training data required for parameter optimization; we examine the mean amplitude histograms of the 14 Dyna sequences and select two with the highest mean amplitudes. In this way, we guarantee that vigorous dynamic deformations are present in the training set. The remaining 12 sequences are used as test data. Figure 6 shows the histogram of mean amplitudes for slow moving (a) and fast moving (b) sequences. We explain next how we fit the parameters to the training sequences.

## 5.2 Overall Optimization Scheme

The huge parameter space and lack of analytical gradients prevents us from optimizing all parameters simultaneously. Similar to [Wang et al. 2015], we use an alternating optimization scheme as below:

---

### Algorithm 1 Parameter Optimization Algorithm.

---

- 1:  $\mathbf{d} = \mathbf{d}_0, \bar{\mathbf{x}} = \bar{\mathbf{x}}_s, \mathbf{p} = \mathbf{p}_0$
  - 2: **repeat**
  - 3:   Update  $\mathbf{d}$  to decrease  $\mathbf{E}(X(\mathbf{d}, \bar{\mathbf{x}}, \mathbf{p}))$  with  $\bar{\mathbf{x}}$  and  $\mathbf{p}$  fixed (Section 5.4)
  - 4:   Update restshape  $\bar{\mathbf{x}}$  (Section 5.3)
  - 5:   Update  $\mathbf{p}$  to decrease  $\mathbf{E}(X(\mathbf{d}, \bar{\mathbf{x}}, \mathbf{p}))$  with  $\mathbf{d}$  and  $\bar{\mathbf{x}}$  fixed (Section 5.5)
  - 6:   Update restshape  $\bar{\mathbf{x}}$
  - 7: **until** minimization of  $\mathbf{E}(X(\mathbf{d}, \bar{\mathbf{x}}, \mathbf{p}))$  converges
- 

As for the initial value  $\mathbf{p}_0$ , we set  $E_0 = 2.0$  kPa,  $\nu_0 = 0.3$ , and  $c_0 = 0.99$  for every node after some test simulations. Note that  $E_0$  has a similar order of Young's modulus for the fatty tissue [Alkhouli et al. 2013].

Each optimization step is detailed below. Before starting the optimization, we estimate the skeletal poses  $\theta$  from the Dyna registrations, as explained next.

### 5.3 Rest Shape Estimation

The captured shape of a static pose is in equilibrium between the gravitational force and the internal force, and we need to estimate the true rest shape that is not affected by gravity in order to increase the accuracy in dynamics simulation. While the method of Chen et al. [2014] is applicable to our problem, we use a simple yet suitably working method. Under the assumption that the sagging effect of soft tissue is linear to the gravitational force, we apply an inverse gravity to  $\bar{x}_s$  to estimate  $\bar{x}$ . Specifically, we divide  $\bar{x}_s$  into two layers given  $\mathbf{d}$ , and then simulate the FEM layer with a given  $\mathbf{p}$  by applying gravitational force in the upward direction.

### 5.4 Layer Estimation

We estimate the RDF values  $\mathbf{d}|_S$  of the surface nodes so that the simulation of the layered model is close to the captured data. In order for the FEM node to reproduce the secondary dynamic effect such as jiggling, we made sure that the FEM nodes have a similar mean amplitude profile with that of the corresponding nodes. So, our goal is to find  $\mathbf{d}|_S$  that reduces Eq. (19) given  $\bar{x}, \mathbf{p}$  for all  $i$ . We will write  $E(X_i(\mathbf{d}))$  instead of  $E(X_i(\mathbf{d}, \bar{x}, \mathbf{p}))$  since  $\bar{x}$  and  $\mathbf{p}$  are fixed. Briefly, our strategy to optimize  $\mathbf{d}$  is to thicken the FEM layer where the secondary dynamic effects need to be emphasized. The optimization procedure starts with setting every node to VSMPL by assigning  $\mathbf{d} = \sigma$  to all surface nodes. Then, the simulation is done purely with VSMPL, which cannot create secondary dynamics. We collect the surface nodes that have errors larger than a threshold, and increase their  $\mathbf{d}$  by a constant number  $\eta$ . This leads to the creation of a FEM layer, and the simulation of the FEM nodes will create some secondary dynamics. We iteratively collect the surface nodes that still have large errors, and increase their  $\mathbf{d}$  until we have found the  $\mathbf{d}$  of all surface nodes.

