Dyna: A Model of Dynamic Human Shape in Motion

Gerard Pons-Moll* Javier Romero* Naureen Mahmood* Michael J. Black*
Max Planck Institute for Intelligent Systems, Tübingen, Germany

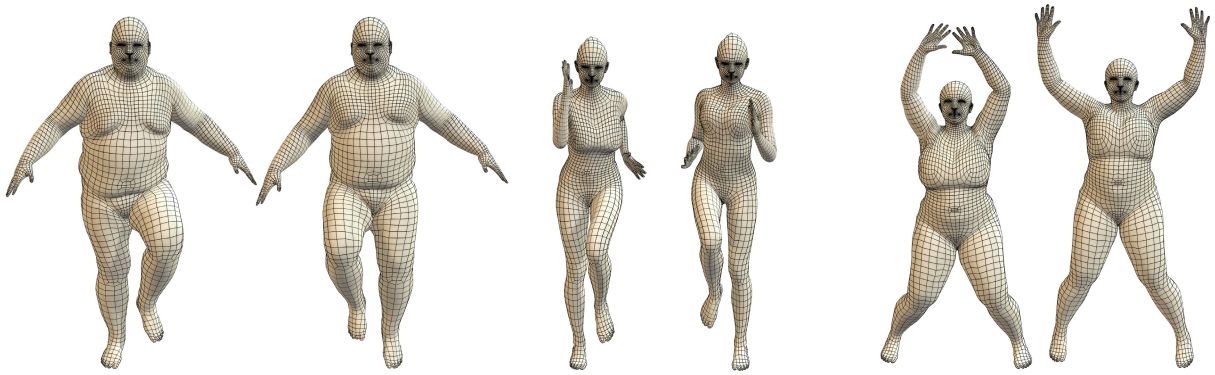

Figure 1: Dyna: *Three different animated body shapes performing different motions with soft-tissue deformations predicted by Dyna. These deformations are predicted by a learned function that depends on body shape, the angular velocity and acceleration of body parts, the velocity and acceleration of the body, and the history of previous non-rigid deformations. Dyna generalizes to new body shapes of varying body mass index and can be applied to stylized characters to automatically add realistic soft-tissue motions to animations.*

Abstract

To look human, digital full-body avatars need to have soft-tissue deformations like those of real people. We learn a model of soft-tissue deformations from examples using a high-resolution 4D capture system and a method that accurately registers a template mesh to sequences of 3D scans. Using over 40,000 scans of ten subjects, we learn how soft- tissue motion causes mesh triangles to deform relative to a base 3D body model. Our *Dyna* model uses a low-dimensional linear subspace to approximate soft-tissue deformation and relates the subspace coefficients to the changing pose of the body. Dyna uses a second-order auto-regressive model that predicts soft-tissue deformations based on previous deformations, the velocity and acceleration of the body, and the angular velocities and accelerations of the limbs. Dyna also models how deformations vary with a person's body mass index (BMI), producing different deformations for people with different shapes. Dyna realistically represents the dynamics of soft tissue for previously unseen subjects and motions. We provide tools for animators to modify the deformations and apply them to new stylized characters.

CR Categories: I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: Human animation, motion capture, human shape, soft-tissue motion

*e-mail:{gerard.pons.moll,jromero,nmahmood,black}@tue.mpg.de

1 Introduction

Interest in creating and animating realistic virtual humans has led to a wide variety of models. The most realistic methods model body shape using 3D scans of many bodies and many poses [Allen et al. 2002; Anguelov et al. 2005; Hasler et al. 2009; Hirshberg et al. 2012; Chen et al. 2013]. These models can capture pose-dependent *static* shape deformations but cannot realistically model the effects of soft-tissue dynamics on bodies in motion. While physics-based models and finite-element modeling provide a possible solution, the complexity of such systems makes them difficult to produce and control. Instead we take a learning-based approach to modeling the deformation of visible surface geometry caused by soft-tissue dynamics. Previous learning-based methods rely on the motion capture of many markers attached to the body [Park and Hodgins 2006; Park and Hodgins 2008]. Markers provide limited spatial resolution, are time consuming to apply, may change the motion we want to observe, and large marker sets pose technical difficulties for marker identification and tracking. Consequently these approaches do not easily scale to capture detailed soft-tissue deformations on a wide variety of body shapes. What is needed is a method to capture surface deformations of the body at high spatial and temporal resolutions and a mathematical model relating these deformations to the motion and body shapes of novel characters.

To that end, we propose a new model called *Dyna* that is learned from examples and is able to produce realistic soft-tissue motions for a wide range of body shapes and motions as illustrated in Fig. 1. Dyna is an extension of the SCAPE model [Anguelov et al. 2005] to include full body deformations that are driven by the motion of the body. While SCAPE approximates static surface deformations of soft tissue, Dyna approximates dynamic surface deformations related to the motion of soft tissue. At the core, Dyna is a mathematical model that relates deformations of the body surface to changing poses of the body in time. To make this feasible, we learn a low-dimensional model of these deformations from 3D scans containing soft tissue in motion. The scans are recorded by a 4D full-body capture system that outputs high-resolution 3D meshes of the body at 60 fps, revealing detailed soft- tissue motions.

Given such data, we develop an algorithm that accurately aligns a template mesh to sequences of 3D scans. The approach uses an existing SCAPE model to make this reliable and is fully automatic. Using this method we have aligned over 40,000 body scans of 10 training subjects and an additional 30,000 scans of 6 test subjects. The training meshes are available for research purposes. [1]

Deviations of these aligned meshes from a SCAPE model, which captures only static deformations, are primarily due to soft-tissue motion. As with SCAPE, these deviations are represented as triangle deformations. Given deformations for 10,000 of the training meshes, per gender, we perform principal component analysis (PCA) to obtain a reduced linear subspace that can approximate the soft-tissue surface deformations in the training set.

Dyna is a second-order autoregressive model that predicts the low-dimensional linear coefficients of soft-tissue motion from the velocity and acceleration of the body, the angular velocities and accelerations of the body parts, and the history of previous soft-tissue shape coefficients. These soft-tissue deformations can be thought of as another layer on top of SCAPE. However, unlike the standard pose-dependent deformations of SCAPE, the soft-tissue ones are a function of full body motion and are constructed from the dynamic PCA space.

Soft-tissue motion varies, not only with the body motion, but with the composition of the body. While it is an imperfect measure, we use body mass index (BMI) as a variable related to the amount of soft-tissue dynamics. For training, we captured subjects (5 men and 5 women) spanning a range of body shapes and BMIs performing many actions. Dyna models the soft-tissue deformations as resulting from a mixture of causes. We train an autoregressive model for each subject and then represent a new subject with a new BMI as a weighted linear combination of these models. This allows us to predict the dynamics for a new person with a different BMI from the training data or to vary the dynamics of a given person by artificially varying their BMI.

While we refer to Dyna as a dynamical model of soft-tissue deformation, we do not attempt to model the physics of soft-tissue dynamics. It is important to draw the distinction between the static and dynamic deformations of real tissue, which are governed by physical forces, and our *mathematical model*, which is an approximation of their observable effect on the surface of the body. When we talk about soft-tissue dynamics in the world, we refer to the causal physical process (which we do not model), whereas, when we talk about soft-tissue deformation in our model, we mean the visible deformation of the surface. Dyna models only this deformation of the 3D surface shape analogous to how SCAPE is a model of static surface deformations and not a model of statics (forces).

We demonstrate Dyna by animating test bodies in motion and show how it generalizes to new body shapes and motions not seen in training. The resulting animations contain visually realistic and compelling soft-tissue deformations. While realism is important, animators often want to edit or manipulate soft-tissue deformations for dramatic effect. To that end, we provide several controls. First, we allow overall attenuation or amplification of the soft-tissue deformations. Second, we provide a way to focus this control to particular body parts, allowing animators to, for example, make the stomach or chest more jiggly. Third, we show how Dyna can be applied to animate stylized characters. For this we need the character mesh to be in alignment with our template mesh and then Dyna generates the soft-tissue dynamics on the new character.

[1] http://dyna.is.tuebingen.mpg.de/

2 Previous Work

The animation of physical dynamics of varying materials is widely studied in computer graphics. Here we focus on methods that reproduce the dynamics of the full human body in motion. There are two main classes of approaches: physically-based and data-driven.