---

#### Algorithm 2 RDF Optimization Algorithm.

---

```

1:  $\sigma = 1, \mathbf{d}|_S = 1, \text{step} = 0, e_{prev,i} = \text{large number}$ 
2:  $b_i = \text{true on } X^S|_E, \text{false on other surface nodes}$ 
3: while there exists a node with  $b_i = \text{false}$  do
4:   smooth  $\mathbf{d}|_S$  by  $\mathbf{d}|_S = (I + \delta L)\mathbf{d}|_S$ 
5:   compute  $\mathbf{d}$  by Eq. (14)
6:   simulate a test sequence and measure  $X_i(\mathbf{d})$  for  $b_i = \text{false}$ 
7:    $e_i = E(X_i(\mathbf{d}))$ 
8:   if  $e_i < \epsilon_1$  or  $|e_{prev,i} - e_i| < \epsilon_2$  then  $b_i = \text{true}$ 
9:   else  $d_i = d_i + \eta$ 
10:  end if
11:   $\text{step} = \text{step} + 1$ 
12:   $e_{prev,i} = e_i$ 
13: end while
14: return  $\mathbf{d}$ 

```

---

Algorithm 2 shows the pseudocode of the optimization algorithm. In line 2,  $b_i$  indicates whether the  $i$ -node has found the optimal value or not. Initially, it is set to false for every node, except the

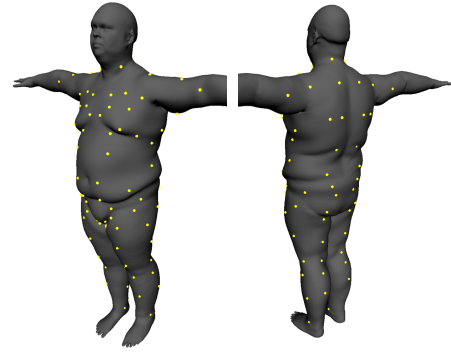


Fig. 7. 130 representative points used for FEM parameter optimization.

nodes  $X^S|_E$  in the end effector, as we will enforce the end effectors are simulated by VSMPL. Since the  $\mathbf{d}$  value of each node is updated individually in every step, we smooth them out over the surface using  $L$ , the Laplacian matrix over  $\bar{x}$  (line 4). To ensure the symmetry of RDF values across the sagittal plane, we can modify  $e_i$  in line 7 to  $e'_i = 0.5(e_i + e_{s(i)})$ , where  $s(i)$  is the symmetric node of  $i$ .

### 5.5 FEM Parameter Estimation

The actual physical parameters of the human body vary greatly per soft tissue and accurately estimating the person-specific parameters is a very complex task. Our goal for the parameter estimation is to obtain suitable values that make the animation look physically plausible, which allows for simplification to some degree.

Since optimizing every individual FEM element takes a prohibitive amount of time, we follow the method taken by Wang et al. [2015], in which Young's moduli of some representative points  $P \subset X^S$  are optimized and those of other points are determined from interpolation. Representative points  $P$  include 130 points selected symmetrically to a mid-sagittal plane (Figure 7). Without using particular knowledge about biological features, we manually selected 65 points on one half of the body, with the remaining points mirrored.

We set a uniform Young's modulus to the boundary nodes, i.e., those included in bones  $B$  and end-effectors  $V_E$ . Regarding the Poisson ratio  $\nu$  and damping  $c$ , we assume that uniform values over the body suffice. Therefore, the overall FEM parameters to be estimated are  $\mathbf{p} = (c, \mathbb{E} = \{E_1, \dots, E_{|P|+1}\}, \nu)$ . We take an alternating optimization scheme for  $\mathbf{p}$ , optimizing  $\mathbb{E}, \nu$ , and  $c$  in each step. Specifically we developed a two-step optimization method for estimating  $\mathbb{E}$  that has the advantage of having constant time complexity with respect to the number of representative points. The key insight is to approximate the cost function as a sum of  $P$  terms each depending only on the Young's modulus  $E_i$  of the representative point. This is analogous to factorizing a function or a distribution into a product of factors. To that end, we cluster skin nodes in the FEM layer into  $|P| + 1$  groups  $\{C_1, \dots, C_{|P|+1}\}$  according to the chosen  $P$  points. The error measure of  $C_i$  is then determined as the sum of errors of the cluster

$$E_{C_i}(\mathbb{E}) = \sum_{j \in C_i} E(X_j(\mathbb{E})). \quad (21)$$



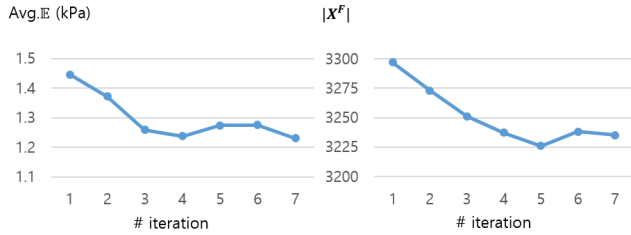


Fig. 8. Convergence of Algorithm 1 with respect to average  $\mathbb{E}$  and  $|\mathbf{X}^F|$ .

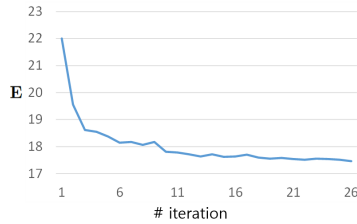


Fig. 9. The plot of  $\mathbb{E}$  during the optimization of  $\mathbb{E}$ .

*Initialization Step.* This step sets the initial Young’s modulus to the representative points. For this, we set a uniform Young’s modulus to all points and measure  $E_{C_i}$  for the simulated sequence. The tested Young’s modulus can be increased from 1.0kPa to 4.0kPa by 0.5, so a total of 7 simulations are performed. Then, each cluster  $C_i$  is given the Young’s modulus with the least  $E_{C_i}$ .

*Update Step.* Computing numerical gradients would require at least  $P$  cost function evaluations and would be prohibitively expensive. Approximating the cost function as a sum of independent terms allows us to approximate the gradient based solely on one evaluation of the cost function. We use the observed inverse-proportional relation between  $E$  and the mean amplitude deviation, i.e. a decreasing  $E$  will increase the mean amplitude. A positive  $E_{C_i}$  requires an increase in the amplitude, which is achieved by decreasing  $E_i$ , and vice versa. Therefore, for a given  $\mathbb{E}$ , we evaluate  $E_{C_i}$  and update  $E_i$  as  $E_i = E_i - \gamma E_{C_i}$ . The positive scalar  $\gamma$  is set to 30000 after some trials, which leads to a change in  $E_i$  less than 0.15kPa. This provides a means to approximate the gradient based only on the evaluation of the error. Note that  $E_i$  of a few clusters with a large portion of VSMPL nodes may not converge with this update rule because the deformation of VSMPL nodes is unaffected by  $E_i$ . To prevent this we bound  $E_i$  by a min 0.1 kPa and a max 7.0 kPa.

*Convergence of the Optimization.* We tested the convergence pattern of Algorithm 1 with respect to the average Young’s modulus and the number of FEM nodes of the outer layer. Figure 8 shows that the parameters converge stably within about 5 iterations, after which the variation of two values decreases to negligible amount, 0.1 kPa and 25, respectively. Figure 9 shows the decreasing pattern of  $\mathbb{E}$  during the optimization. The initial error is the one measured after the layer estimation. The error in the second step is measured after the initialization step, and the errors in subsequent

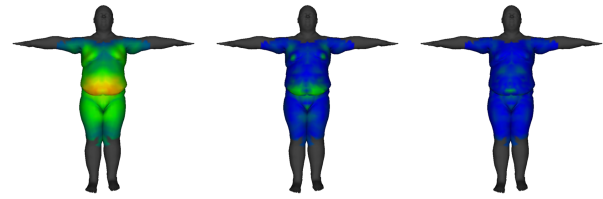


Fig. 10. Per-node mean amplitude deviation visualized in color. Grey area indicates VSMPL skin nodes. Left: Pure VSMPL simulation ( $E = 26.4214$ ). Middle: After RDF optimization ( $E = 12.6827$ ). Right: After Young’s modulus optimization ( $E = 10.9774$ ).

steps are from the iterative update steps. Figure 10 illustrates the error reduction after optimizing RDF and Young’s modulus.

The remaining parameters of  $\mathbf{p}$ ,  $\nu$  and  $c$ , are scalar values and thus allow for exhaustive search among the values sampled in equal intervals from a reasonably selected range of numbers. The parameters  $c$  and  $\nu$  are evaluated in intervals of  $[0.98, 0.998]$  with steps of 0.002 and  $[0.2, 0.4]$  with steps of 0.02, respectively. We found that Poisson’s ratio  $\nu$  plays a minor role compared to the Young’s modulus.

## 6 RESULTS

We have optimized the parameters of the layered models for every subject provided by Dyna (<http://dyna.is.tue.mpg.de/>) and evaluated the performance of the layered model and the optimization scheme discussed so far. The supplementary video shows the soft tissue deformation created by our method compared with the ground truth data.

Figure 11 shows the estimated RDF values on the character surfaces. One can see that the estimated FEM layers are thicker for the obese subjects who show large flesh jiggling during highly dynamic movements. In contrast, the estimated FEM layers are thin for slim subjects. This suggests that our optimization scheme estimates an optimal RDF reasonably well. Although by judging from the image, the degree of obesity seems similar for Subjects #1 and #3, one can see that Subject #3 is more muscular than Subject #1, and the optimized RDF values capture the difference between the two subjects correctly.