Physically-based models. The classic approach to "layered character construction" [Chadwick et al. 1989] uses a skeleton that drives soft-tissue motions including kinematic deformations (e.g. muscle bulging) and dynamics. The fat/tissue layer is typically represented by a simplified, low-resolution, mass-spring model. Many methods exist for controlling dynamic simulation of general rigged models [Capell et al. 2002] using finite element methods (FEM); e.g. dynamic elements can be layered on top of a standard skinned model [Larboulette et al. 2005].

Numerous authors have studied the material properties of human soft tissue with the goal of realistic animation [Maurel et al. 1998] and numerous methods for modeling and animating muscles have been proposed [Assassi et al. 2012; Aubel and Thalmann 2001; Fan et al. 2014; Lee et al. 2009; Scheepers et al. 1997; Sifakis et al. 2005; Teran et al. 2005; Terzopoulos and Waters 1990; Wilhelms and Van Gelder 1997]. Creating realistic musculature and fat by hand is difficult and setting the physical parameters is even harder. Consequently, one approach attempts to infer this from meshes (either artist defined or scans); e.g. [Pratscher et al. 2005].

Bickel et al. [2009] design a trinocular stereo system combined with a force probe to deform simple objects and capture their physical properties. They fit an FEM model to the data and then simulate the materials in motion. Their capture approach is not practical for full bodies and does not address object self motion or articulation.

Until recently, detailed physical models have not produced results that look as realistic as methods that directly learn surface deformations from examples. Recently, Weta Digital has employed a physical approach in many feature films with beautiful results. Such methods, however, are labor intensive and require expert animators to create and animate the models. A model built for one character cannot be quickly repurposed to others. Weta and others often focus on stylized characters whereas here we focus on generating realistic humans where viewers are particularly sensitive to errors.

Approximating physical models from simulations. Since controlling and efficiently animating finite-element models is difficult, there is a long literature on approximating dynamics based on simulations [Hahn et al. 2012; James and Pai 2002; Kim and James 2009; Shi et al. 2008]. Using clothing simulations, there are methods that learn low-dimensional (low-D) PCA models of cloth and capture the dynamics of cloth as autoregressive models of cloth parameters and kinematics [de Aguiar et al. 2010; Guan et al. 2012]. These are similar in spirit to our approach but are not learned from real data. No physics-based model exists for generating training body meshes with the realism of 4D scanning.

Learning models of deformation. Capell et al. [2007] make forces part of a surface deformation rig and optimize the forces to model human body deformations. The forces, however, are learned from *static* body scans to capture kinematic deformations. EigenSkin [Kry et al. 2002] captures fine deformations as deviations mapped to a rest pose using a low-D subspace. This is similar in spirit to how we learn soft-tissue deformations except that we model temporal dynamics as well.

Kim and James [2011] note that learning a global linear subspace to represent soft-tissue dynamics is quite inefficient and, instead, break the body into regions and learn separate modal models for

each region. They then have the problem of stitching these back together in a seamless way. They learn their method from body scans but the animations lack realism. In contrast, we take a global approach here because we find it better captures correlated effects across the body and with the overall motion of the skeleton. When a person wiggles their arm, soft tissue in the torso and legs is affected – human body deformations are not fully local.

Related to the idea of learning low-D models is work on generic mesh sequence compression. Alexa and Müller [2000] introduce the idea of representing sequences of 3D meshes using PCA. Karni and Gotsman [2004] take this further and combine PCA with linear predictive coding. The prediction does not code differences from a model and does not relate temporal dynamics to accelerations of an underlying model. Kim and James [2009] take a different approach to on-line model reduction. They focus on a dynamics simulation approach and incrementally approximate the mass and damping matrices using a low-D linear model.

Data-driven models. Most previous methods for capturing natural body shape have used laser scanners, which are slow and require holding a static pose [Allen et al. 2002; Anguelov et al. 2005]. Such scans can be used to build a good model of pose-dependent deformation but they do not capture the dynamics of soft tissue. Work on modeling deforming bodies typically relies on sparse surface data and has a long history [Metaxas and Terzopoulos 1993]. Here we focus on methods that acquire surface data of moving bodies, fit meshes to them, and attempt to model soft-tissue dynamics.

Sparse 3D data. The work most similar to ours is the seminal work of Park and Hodgins [2006; 2008]. They use a marker-based capture system with hundreds of markers placed on the body. In [Park and Hodgins 2006] they capture 350 markers on an actor in motion, fit a 3D model of the actor to the data, and then realistically replay the action on the character with soft-tissue motions. They segment the body into rigidly moving parts and then compute the residual deformation due to soft tissue. They model this with simple modes of deformation plus a further residual deformation using a scattered data interpolation method. The method captures soft-tissue motions but does not model their temporal dynamics.

In [Park and Hodgins 2008], they go further to drive soft-tissue motion from skeletal mocap. Like us, deformations are factored into static and dynamic deformations. They perform PCA on the dynamic marker motions and model the motion in a low-D space using a second order dynamics equation. They do this analysis per body part, not across the whole body. They demonstrate the method on one actor, who was captured and then simulated in a limited set of motions.

Neumann et al. [2013a] use approximately 1000 dots painted on the body and a multi-camera setup to track them. They do this for one shoulder and arm on several subjects. Manual input is used to align one template mesh per subject and then track the mesh across poses. Then muscle deformations are learned as a non-linear function of body type, pose and external forces. They show generalization to new motions and forces but, while they capture sequences, they do not model temporal dynamics. We capture the full body at much higher resolution, less intrusively, and go further to define a dynamical model of soft-tissue deformation.

At the other extreme, Loper et al. [2014] extract full body soft-tissue motions from standard sparse mocap marker sets. They approximate these soft-tissue motions as shape variations within the space of human body shapes. They do not *model* dynamics, rather they fit body shape to mocap marker data that contains dynamics.

Dense 3D data. For the above methods, the set of markers, while dense compared with standard mocap systems, only provides a sparse sampling of the body surface. In constrast, Neumann et al. [2013b], like us, work with 3D mesh sequences. From these they learn a sparse model of deformations. They model meshes in terms of these sparse components but do not model or analyze temporal dynamics.

Figure 2: 4D scanner. *Our scanning system captures full-body 3D mesh sequences at 60 fps using active stereo. The color image from each of 22 scanning units is shown (see text).*

Tsoli et al. [2014] capture 3D meshes of people breathing, represent the deformations in a low-dimensional space, and develop a temporal model of the coefficients in this space. Unlike our work, their scans are of static shapes and their temporal model relates to lung volume, not body motion.

de Aguiar and Ukita [2012] capture relatively coarse meshes of the full body using a multi-camera silhouette method. They model the motion of the mesh using a standard blend-skinned model and then model the dynamics of deviations from this using a Gaussian process dynamical model. This is largely a "capture and replay" method that admits a small amount of user editing to the motion.

None of the above methods captures realistic full body soft-tissue dynamics or the correlations between body shape and deformation.

3 Dynamic Human Shape Capture: Methods

Dyna relies on two technologies to capture and process the data of full bodies in motion: 4D scanning and 4D mesh alignment. Together we call these **4cap**, for 4D motion capture. 4Cap is like motion capture in that it provides high-framerate 3D data about the body surface that is in correspondence over time. The distinction between 4cap and mocap is that, rather than registering a sparse set of markers across time, we register a dense set of vertices corresponding to a base template mesh. This is similar to previous markerless full-body performance capture systems [de Aguiar et al. 2008; Stark and Hilton 2007] but focuses on full-body soft-tissue deformations with significantly higher spatial and temporal resolution and accuracy. We describe the technology below. Dyna is built on top of a SCAPE model [Anguelov et al. 2005], which we also summarize.

3.1 4D scanning

To model human soft-tissue dynamics we first need to be able to capture it. To that end, we use a custom-built multi-camera active stereo system (3dMD LLC, Atlanta, GA) to capture temporal sequences of full-body 3D scans at 60 frames per second (fps) (Fig. 2). We find that 60fps is sufficient to capture soft-tissue deformations due to short term impulsive external forces such as ground contact and the secondary effects of waves propagating through the tissue (see **Supplemental Video**).

The system uses 22 pairs of stereo cameras, 22 color cameras, 34 speckle projectors and arrays of white-light LED panels. The projectors and LEDs flash at 120fps to alternate between stereo capture and color capture; we do not use the color information here. The projected texture pattern makes stereo matching more accurate, dense, and reliable compared with passive stereo methods. The stereo pairs are arranged to give full body capture for a range of activities, enabling us to capture people in motion; see Fig. 2. The system outputs 3D meshes with approximately $150,000$ vertices on average. Example meshes are shown in Fig. 3.

3.2 Data collection.

For training, we scanned 10 subjects (5 men and 5 women) of various shapes; Fig. 3 shows one scan for each subject. All training subjects were professional models working under a modeling contract. Additionally, all subjects gave their informed written consent for the analysis and publication of their 3D scan data including images and scans of their faces. To test generalization, we scanned an additional 3 men and 3 women, all non-professionals.

Subjects wore minimal form-fitting clothing (bikini briefs for both men and women and a sports-bra for women). To elicit soft-tissue motions we defined a protocol with 14 motions that included various "jiggling" or "shaking" motions as well as activities such as jumping, hopping and running in place that involve impact with the ground. Each capture began with the subject in an "A" pose to simplify automatic mesh alignment.

Since deformations due to dynamics depend on the physical properties of the adipose tissue, muscle, and skin we chose subjects to span different levels of BMI, from low BMI (20.5 for men, 20.7 for women) to high BMI (47.7 for men, 34.9 for women); see Fig. 3.

We captured roughly $40,000$ training scans where there is significant motion. This is split into separate male and female sets, each with about $20,000$ scans. These were further split into various training and testing sets as explained in Sec. 6. We captured a further test set of $\sim 30,000$ scans of 6 subjects to test generalization (Sec. 6).

3.3 Body Modeling: BlendSCAPE

SCAPE [Anguelov et al. 2005] represents body shape in terms of 3×3 deformation matrices [Sumner and Popović 2004] that transform triangles, t, in a template mesh, $\mathcal{T}$, into corresponding triangles in an instance mesh, $\mathcal{M}$; the reader is referred to [Anguelov et al. 2005] for details.

We define the body pose, $\boldsymbol{\theta}$, as the axis-angle representation of relative rotations between body parts. To reconstruct a mesh, $\mathcal{M}$, three types of deformation gradients are applied to the triangles, t, of the template mesh, $\mathcal{T}$. **1)** Pose dependent deformations, $\mathbf{Q}_t(\boldsymbol{\theta})$, are a linear function of pose parameters, $\boldsymbol{\theta}$; see [Anguelov et al. 2005]. To learn this linear function we captured and registered approximately 1800 scans of people in a wide variety of static poses. **2)** Identity-dependent transformations, $\mathbf{S}_t(\boldsymbol{\beta})$, which are a linear function of a vector of body shape coefficients, $\boldsymbol{\beta}$. **3)** Rigid rotations, $\mathbf{R}_{l[t]}(\boldsymbol{\theta})$, which are the absolute rotation for body part $l[t]$ obtained by accumulating the relative rotations along the kinematic chain from the root. Here we use a variant called BlendSCAPE [Hirshberg et al. 2012] in which the deformation of a triangle is a linear combination of several parts that influence the triangle

$$\mathbf{R}_t^*(\boldsymbol{\theta}) = \sum_i w_{ti} \mathbf{R}_i(\boldsymbol{\theta}) \tag{1}$$

where the w_{ti} are blend weights that say how much part i influences triangle t. The sum is over all the parts in the kinematic tree, however, the blend weights are sparse and only a few parts influence any triangle (cf. standard linear-blend skinning [Lewis et al. 2000]). The part segmentation and blend weights are artist-defined.

To train Dyna we use sequences of meshes, indexed by k, corresponding to each subject, j. Specifically, given the edges $\hat{\mathbf{v}}_{t,e}$, where $e \in \{0,1\}$, of each triangle t in the template, we represent the edges, $\mathbf{v}_{t,e}$, of triangle t belonging to the k-th mesh of subject j as

$$\mathbf{v}_{t,e}(\boldsymbol{\beta}^j, \boldsymbol{\theta}_k^j) = \mathbf{R}_{l[t]}^*(\boldsymbol{\theta}_k^j)\mathbf{S}_t(\boldsymbol{\beta}^j)\mathbf{Q}_t(\boldsymbol{\theta}_k^j)\hat{\mathbf{v}}_{t,e}. \tag{2}$$

To form a watertight mesh, SCAPE stitches the triangles together by solving for the mesh that best matches the deformed edges; see [Anguelov et al. 2005].

To represent different body shapes we train separate male and female models using approximately 2000 laser scans for each gender from the US and European CAESAR datasets [Robinette et al. 2002]. After aligning the scans with the template [Bogo et al. 2014; Hirshberg et al. 2012], we compute the mean deformation, $\boldsymbol{\mu}_S$, and subtract this from all training deformations. We then take these triangle deformations and stack them in a vector. We perform PCA on the matrix of shape training vectors and take the leading eigenvectors to form a low-D shape subspace, $\mathbf{U}_S$. The product of the shape parameters, $\boldsymbol{\beta}$. and the basis vectors, $\mathbf{u}_S$, in $\mathbf{U}_S$ linearly approximates a wide variety of body shape deformations from the mean shape, $\boldsymbol{\mu}_S$. We typically use $S = 300$ shape components.

3.4 Mesh Alignment

The first step in building Dyna brings all the temporal 3D scans, $\mathcal{S}_k^j$, for all subjects, j, into correspondence by aligning (registering) a 3D body template to all of them. The template is a watertight triangulated mesh with 6890 vertices and $13,776$ triangles. This registration is performed in two steps. First, a subject-specific shape model, $\mathbf{S}^j$, is computed based on a subset of scans where the person is static. Second, all scans, $\mathcal{S}_k^j$, are aligned using the subject-specific shape models, which makes optimization in the second step substantially faster and better behaved. The process is fully automatic.

3.4.1 Step 1: Subject-specific model creation

The goal of the first step is to compute the shape deformations, $\mathbf{S}^j$, for each subject, j, to capture their detailed body shape more accurately than with the low-D subspace approximation, $\mathbf{S}(\boldsymbol{\beta}^j)$, in Eq. (2). Each subject is captured performing a variety of movements where each movement begins in the A-pose. We take the first frames of each sequence, giving a set of frames, $\mathcal{A}$. We register these scans $\mathcal{S}_k^j, k \in \mathcal{A}$, to a common template using a method similar to [Bogo et al. 2014] that regularizes the aligned templates to be similar to a BlendSCAPE model. This gives a set of aligned template meshes, $\mathcal{M}_k^j$, and their poses, $\boldsymbol{\theta}_k^j$, $k \in \mathcal{A}$. We then compute the mesh that best explains all the individual aligned templates in a regularized least squares sense.

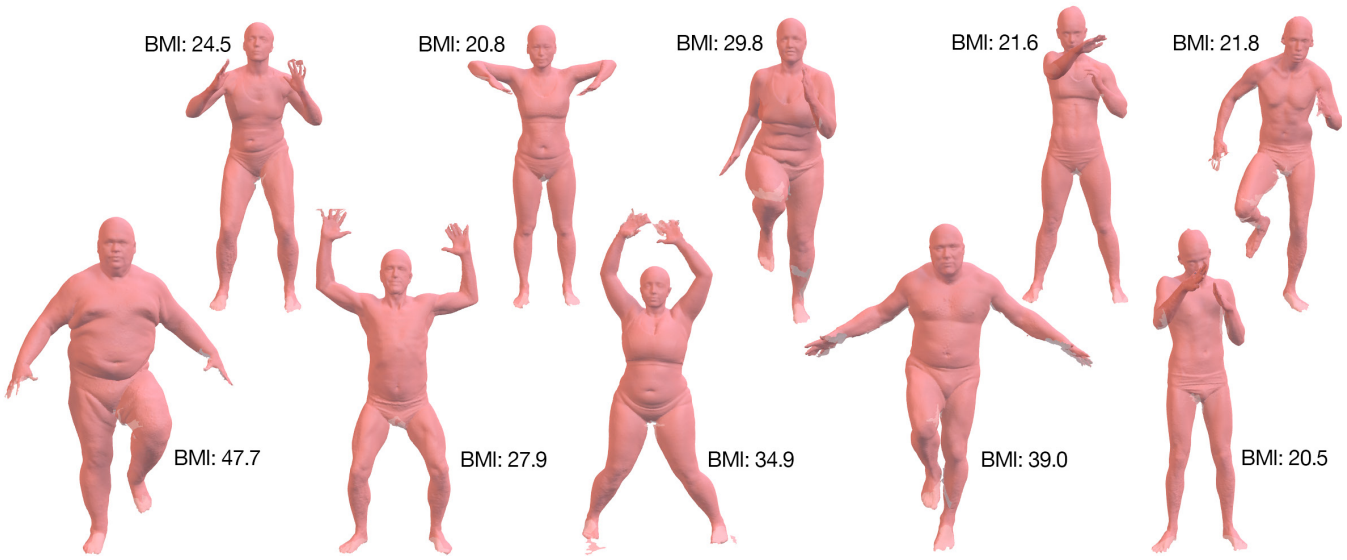

Figure 3: Data capture. *Raw scans of 10 professional models with their BMI. One time instant, from a sequence, is shown for a variety of motions.*