Table 1 reports the statistics of Young’s modulus for each subject. As expected, the recovered Young’s moduli are larger for slim or muscular subjects and low for overweight subjects. The physical parameters are estimated such that they produce optimal results with respect to the FEM layer created by RDF values, and may converge to different optimal parameters if we change the RDF values. Thus, these parameters are not to be considered as the exact

Subject	1	2	3	4	5	6	7	8	9	10
Avg.	1.24	2.13	1.5	2.24	2.16	2.11	1.92	2.32	1.72	2.19
Min.	0.1	0.1	0.1	0.29	0.1	0.87	0.1	0.1	0.17	0.1
Max.	4.0	4.25	4.0	4.0	7.0	4.0	4.58	7.0	4.56	4.15
SD	0.71	1.2	1.07	0.95	1.34	1.04	1.09	1.74	1.04	0.98

Table 1. Statistics of each subject’s Young’s modulus (kPa) on all test points.



Fig. 11. Colored visualization of the estimated RDF values of each character.

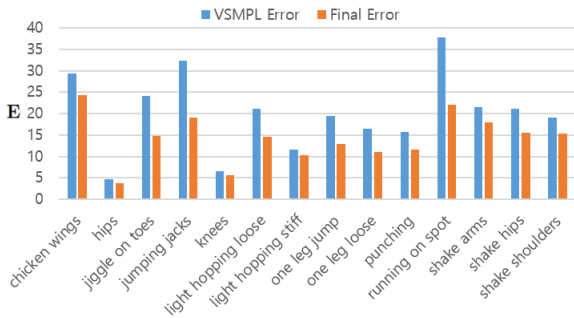


Fig. 12. Average errors  $E$  for all subjects of pure VSMPL simulation (blue) and after optimization (orange). “Jumping jacks” and “running on spot” sequences are used for training.

physical properties of a human subject. Rather, they are optimized for a particular layer model.

As explained in Section 5, we select two dynamic sequences for the training data and use the remaining sequences for the test data, which can be viewed in the supplementary video. Figure 12 shows the average  $E$  (Eq. (20)) for all the subjects. The blue and orange lines indicate the errors at the initial stage and after optimization, respectively. The results show that the optimized parameters can be suitably generalized to other sequences.

**Robustness of Representative Points.** More accurate parameter optimization will be possible by optimizing over the whole surface nodes rather than over a subset of points, but our strategy is a good trade-off between performance and computational cost during training. In order to test the robustness of our method with respect to the placement of representative points, we ran 10 different random placements of the points on Subject#1 and measured differences in

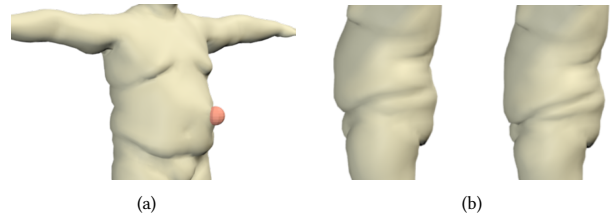


Fig. 13. (a) Creating soft tissue deformation with user interaction. (b) Rest shapes under regular gravity (left) and 7 times heavier gravity (right).

the error  $E$  and average Young’s modulus between manual and random selections. The normalized error difference (difference divided by original error) was 8.5% on average with a variance of 0.8%, and the normalized difference of average Young’s modulus was 4.4% on average with a variance of 0.03%. This indicates that our method is not sensitive to the placement of the points.

**Further Applications.** Simulating the FEM layer not only allows the creation of a realistic secondary dynamic effect, but also the generation of soft tissue deformation due to contact interactions. Figure 13 (a) shows a snapshot in which the belly region is naturally deformed by a user’s pulling action. Figure 13 (b) shows an experiment to create an increased sagging effect of soft tissue due to an increased gravity.

One of the strong advantages of our layered model is that all subject shapes share the same topology. Hence, we can directly apply the learned parameters from one subject to another producing interesting effects like making a heavy subject rigid and vice-versa. Figure 14 shows a case where Subject #1’s parameters have been applied to a new character. Furthermore, since the segmentation into layers is controlled by the RDF, artists can easily modify the

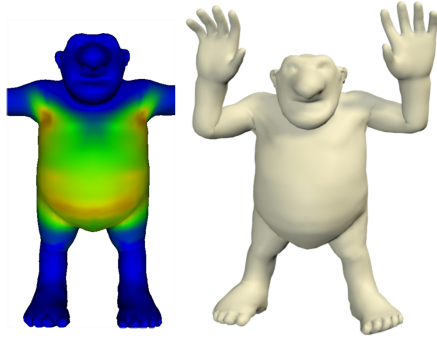


Fig. 14. A new character with Subject #1's parameters applied. Left : Transferred RDF from Subject #1. Right : Resulting animation.

estimated values using a paintbrush-like UI, which allows them to produce special effects as shown in the supplementary video.