Since we do this for a specific subject, j, we drop the subject index here. Specifically we simultaneously solve for the low-D subject-specific body shape parameters, $\boldsymbol{\beta}$, the scan-specific pose parameters, $\boldsymbol{\theta}_k$, and a deformed template, $\mathcal{M}_k$, that minimize

$$\sum_{k \in \mathcal{A}} E(\boldsymbol{\beta}, \boldsymbol{\theta}_k, \mathcal{M}_k; \mathcal{S}_k) \tag{3}$$

with

$$E(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathcal{M}; \mathcal{S}) = w_g E_g + w_c E_c + w_\theta E_\theta + w_d E_d \tag{4}$$

$$E_g(\mathcal{M}; \mathcal{S}) = \sum_{x_s \in S} \rho(\min_{x_m \in \mathcal{M}} \|x_m - x_s\|) \tag{5}$$

$$E_c(\mathcal{M}, \boldsymbol{\beta}, \boldsymbol{\theta}) = \sum_{t,e} w_t \|\mathcal{M}_{t,e} - \mathbf{v}_{t,e}(\boldsymbol{\beta}, \boldsymbol{\theta})\|_F^2 \tag{6}$$

$$E_\theta(\boldsymbol{\theta}) = D_M(\boldsymbol{\theta}; \boldsymbol{\mu}_\theta, \Sigma_\theta) \tag{7}$$

$$E_d(\boldsymbol{\beta}) = D_M(\boldsymbol{\beta}; 0, \Sigma_\beta), \tag{8}$$

where E_g encourages the vertices, x_m, of the deformed template, $\mathcal{M}$, to be close to the scan surface. For all points, x_s, on the surface of the scan, $\mathcal{S}$, we find the closest point, x_m, on the deformed template; where $\rho(\cdot)$ is a robust Geman-McClure penalty function. E_c encourages the template deformations to be close to the best BlendSCAPE model and vice versa; we call this a "coupling" term and it plays an important role in regularizing the deformation of the template $\mathcal{M}$, where $\mathcal{M}_{t,e}$ is the edge, e, of triangle t. The scalar weight, w_t, is set empirically to increase the coupling strength for parts like hands and feet where the scans are noisier.

Note that Eq. (3) uses a single body shape for all the scans of a subject but lets pose vary since not all scans will be in the exact same A-pose. The terms $E_\theta(\boldsymbol{\theta})$ and $E_d(\boldsymbol{\beta})$ are penalty terms that impose a prior on poses and shapes, respectively. Specifically $D_M(x; \boldsymbol{\mu}, \Sigma)$ is the Mahalanobis distance from x to the mean, $\boldsymbol{\mu}$, with covariance Σ. For body shape, Σ_β is a diagonal matrix with the squared singular values estimated via PCA from CAESAR with zero mean. For pose, Σ_θ is learned from our pose training set containing varied poses and the mean pose, $\boldsymbol{\mu}_\theta$, is roughly an A-pose.

Optimization is performed in stages. We first set the regularizer weights such that the optimization initially performs model-only alignment to fit the BlendSCAPE parameters to the scan. Then the regularizer weights are decreased and the data weight increased, allowing the aligned mesh, $\mathcal{M}$, to deform away from the body model to fit the scan.

Since the objective in Eq. (3) is highly non-convex, initialization plays an important role. Here we initialize the poses, $\boldsymbol{\theta}_k$, to $\boldsymbol{\mu}_\theta$, the shape to the mean shape in CAESAR ($\boldsymbol{\beta} = 0$), and the registration edges, $\mathcal{M}_{t,e}$, to the model edges, $\mathbf{v}_{t,e}(\boldsymbol{\beta}, \boldsymbol{\theta}_k)$, estimated in the preceding stage.

Given the aligned meshes we now solve for a single subject-specific mesh that best explains all of them. Specifically, we solve for the subject-specific deformation, $\mathbf{S}^j$, that gives a mesh (Eq. (2)) that minimizes the squared distance to the meshes, $\mathcal{M}_k^j$, subject to a smoothness term that penalizes differences in deformations, $\mathbf{S}_t^j$, for triangles, t, that share an edge. Below we then replace $\mathbf{S}_t(\boldsymbol{\beta}^j)$ in Eq. (2) with $\mathbf{S}_t^j$ to create subject-specific shapes in any pose.

3.4.2 Step 2: Subject-specific sequence registration

Given subject-specific models, $\mathbf{S}^j$, we align the full sequences of scans $\mathcal{S}_k^j$ to the template, while regularizing the solution to the subject-specific model by minimizing

$$E(\mathcal{M}_k^j, \boldsymbol{\theta}_k^j; \mathcal{S}_k^j) = w_g E_g + w_c E_c \tag{9}$$

$$E_c(\mathcal{M}_k^j, \boldsymbol{\theta}_k^j) = \sum_{t,e} w_t \|\mathcal{M}_{t,e,k}^j - \mathbf{v}_{t,e}(\boldsymbol{\theta}_k^j)\|_F^2, \tag{10}$$

where $E_g(\cdot)$ is the same as in Eq. (5) and where

$$\mathbf{v}_{t,e}(\boldsymbol{\theta}_k^j) = \mathbf{R}_{l[t]}^*(\boldsymbol{\theta}_k^j) \mathbf{S}_t^j \mathbf{Q}_t(\boldsymbol{\theta}_k^j) \hat{\mathbf{v}}_{t,e}. \tag{11}$$

Equation (9) is a simplification of Eq. (3). We no longer optimize over the shape parameters $\boldsymbol{\beta}^j$ as the shape is replaced by $\mathbf{S}^j$. Note also that the pose prior, $E_\theta(\cdot)$, is not used. Since our sequences are captured at 60fps, the changes of pose and shape between one frame and the next are relatively small. This makes accurate alignment of sequences more stable and efficient than aligning scans in arbitrary unknown poses. We initialize the optimization at each frame with

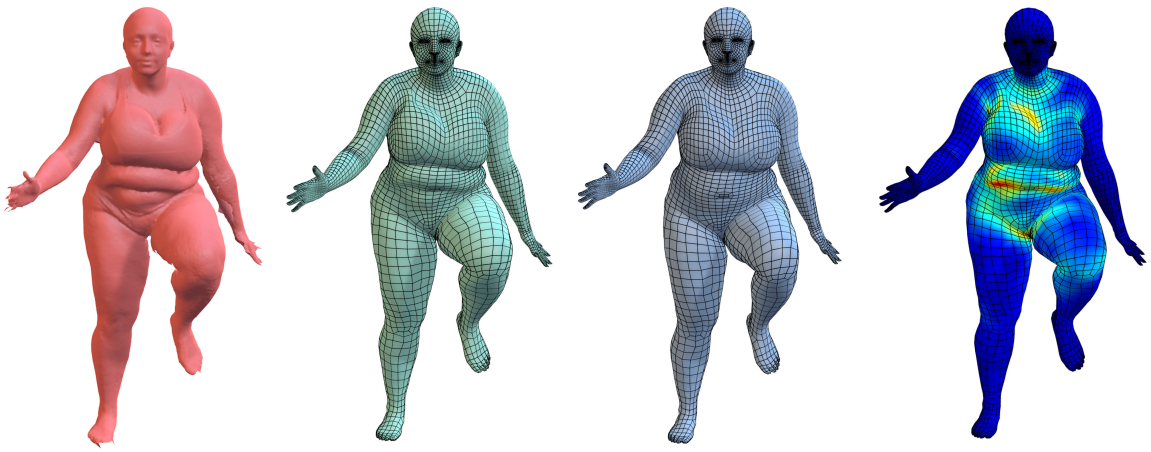

Figure 4: Data pre-processing. *From left to right: 1) an example of a raw scan, 2) template body mesh aligned to the scan 3) subject specific BlendSCAPE body model best fitting the aligned mesh. 4) Frobenius norm distance between the deformations of the aligned template mesh and BlendSCAPE. These differences are color coded (hot colors are large differences). Note that our model is trained from these differences of triangle deformations. This reflects how the aligned body shape deviates from BlendSCAPE.*

the pose from the previous frame and only small changes are required. For the first frame, we use the aligned mesh obtained during the subject-specific model creation step.

3.4.3 Optimization

Both the first and second step objective functions ((3) and (9)) are optimized with Powell's dog leg [Powell 1970]. While the first step requires three stages and typically several optimization iterations, the second uses only one stage and typically converges in less than three iterations (if the motion is not extreme).

3.4.4 Alignment results

The result of the alignment process is $\sim 70,000$ meshes, $\mathcal{M}_k^j$, that are in correspondence with the template, $\mathcal{T}$. Figure 4 shows a representative scan, $\mathcal{S}_k^j$, and the corresponding aligned mesh, $\mathcal{M}_k^j$. Note that we show the quad structure of the template mesh to help the reader visualize the correspondence between meshes. See the **Supplementary Video** for movies showing the temporal alignment. The $40,000$ training meshes are available on line for research purposes along with a video showing all training alignments.

3.5 Dynamic Shape Space

The alignment of all the scans to a set of registered meshes facilitates statistical analysis of soft-tissue deformations. From these meshes we compute the total deformations, $\mathbf{T}_k^j$, which we then split into different causes. Let $\mathbf{v}_{t,e,k}^j$ be an edge of $\mathcal{M}_k^j$. To recover the deformations for $\mathcal{M}_k^j$, we simultaneously solve for pose, $\boldsymbol{\theta}_k^j$, and the total non-rigid deformations, $\mathbf{T}_k^j$, by minimizing (see [Anguelov et al. 2005])

$$\underset{\boldsymbol{\theta}_k^j, \mathbf{T}_k^j}{\arg\min} \sum_t \sum_{e=0,1} \|\mathbf{R}_{l[t]}^*(\boldsymbol{\theta}_k^j)\mathbf{T}_k^j\mathbf{Q}_t(\boldsymbol{\theta}_k^j)\hat{\mathbf{v}}_{t,e} - \mathbf{v}_{t,e,k}^j\|_F^2 + \lambda \sum_{t_1,t_2 \text{ adj}} \|\mathbf{T}_{t_1,k}^j - \mathbf{T}_{t_2,k}^j\|_F^2. \quad (12)$$

The first term minimizes the reconstruction error between the edges, $\mathbf{v}_{t,e,k}^j$, of the aligned meshes, $\mathcal{M}_k^j$, and their BlendSCAPE representation. The second term enforces smooth deformations between adjacent triangles. Note that the smoothness term here is applied only to the shape deformations. This is applied over all adjacent triangles t_1, t_2 adj. In practice, we have found good results optimizing Eq. (12) using alternation; that is first optimizing for pose, $\boldsymbol{\theta}_k^j$, and then optimizing for the total deformations, $\mathbf{T}_k^j$, for every scan, k, of every subject, j.

We assume that this total deformation is the sum of the identity deformation, $\mathbf{S}^j$, of the person, which is constant, and a dynamic shape component, $\mathbf{D}_k^j$, that varies. To find the dynamic part, we compute the average shape deformation, $\bar{\mathbf{S}}^j$, for each subject

$$\bar{\mathbf{S}}^j = \frac{1}{N}\sum_k \mathbf{T}_k^j. \quad (13)$$

Note that $\bar{\mathbf{S}}^j$ may be different from $\mathbf{S}^j$ defined above, which is learned only from static A-poses. Figure 4 shows an example of a subject specific shape posed according to the estimated pose, $\boldsymbol{\theta}_k^j$. This person-specific model lacks dynamic soft-tissue motions as these are averaged out.

Let $\mathbf{D}_k^j = \mathbf{T}_k^j - \bar{\mathbf{S}}^j$ be the residual deformation after subtracting the mean shape from the overal shape. This effectively factors out pose, $\boldsymbol{\theta}_k^j$, pose-dependent deformations, $\mathbf{Q}_t(\boldsymbol{\theta}_k^j)$, and identity $\bar{\mathbf{S}}_t^j$ to focus on what is left. If the BlendSCAPE model accurately captures pose dependent deformations and static shape, then the residual shape deformations should be due only to the deformation of soft tissue. Figure 4 (far right) visualizes these deformation differences between an aligned mesh and the personalized BlendSCAPE model. Here we show color-coded distances (Frobenius norm) between triangle deformations. It is these deformations that we want to model.

From our thousands of high-dimensional deformations $\mathbf{D}_k^j$, we want to learn a model of soft-tissue deformation. To facilitate that, we first reduce the dimensionality of the data via PCA; this is analogous to learning the identity-dependent shape space in SCAPE. Taking the $\mathbf{D}_k^j$ of all subjects, we compute the mean, $\boldsymbol{\mu}_D$, and subtract this from the deformations. We then stack the mean-subtracted deviations at each frame into a vector, form a matrix of these, and perform PCA. Note that we separate the aligned training data by gender giving approximately $20,000$ meshes each. We withhold

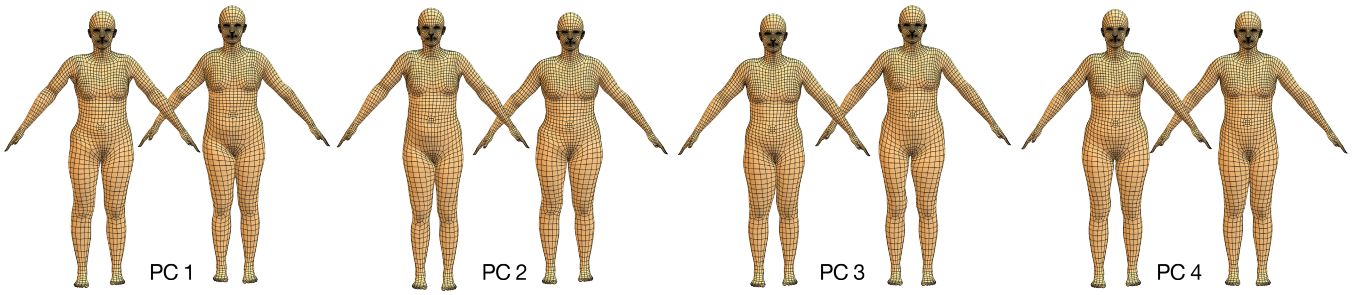

Figure 5: Principal components of soft-tissue deformation. *Several representative components illustrating soft-tissue deformations; pairs of bodies shown at $\pm 5 std$ from the mean. Note the variation in shape in the hips, thighs, and chest. We analyze male and female deformations separately. See* **Supplemental Video**.

half the data for testing and perform PCA on $10,000$ meshes per gender.

We take the D principal components, $\mathbf{U}_D$, corresponding to the largest eigenvalues. Unless otherwise stated, we use $D = 300$ components accounting for 95% of the data variance. Several of these component directions are visualized in Fig. 5; see **Supplemental Video** for movies.

The total non-rigid shape of a given subject is approximated as

$$\hat{\mathbf{T}}_k^j = \bar{\mathbf{S}}^j + \boldsymbol{\mu}_D + \sum_{d=1}^{D} \boldsymbol{\delta}_{d,k} \mathbf{u}_{D,d} \tag{14}$$

where $\mathbf{u}_{D,d}$ is the d-th basis vector in $\mathbf{U}_D$ and the linear coefficients $\boldsymbol{\delta}_{d,k}$ determine the dynamic shape variations. A sequence of such coefficients $\boldsymbol{\delta}_{d,k}$ approximates the soft-tissue motion in our training scans. In the following section we learn a temporal model of how these coefficients evolve with body motion.