**Computation Time.** The number of nodes and elements of the original volumetric mesh are 9757 and 37717, respectively. After RDF optimization, they are reduced to 25-40 % of the original, which results in an increase in computational speed by 3-5.5 times. Our method runs in close to real time rates. The simulation time step can be set to 2 ms for “running on spot” and to 4 ms for all other sequences with a computation time of around 3.7 ms per simulation step with 8 core Intel Xeon CPU at 2.4GHz. Computing the boundary node positions of VSMPL layer is performed at each display time step, which costs an additional 2 ms. The offline parameter optimization took around two hours in each iteration (line 3 to 6) of Algorithm 1, with convergence reached after 3-4 iterations. In all, it takes 6-8 hours to obtain the optimal parameters for one subject.

## 7 DISCUSSION AND FUTURE WORK

In this paper we proposed a layered simulation model that combines VSMPL and FEM simulation. The model has the major advantage that it allows an efficient parameter estimation and increases the computational efficiency while retaining the capabilities of physics simulation. We developed a RDF-based layer modeling approach, which provides intuitive control of the layer composition. Given the captured soft tissue deformation, we optimized the layer parameters as well as the physical parameters that simulate a character soft tissue deformation which is close to the captured data.

The method presented in this paper has some limitations, which give interesting future research directions. First, our method for optimizing parameters could be improved. Non-uniform damping coefficients will make the simulation better follow the captured sequence. Second, we used line segments as the skeleton model because they are easy to create and it is a standard format used in industry. However, we expect that RDF can be better constructed if the actual 3D geometry of the human skeleton is used, especially the bones such as the rib cage and pelvis that cannot be well modeled by line segments. In our work we excluded the hands, feet, and head from the FEM simulation. Since these parts have a very thin skin layer, we need to construct a high resolution volumetric mesh that reflects the anatomical structure of human soft tissue in order to

extend the layered model to the entire body. In addition, the effect of stretching and sliding of thin skin would require special treatments [Li et al. 2013]. Moreover, in this work we ignored the compression effects of clothing in the body. Future work will leverage recent methods to capture human body and clothing [Pons-Moll et al. 2017; Zhang et al. 2017] to study such interactions.

Our method provides the means to create a person-specific simulation model. By constructing a large database of such models, we will be able to build a statistical method that can predict a suitable layered model for new, unseen characters. This will greatly simplify the laborious task of character modeling.