4 DYNA

Animating the body with BlendSCAPE produces realistic pose-dependent deformations but the resulting animations lack the natural non-rigid dynamics that occur during human motion. One of our key contributions is to extend BlendSCAPE so that it realistically generates such dynamics. To that end, we first define the total non-rigid deformation of a body as a combination of identity and soft-tissue deformations as

$$\mathbf{T}(\boldsymbol{\beta}, \boldsymbol{\delta}) = \boldsymbol{\mu}_S + \sum_{s=1}^{S} \boldsymbol{\beta}_{s,k} \mathbf{u}_{S,s} + \boldsymbol{\mu}_D + \sum_{d=1}^{D} \boldsymbol{\delta}_{d,k} \mathbf{u}_{D,d} \tag{15}$$

$$= \mathbf{S}(\boldsymbol{\beta}) + \mathbf{D}(\boldsymbol{\delta}). \tag{16}$$

Hence, in Dyna, triangles edges are

$$\mathbf{v}_{t,e}(\boldsymbol{\beta}, \boldsymbol{\theta}_k, \boldsymbol{\delta}_k) = \mathbf{R}^*_{l[t]}(\boldsymbol{\theta}_k) \mathbf{T}_t(\boldsymbol{\beta}, \boldsymbol{\delta}_k) \mathbf{Q}_t(\boldsymbol{\theta}_k) \hat{\mathbf{v}}_{t,e} \tag{17}$$

$$= \mathbf{R}^*_{l[t],k}(\boldsymbol{\theta}_k)(\mathbf{S}_t(\boldsymbol{\beta}) + \mathbf{D}_t(\boldsymbol{\delta}_k)) \mathbf{Q}_t(\boldsymbol{\theta}_k) \hat{\mathbf{v}}_{t,e}. \tag{18}$$

Note that, like SCAPE, this factors deformations due to different causes: specifically, body shape deformations, $\mathbf{S}(\boldsymbol{\beta})$, pose-dependent deformations, $\mathbf{Q}(\boldsymbol{\theta})$, and dynamics-dependent deformations, $\mathbf{D}(\boldsymbol{\delta})$. Below we define the dynamic deformations to depend on velocities and accelerations; that is, they are not dependent on static pose.

Our goal is to produce a realistic animation given a subject shape $\boldsymbol{\beta}$ and a sequence of poses, $\boldsymbol{\theta}_k$. Dynamic deformations are the result of inertia and impact and are therefore related to velocities and accelerations. Hence the input to our model are angular velocity and acceleration $(\dot{\boldsymbol{\theta}}_k, \ddot{\boldsymbol{\theta}}_k)$ at all the joints in the body as well as the velocity and acceleration of the root of the body at time k, $(\boldsymbol{v}_k, \boldsymbol{a}_k)$. Velocities and accelerations are computed using forward finite differences at 60 fps.

To model the typical ripples of soft tissue due to an impact we use an auto-regressive model of order M. Let $\{\hat{\boldsymbol{\delta}}_{k-1} \dots \hat{\boldsymbol{\delta}}_{k-M}\} \in \mathcal{Y}$ be the set containing the history of estimated low-D dynamic deformation coefficients. We use $M = 2$ in all our experiments. Altogether, let $\mathbf{x}_k = \{\dot{\boldsymbol{\theta}}_k, \ddot{\boldsymbol{\theta}}_k, \boldsymbol{v}_k, \boldsymbol{a}_k, \hat{\boldsymbol{\delta}}_{k-1}, \hat{\boldsymbol{\delta}}_{k-2}\} \in \mathcal{X}$ denote the set of inputs to our model. We call this the dynamic control vector.

Finally, we note that the dynamic deformations depend on body shape and the amount of fat on the body; generally, a person with high BMI will jiggle more than one with low BMI. This means that our model of soft-tissue motion will depend on the shape identity coefficients, $\boldsymbol{\beta}$.

The goal is then to learn a function $f : \mathcal{X} \times \mathbb{R}^S \mapsto \mathbb{R}^D$ that predicts the low dimensional representation of the dynamics at time k from $\mathbf{x}_k$ and the subject static shape coefficients $\boldsymbol{\beta} \in \mathbb{R}^S$. Hence, the edges, $\mathbf{v}_{t,e}(\boldsymbol{\beta}, \boldsymbol{\theta}_k, \boldsymbol{\delta}_k)$ of triangle t are reconstructed at frame k as

$$\mathbf{R}^*_{l[t]}(\boldsymbol{\theta}_k)(\mathbf{S}_t(\boldsymbol{\beta}) + \mathbf{D}_t(f(\mathbf{x}_k, \boldsymbol{\beta}))) \mathbf{Q}_t(\boldsymbol{\theta}_k) \hat{\mathbf{v}}_{t,e}. \tag{19}$$

This generalizes SCAPE to include soft-tissue deformations.

4.1 Predictive Model

We now learn $f(\mathbf{x}_k, \boldsymbol{\beta})$, which takes as input the dynamic control vector and the static shape of a given subject and predicts the non-rigid dynamic deformations. Let the n-th element, $\mathbf{x}_k$, in the dynamic control vector be denoted as $\mathbf{x}_k[n]$ (i.e. $\mathbf{x}_k[0] = \dot{\boldsymbol{\theta}}_k$). We start with formulating the problem of learning a temporal function, $g_j(\mathbf{x})$, that is specific to subject j.

Single-person model. We adopt an auto-regressive model with exogenous inputs (ARMAX) relating soft-tissue deformations to the dynamic control vector of subject j

$$\hat{\boldsymbol{\delta}}_k^j = g_j(\mathbf{x}_k) = \sum_{n=1}^{4} \mathbf{B}_n^j \mathbf{x}_k[n] + \sum_{l=1}^{M} \mathrm{diag}(\mathbf{a}_l^j) \hat{\boldsymbol{\delta}}_{k-l}^j + \mathbf{b}^j \tag{20}$$

where the parameters of the model are the constant offset $\mathbf{b}^j$, the matrices $\{\mathbf{B}_l^j\}_{i=1}^4$ and the vectors of auto-regressive (AR) coefficients $\{\mathbf{a}_l^j\}$. Since the PCA coefficients are uncorrelated (in the

training set) we force the auto-regressive matrices to be diagonal; this significantly reduces over-fitting and the number of parameters to be learned. In addition, this allows to train and predict every coefficient independently. This AR linear model captures the angular velocity and angular accelerations of the body parts. Note that the model is linear in the parameters. Hence, it can be efficiently learned given a training set.

This model, when trained using data from a single subject, is already quite powerful and can synthesize non-rigid deformations for new motions of the same subject. This is useful for modeling an actor and then editing their motions or animating them in new ways. Such a model, however, does not generalize well to deformations due to very different body shapes. To address this, one could train the model in Eq. (20) using more than one subject. This reduces over-fitting to a single subject but the model produces deformations that look like the average of deformations of the subjects in the training data.

Instead we want the dynamics of a heavy person to be explained by a model trained using heavy people, and the dynamics of light person should be explained by a model trained with lighter people. In particular, we make the assumption that the dynamics of people with similar BMI will show similar dynamics. This is of course an approximation. For example, BMI is well known to be overestimated in muscular people. Additionally, two people with the same BMI may have very different proportions of visceral to subcutaneous adipose tissue, resulting in vary different soft-tissue dynamics. Here we find BMI to be a useful and intuitive variable for animation control; 4D scans with more precise data about the type and distribution of adipose tissue are not available.

Mixture model. Given a training dataset consisting of N_{subj} subjects, we model the coefficients of a new subject using a mixture model

$$\hat{\boldsymbol{\delta}}_k = f(\mathbf{x}_k, a) = \frac{1}{Z} \sum_{j}^{N_{\text{subj}}} \exp\left(-\frac{\|a^j - a\|}{\tau}\right) g_j(\mathbf{x}_k), \quad (21)$$

where a indicates the BMI of the body we want to animate and $Z = \sum_j \alpha_j$ is a constant to make the sum of the mixture weights $\alpha_j = \exp(-\|a^j - a\|)/\tau)$ equal to one after normalization. Note that a^j is the BMI of the j-th subject in the training database. The parameter τ is a kernel width that controls the influence of each of the trained subject models in the prediction. For $\tau \mapsto \infty$ one obtains equal weights and therefore the model does not depend on the test subject BMI, and for $\tau \mapsto 0$ one obtains support from the subject model in the training data that is closest in BMI. One can also view this as a radial basis function (RBF) approximation in which we compute the weighted distance to each subject using the kernel above. Each subject's BMI defines an RBF center for scattered data interpolation.

An advantage of the model is that BMI is a meaningful quantity that, for example, an animator can intuitively control. That is, one can easily change the predicted dynamics by varying BMI. In some situations however, one wants to animate a given mesh without any knowledge of the BMI of that subject. Therefore, given a labeled dataset of static meshes [Robinette et al. 2002] we learn a function that estimates the BMI of a given static shape. In particular, we learn a linear mapping $\mathbf{C}$ to obtain estimates of weight, w, and height, h. Let $\mathbf{f} = [w, h]$, then $\hat{\mathbf{f}} = \mathbf{C}\boldsymbol{\beta}$. [2] From these estimates of weight and height we can compute the standard BMI measure, defined as $a = \frac{w}{h^2}$ and consequently we can compute the distance inside the exponential in Eq. (21). Let $\mathrm{d}_{\text{BMI}}(\cdot) : \mathbb{R}^S \times \mathbb{R}^S \mapsto \mathbb{R}$ denote the function that takes two vectors of static body shape coefficients and returns the absolute BMI distance.

Consequently, the mixture in Equation (21) can be written as

$$f(\mathbf{x}, \boldsymbol{\beta}) = \frac{1}{Z} \sum_{j=1}^{N_{\text{subj}}} \exp\left(-\frac{\mathrm{d}_{\text{BMI}}(\boldsymbol{\beta}^j, \boldsymbol{\beta})}{\tau}\right) g_j(\mathbf{x}) \quad (22)$$

from which we predict dynamics at time k using $\hat{\boldsymbol{\delta}}_k = f(\mathbf{x}_k, \boldsymbol{\beta})$ which is a function of static shape and motion as desired.

4.2 Dyna Training

Let $\mathbf{p}^j = \{\mathbf{b}^j, \{\mathbf{B}_n^j\}_{n=1}^4, \{\mathbf{a}_l^j\}_{l=1}^M\}$ be the parameters of subject model j. Then the model parameters to be learned are are $\mathbf{p} = \{\mathbf{p}^j\}_{j=1}^{N_{\text{subj}}}$ and the kernel width, τ, which is shared across all subjects. To that end, we construct a training dataset $\mathcal{D} := \{(\mathbf{x}_k^j, \boldsymbol{\delta}_k^j)\}, \forall j, k$ containing pairs of dynamic control vectors, $\mathbf{x}_k^j$ and the ground truth soft-tissue deformation coefficients, $\boldsymbol{\delta}_k^j$. Let $\mathcal{D}^j$ denote a subset of the complete training set, $\mathcal{D}$, consisting of training data of only subject j.

Recall, $\mathbf{x}_k = \{\dot{\boldsymbol{\theta}}_k, \ddot{\boldsymbol{\theta}}_k, \boldsymbol{v}_k, \boldsymbol{a}_k, \hat{\boldsymbol{\delta}}_{k-1}, \hat{\boldsymbol{\delta}}_{k-2}\} \in \mathcal{X}$. The root velocity and acceleration are computed using forward differences giving estimates in world coordinates $\boldsymbol{v}^W, \boldsymbol{a}^W$. Velocities and accelerations are then transformed to the root body coordinates by $\boldsymbol{v}_k = \mathbf{R}_k^T \boldsymbol{v}_k^W$, $\boldsymbol{a}_k = \mathbf{R}_k^T \boldsymbol{a}_k^W$, where $\mathbf{R}_k$ is the root orientation of the body; this provides invariance to body orientation. To compute the linear coefficients, $\boldsymbol{\delta}_k^j$, we compute $\mathbf{D}_k^j = \mathbf{T}_k^j - \bar{\mathbf{S}}^j$ and then project the vectorized deformations $\mathbf{D}_k^j$ onto the low-D dynamic subspace, $\mathbf{U}_D$.

To train the mixture in Eq. (21), we begin by training the N_{subj} subject specific models, $g_j(\mathbf{x}, \mathbf{p}^j)$ separately using $\mathcal{D}^j$. We define the empirical risk using a squared loss as

$$\mathcal{R}(\mathbf{p}^j; \mathcal{D}^j) = \frac{1}{|\mathcal{D}^j|} \sum_{k=1}^{|\mathcal{D}^j|} \|\boldsymbol{\delta}_k^j - g_j(\mathbf{x}_k; \mathbf{p}^j)\|^2. \quad (23)$$

Since $g_j(\cdot)$ is linear, we compute the optimal parameters, $\mathbf{p}^j$, by simply solving a system of linear equations.

Even with careful regularization, such models will be very specific to the given subject and will not generalize well. Training on all the subjects together will produce a generic model that generalizes but is non-specific. To train subject-specific models that generalize we train them using all the data but *bias* the solution towards the particular subject's data. That is we fit the subject-specific parameters using all the data but weight the data for that subject more highly. To do so, modify the empirical risk and minimize what we call *weighted risk*

$$\mathcal{R}_w(\mathbf{p}^j; \mathcal{D}) = \sum_{i=1}^{N_{\text{subj}}} \sum_{k=1}^{|\mathcal{D}^i|} \frac{1}{|\mathcal{D}^i|} \|w_i(\boldsymbol{\delta}_k^i - g_i(\mathbf{x}_k^i; \mathbf{p}^j)\|^2 \quad (24)$$

$$= \sum_{i=1}^{N_{\text{subj}}} w_i^2 \mathcal{R}(\mathbf{p}^j; \mathcal{D}^i) \quad (25)$$

where w_i are per-subject weights that control the overall influence of a particular dataset on the learned model. To bias the model to subject j, we set $w_i = \sqrt{2(N_{\text{subj}} - 1)} : i = j$ and $w_i =$

[2] Actually since weight is linear with volume, the weight is corrected as $w' = w^{\frac{1}{3}}$ to obtain a more linear relationship with the shape coefficients.

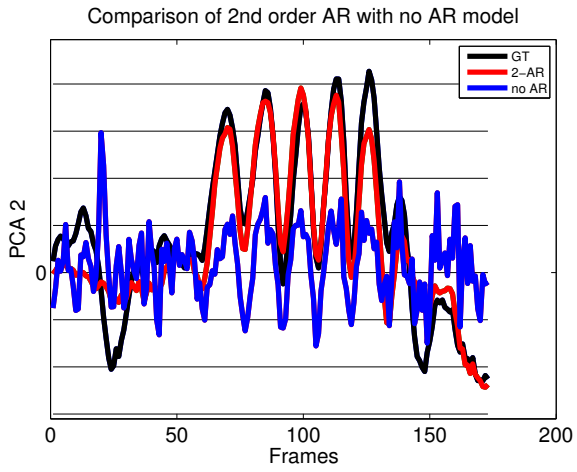

Figure 6: Dyna prediction. *True (GT, black) and predicted (red) values of the second dynamic shape coefficient are shown for a "jiggle on toes" motion. The red curve shows the full 2nd order AR model prediction (2-AR). For comparison the blue curve shows the prediction with a 0th order model (no AR).*

$1 : i \neq j$ so that the weight of the data for subject j has twice the influence in the training as the rest of the subjects' data together. By minimizing the weighted risk we can train a subject model using *all* the data available which heavily reduces over-fitting while keeping the properties of the subject-specific deformations. Note that the weighted risk is still linear and can be minimized as before using weighted least squares.

We are only left with finding the optimal width, τ, of the kernel in Eq. (21). This could be done by optimizing τ to minimize prediction error using a leave-one-out strategy. With only 5 subjects per gender, however, we found that we had insufficient training data for this approach. Consequently we manually set $\tau = 2.0$ for all experiments.

5 Computational time

The only computationally expensive elements of Dyna are 4D alignment and the PCA computation. Mesh alignment (Eq. (8)) takes between 20 and 30 seconds per frame on a 2.40GHz Intel Xeon CPU E5620. 16 alignments were run simultaneously on a single computer, so all the training alignments were computed in less than a day in a single machine. Factorization of deformation gradients into pose and dynamic components (Eq. (12)) takes 5 seconds per frame. PCA (Sec. 3.5) takes around 35 minutes for 10,000 scans. Training time after computing PCA is fast since it works with the low-dimensional representation; it is linear with the number of training points. For 20,000 training data points and 5 subjects, it takes 7 seconds to learn the mixture model with no autoregression and it takes 7 minutes with second order autoregression (Sec. 4.2). The autoregressive model takes a bit longer to train due to the fact that we impose a diagonal structure in the autoregression matrices to avoid overfitting. Animation with Dyna involves predicting the dynamic coefficients, which takes around 1ms per frame. The forward kinematics and the stitching process of SCAPE takes 94ms per frame in an unoptimized Python implementation.

6 Experiments

Figure 6 illustrates soft-tissue deformations predicted by Dyna compared with ground truth. It shows the trajectory of one of the dynamic shape coefficients (principal component 2 in this case) during a "jiggle on toes" movement. The subject was not present in the training set and the movement was held-out for training the AR model (generalization Case 4, see Section 6.1). The ground truth is obtained by projecting the aligned test meshes onto the PCA space. The figure compares the predicted coefficient using the second-order AR model against a zero-order model. The zero-order model uses only the angular velocities and accelerations of the limbs and the velocity and acceleration of the torso.