## REFERENCES

- Dicko AliHamadi, Tiantian Liu, Benjamin Gilles, Ladislav Kavan, François Faure, Olivier Palombi, and MariePaule Cani. 2013. Anatomy Transfer. *ACM Trans. Graph.* 32, 6, Article 188 (Nov. 2013), 8 pages. DOI : <https://doi.org/10.1145/2508363.2508415>
- Nadia Alkhoul, Jessica Mansfield, Ellen Green, James Bell, Beatrice Knight, Neil Liveredge, Ji Chung Tham, Richard Welbourn, Angela C Shore, Katarina Kos, and others. 2013. The mechanical properties of human adipose tissues and their relationships to the structure and composition of the extracellular matrix. *American Journal of Physiology-Endocrinology and Metabolism* 305, 12 (2013), E1427–E1435.
- Brett Allen, Brian Curless, Zoran Popović, and Aaron Hertzmann. 2006. Learning a correlated model of identity and pose-dependent body shape variation for real-time synthesis. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 147–156.
- Steven S An, Theodore Kim, and Doug L James. 2008. Optimizing cubature for efficient integration of subspace deformations. In *ACM Transactions on Graphics (TOG)*, Vol. 27. ACM, 165.
- Alexis Angelidis and Karan Singh. 2007. Kinodynamic Skinning Using Volume-preserving Deformations. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '07)*. 129–140. <http://dl.acm.org/citation.cfm?id=1272690.1272709>
- Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. 2005. SCAPE: shape completion and animation of people. In *ACM Transactions on Graphics (TOG)*, Vol. 24. ACM, 408–416.
- Oscar Kin-Chung Au, Chiew-Lan Tai, Hung-Kuo Chu, Daniel Cohen-Or, and Tong-Yee Lee. 2008. Skeleton extraction by mesh contraction. In *ACM Transactions on Graphics (TOG)*, Vol. 27. ACM, 44.
- Ilya Baran and Jovan Popović. 2007. Automatic rigging and animation of 3d characters. In *ACM Transactions on Graphics (TOG)*, Vol. 26. ACM, 72.
- Jernej Barbic and Doug L James. 2005. Real-time subspace integration for St. Venant-Kirchhoff deformable models. In *ACM transactions on graphics (TOG)*, Vol. 24. ACM, 982–990.
- Xiang Chen, Changxi Zheng, Weiwei Xu, and Kun Zhou. 2014. An Asymptotic Numerical Method for Inverse Elastic Shape Design. *ACM Trans. Graph.* 33, 4, Article 95 (July 2014), 11 pages. DOI : <https://doi.org/10.1145/2601097.2601189>
- Crispin Deul and Jan Bender. 2013. Physically-Based Character Skinning. *VRIPHYS 13* (2013), 25–34.
- Olivier Dienne and Martin de Lasa. 2013. Geodesic voxel binding for production character meshes. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 173–180.
- Ming Gao, Nathan Mitchell, and Eftychios Sifakis. 2014. Steklov-Poincaré Skinning. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '14)*. 139–148. <http://dl.acm.org/citation.cfm?id=2849517.2849541>
- Fabian Hahn, Sebastian Martin, Bernhard Thomaszewski, Robert Sumner, Stelian Coros, and Markus Gross. 2012. Rigspace Physics. *ACM Trans. Graph.* 31, 4, Article 72 (July 2012), 8 pages. DOI : <https://doi.org/10.1145/2185520.2185568>
- Naoya Iwamoto, Hubert PH Shum, Longzhi Yang, and Shigeo Morishima. 2015. Multi-layer Lattice Model for Real-Time Dynamic Character Deformation. In *Computer Graphics Forum*, Vol. 34. Wiley Online Library, 99–109.
- Alec Jacobson, Ilya Baran, Ladislav Kavan, Jovan Popović, and Olga Sorkine. 2012. Fast Automatic Skinning Transformations. *ACM Trans. Graph.* 31, 4, Article 77 (July 2012), 10 pages. DOI : <https://doi.org/10.1145/2185520.2185573>
- Alec Jacobson, Ilya Baran, Jovan Popovic, and Olga Sorkine. 2011. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.* 30, 4 (2011), 78.
- Doug L. James and Dinesh K. Pai. 2002. DyRT: Dynamic Response Textures for Real Time Deformation Simulation with Graphics Hardware. *ACM Trans. Graph.* 21, 3 (July 2002), 582–585. DOI : <https://doi.org/10.1145/566654.566621>
- Doug L. James and Christopher D. Twigg. 2005. Skinning Mesh Animations. In *ACM SIGGRAPH 2005 Papers (SIGGRAPH '05)*. ACM, New York, NY, USA, 399–407. DOI : <https://doi.org/10.1145/1186822.1073206>

- Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. 2007. Harmonic Coordinates for Character Articulation. *ACM Trans. Graph.* 26, 3, Article 71 (July 2007). DOI: <https://doi.org/10.1145/1276377.1276466>
- Petr Kadleček, Alexandru-Eugen Ichim, Tiantian Liu, Jaroslav Krivánek, and Ladislav Kavan. 2016. Reconstructing Personalized Anatomical Models for Physics-based Body Animation. *ACM Trans. Graph.* 35, 6, Article 213 (Nov. 2016), 13 pages. DOI: <https://doi.org/10.1145/2980179.2982438>
- Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O'Sullivan. 2008. Geometric skinning with approximate dual quaternion blending. *ACM Transactions on Graphics (TOG)* 27, 4 (2008), 105.
- Ladislav Kavan and Olga Sorkine. 2012. Elasticity-inspired Deformers for Character Articulation. *ACM Trans. Graph.* 31, 6, Article 196 (Nov. 2012), 8 pages. DOI: <https://doi.org/10.1145/2366145.2366215>
- Junggon Kim and Nancy S. Pollard. 2011. Fast Simulation of Skeleton-driven Deformable Body Characters. *ACM Trans. Graph.* 30, 5, Article 121 (Oct. 2011), 19 pages. DOI: <https://doi.org/10.1145/2019627.2019640>
- Theodore Kim and Doug L. James. 2011. Physics-based Character Skinning Using Multi-domain Subspace Deformations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '11)*. 63–72. DOI: <https://doi.org/10.1145/2019406.2019415>
- Yeara Kozlov, Derek Bradley, Moritz Bäcker, Bernhard Thomaszewski, Thabo Beeler, and Markus Gross. 2017. Enriching Facial Blendshape Rigs with Physical Simulation. *Computer Graphics Forum* 36, 2 (2017). DOI: <https://doi.org/10.1111/cgf.13108>
- Paul G. Kry, Doug L. James, and Dinesh K. Pai. 2002. EigenSkin: Real Time Large Deformation Character Skinning in Hardware. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '02)*. 153–159. DOI: <https://doi.org/10.1145/545261.545286>
- Caroline Larboulette, Marie-Paule Cani, and Bruno Araldi. 2005. Dynamic skinning: adding real-time dynamic effects to an existing character animation. In *Proceedings of the 21st spring conference on Computer graphics*. ACM, 87–93.
- Binh Huy Le and Jessica K Hodgins. 2016. Real-time skeletal skinning with optimized centers of rotation. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 37.
- Sung-Hee Lee, Eftychios Sifakis, and Demetri Terzopoulos. 2009. Comprehensive Biomechanical Modeling and Simulation of the Upper Body. *ACM Trans. Graph.* 28, 4, Article 99 (Sept. 2009), 17 pages. DOI: <https://doi.org/10.1145/1559755.1559756>
- Yuencheng Lee, Demetri Terzopoulos, and Keith Waters. 1995. Realistic Modeling for Facial Animation. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. ACM, New York, NY, USA, 55–62. DOI: <https://doi.org/10.1145/218380.218407>
- J. P. Lewis, Matt Corder, and Nickson Fong. 2000. Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-driven Deformation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. 165–172. DOI: <https://doi.org/10.1145/344779.344862>
- Duo Li, Shinjiro Sueda, Debanga R. Neog, and Dinesh K. Pai. 2013. Thin Skin Elastodynamics. *ACM Trans. Graph.* 32, 4, Article 49 (July 2013), 10 pages. DOI: <https://doi.org/10.1145/2461912.2462008>
- Siwang Li, Jin Huang, Fernando de Goes, Xiaogang Jin, Hujun Bao, and Mathieu Desbrun. 2014. Space-time editing of elastic motion through material optimization and reduction. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 108.
- Yaron Lipman, David Levin, and Daniel Cohen-Or. 2008. Green Coordinates. *ACM Trans. Graph.* 27, 3, Article 78 (Aug. 2008), 10 pages. DOI: <https://doi.org/10.1145/1360612.1360677>
- Libin Liu, KangKang Yin, Bin Wang, and Baining Guo. 2013. Simulation and Control of Skeleton-driven Soft Body Characters. *ACM Trans. Graph.* 32, 6, Article 215 (Nov. 2013), 8 pages. DOI: <https://doi.org/10.1145/2508363.2508427>
- Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2015. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 248.
- M.I.A. Lourakis. Jul. 2004. levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++. <http://www.ics.forth.gr/~lourakis/levmar/>. (Jul. 2004).
- Nadia Magnenat-Thalmann, Richard Laperrière, and Daniel Thalmann. 1988. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings on Graphics interface '88*. Citeseer.
- Richard Malgat, Benjamin Gilles, David IW Levin, Matthieu Nesme, and François Faure. 2015. Multifarious hierarchies of mechanical models for artist assigned levels-of-detail. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 27–36.
- Aleka McAdams, Yongning Zhu, Andrew Selle, Mark Empey, Rasmus Tamstorf, Joseph Teran, and Eftychios Sifakis. 2011. Efficient Elasticity for Character Skinning with Contact and Collisions. *ACM Trans. Graph.* 30, 4, Article 37 (July 2011), 12 pages. DOI: <https://doi.org/10.1145/2010324.1964932>
- Eder Miguel, Derek Bradley, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Miguel A Otaduy, and Steve Marschner. 2012. Data-Driven Estimation of Cloth Simulation Models. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 519–528.
- Alex Mohr and Michael Gleicher. 2003. Building Efficient, Accurate Character Skins from Examples. In *ACM SIGGRAPH 2003 Papers (SIGGRAPH '03)*. ACM, New York, NY, USA, 562–568. DOI: <https://doi.org/10.1145/1201775.882308>
- Tomohiko Mukai and Shigeru Kuriyama. 2016. Efficient Dynamic Skinning with Low-rank Helper Bone Controllers. *ACM Trans. Graph.* 35, 4, Article 36 (July 2016), 11 pages. DOI: <https://doi.org/10.1145/2897824.2925905>
- Matthias Müller and Markus Gross. 2004. Interactive virtual materials. In *Proceedings of Graphics Interface 2004*. 239–246.
- Akihiko Murai, Q Youn Hong, Katsu Yamane, and Jessica K Hodgins. 2016. Dynamic skin deformation simulation using musculoskeletal model and soft tissue dynamics. *Computational Visual Media* (2016), 1–12.
- Sang Il Park and Jessica K Hodgins. 2006. Capturing and animating skin deformation in human motion. In *ACM Transactions on Graphics (TOG)*, Vol. 25. ACM, 881–889.
- Sang Il Park and Jessica K Hodgins. 2008. Data-driven modeling of skin and muscle deformation. In *ACM Transactions on Graphics (TOG)*, Vol. 27. ACM, 96.
- A. Pentland and J. Williams. 1989. *Good Vibrations: Modal Dynamics for Graphics and Animation*. Vol. 23. 207–214 pages.
- Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael Black. 2017. ClothCap: Seamless 4D Clothing Capture and Retargeting. *ACM Transactions on Graphics, (Proc. SIGGRAPH)* 36, 4 (2017).
- Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J Black. 2015. Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 120.
- Gerard Pons-Moll and Bodo Rosenhahn. 2011. *Model-Based Pose Estimation*. Springer, Chapter 9, 139–170.
- Hyewon Seo, Frederic Cordier, and Nadia Magnenat-Thalmann. 2003. Synthesizing animatable body models with parameterized shape modifications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 120–125.
- Hang Si. 2015. TetGen, a Quality Tetrahedral Mesh Generator. *AMC Trans. Math. Softw.* 41, 2 (2015), 11. <http://wias-berlin.de/software/tetgen/>
- Weiguang Si, Sung-Hee Lee, Eftychios Sifakis, and Demetri Terzopoulos. 2014. Realistic Biomechanical Simulation and Control of Human Swimming. *ACM Trans. Graph.* 34, 1, Article 10 (Dec. 2014), 15 pages. DOI: <https://doi.org/10.1145/2626346>
- PeterPike J Sloan, Charles F Rose III, and Michael F Cohen. 2001. Shape by example. In *Proceedings of the 2001 symposium on Interactive 3D graphics*. ACM, 135–143.
- Timo von Marcard, Gerard Pons-Moll, and Bodo Rosenhahn. 2016. Human Pose Estimation from Video and IMUs. *Transactions on Pattern Analysis and Machine Intelligence PAMI* (Jan. 2016).
- Timo von Marcard, Bodo Rosenhahn, Michael Black, and Gerard Pons-Moll. 2017. Sparse Inertial Poser: Automatic 3D Human Pose Estimation from Sparse IMUs. *Computer Graphics Forum 36(2), Proceedings of the 38th Annual Conference of the European Association for Computer Graphics (Eurographics)* (2017).
- Bin Wang, Longhua Wu, KangKang Yin, Uri Ascher, Libin Liu, and Hui Huang. 2015. Deformation capture and modeling of soft objects. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 94.
- Xiaohuan Corina Wang and Cary Phillips. 2002. Multi-weight enveloping: least-squares approximation techniques for skin animation. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 129–138.
- Xiaofeng Wu, Rajaditya Mukherjee, and Huamin Wang. 2015. A Unified Approach for Subspace Simulation of Deformable Bodies in Multiple Domains. *ACM Trans. Graph.* 34, 6, Article 241 (Oct. 2015), 9 pages. DOI: <https://doi.org/10.1145/2816795.2818065>
- Hongyi Xu and Jernej Barbič. 2016. Pose-space Subspace Dynamics. *ACM Trans. Graph.* 35, 4, Article 35 (July 2016), 14 pages. DOI: <https://doi.org/10.1145/2897824.2925916>
- Hongyi Xu, Yijing Li, Yong Chen, and Jernej Barbič. 2015. Interactive material design using model reduction. *ACM Transactions on Graphics (TOG)* 34, 2 (2015), 18.
- Yin Yang, Weiwei Xu, Xiaohu Guo, Kun Zhou, and Baining Guo. 2013. Boundary-Aware Multidomain Subspace Deformation. *IEEE Transactions on Visualization and Computer Graphics* 19, 10 (2013), 1633–1645. DOI: <https://doi.org/doi.ieeecomputersociety.org/10.1109/TVCG.2013.12>
- Chao Zhang, Sergi Pujades, Michael Black, and Gerard Pons-Moll. 2017. Detailed, accurate, human shape estimation from clothed 3D scan sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Lifeng Zhu, Xiaoyan Hu, and Ladislav Kavan. 2015. Adaptable Anatomical Models for Realistic Bone Motion Reconstruction. *Comput. Graph. Forum* 34, 2 (May 2015), 459–471. DOI: <https://doi.org/10.1111/cgf.12575>