6.1 Generalization

As with any model trained from examples, it is important to evaluate how well it generalizes. To that end, we evaluate generalization to new motions and new shapes in four use cases. In the first two cases, the PCA space is constructed from half the training data for each gender. In these cases we test on the training subjects but the Dyna AR model is tested on held-out motions. These cases correspond to having a known set of actors and generalizing them to perform new motions. The second two cases test on the 6 subjects not present in the training data and correspond to having completely new subjects.

The cases are: *(1) New motion instance*: Given a subject in the database we animate them with a motion that they did not perform in the training set. We train Dyna with all the training data, leaving out the specific sequence of the subject being tested. In this case the training set contains motions similar to the test motion but performed by other subjects. *(2) Completely new motion*: Here we remove a motion completely from the AR training set and then animate our known subjects performing this motion. This tests generalization to new motions. *(3) Completely new subject but known motion*: We evaluate how well the algorithm generalizes to completely new subjects using motions present in training. *(4) Completely new subject and completely new motion*: In this most challenging scenario, neither the motion nor the subject are present in training.

A representative example of Case 1 is shown in Fig. 7. Cases 2,3 and 4 are shown in Fig. 8. In Fig. 7, the comparison between the aligned mesh, BlendSCAPE, and Dyna shows how Dyna better captures the motion of the soft tissue. Note that the Dyna shape is driven only by body shape and the body motion; it is not fit to the aligned scan. Note that Dyna does not replicate the deformations of the ground truth meshes exactly but rather qualitatively; this is what we expect and this makes it difficult to compare Dyna numerically to aligned scans. Rather, Dyna synthesizes a soft-tissue deformation that is visually plausible.

The dynamic nature of the animation is better viewed and compared in the **Supplemental Video**, which shows the four generalization cases. The video results demonstrate that the model generalizes both to new motions and new shapes.

6.2 Soft-tissue deformations with static subjects

Dyna can produce realistic animations of subjects where we only know their static body shape. To this end, we first predict the subject BMI based on the subject static shape. Then, given a motion sequence, Dyna produces different deformations depending on the estimated BMI. This is illustrated in Fig. 9, where we animate static subjects from the CAESAR dataset with a "jiggle on toes" motion.

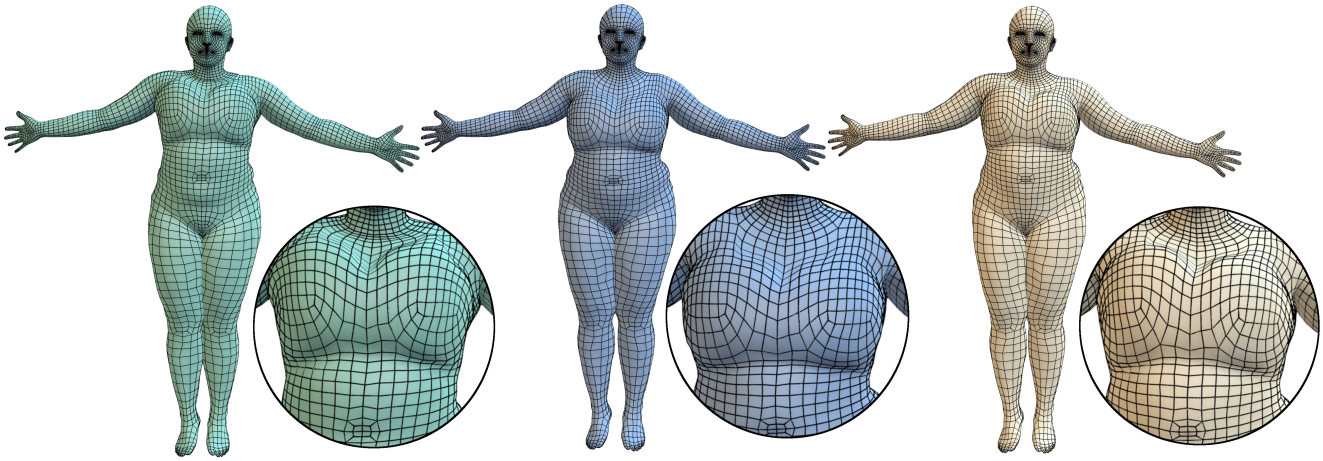

Figure 7: Generalization (Case 1): *Subject performing jumping jacks. The subject is in the training set, but not doing this motion. Left to right: alignment (green), BlendSCAPE (blue), Dyna (white). Detail regions compare the deformation of the stomach and chest. Note how Dyna captures the deformation of the chest and stomach.*

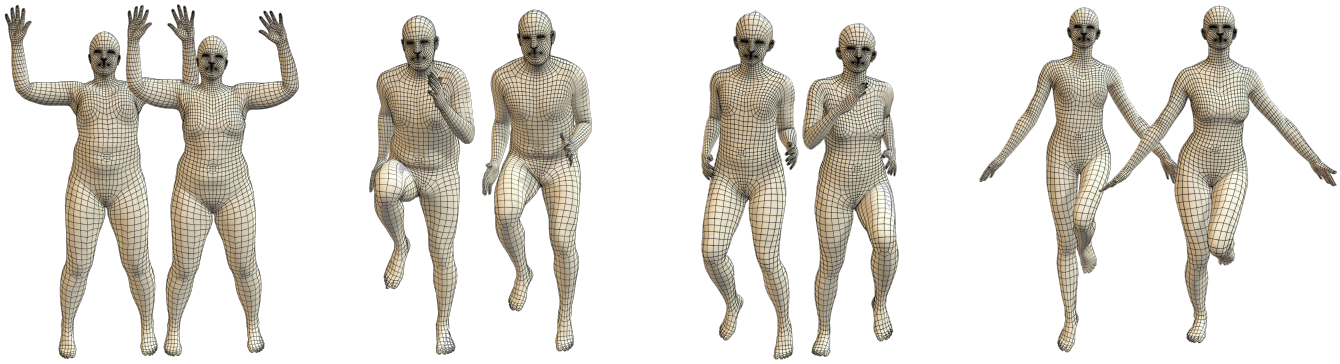

Figure 8: Generalization (Cases 2, 3, 4)*: We show two frames for different subjects and different movements for different generalization cases. First from the left, we show an example of generalization Case 2 for a "jiggle on toes" sequence. Second, we show a generalization Case 3 for running on the spot. The rest of the examples show generalization Case 4, the most challenging scenario where neither the motion nor the subject is present in the training data. Dyna produces realistic deformations even in Case 4.*

6.3 Soft-tissue deformations with weight gain/loss

We allow animators to edit how the soft-tissue deformations behave to produce effects such as exaggeration. For example, an animator can directly control the static and dynamic deformation by changing a character's weight. Increasing the subject's weight, while keeping height and other proportions fixed, results in a change in BMI. To achieve this, we train a linear regressor from a feature vector consisting of height and rectified weight, $\mathbf{f} = [w^{\frac{1}{3}}, h]$, to static shape coefficients, $\boldsymbol{\beta}$ (cf. [Allen et al. 2003]). More features, such as arm length, can be added to the regressor to constrain other properties of the person's shape, although we only use height and rectified weight for this experiment. To increase the person's weight by some amount, we simply compute the new feature $\mathbf{f}_{\text{new}}$ and the new deformation as

$$\bar{\mathbf{S}}_{\text{new}} = \bar{\mathbf{S}}_{\text{old}} + \mathbf{U}_S\mathbf{A}(\mathbf{f}_{\text{new}} - \mathbf{f}_{\text{old}}) \tag{26}$$

where $\mathbf{U}_S$ is the matrix containing the PCA basis vectors, $\mathbf{A}$ is the regression matrix, and $\mathbf{f}_{\text{old}}$ and $\mathbf{f}_{\text{new}}$ are the feature vectors of the orignal body and the desired one respectively. Note that we change the weight and keep the height the same. This linear prediction gives deformations that, when added to the original body, produce a body shape with the desired weight. This results in a new BMI that makes the prediction of soft-tissue deformations change according to Eq. (21). To illustrate this in Fig. 10 we change the original subject's BMI (middle) by decreasing her weight by 30kg (left) and increasing her weight by 30kg (right). The **Supplemental Video** shows how the soft-tissue deformations are attenuated as the weight decreases and accentuated as the weight increases as one would expect. We note however, that Dyna models the dynamics produced by decreasing weight better than by increasing weight. A better model would require more training subjects with high BMI.

6.4 Soft-tissue retargeting

An animator may want to give a character the soft-tissue motions of someone else. One can think of this as the retargeting of soft tissue to a new character. This is straightforward. We simply generate the soft-tissue deformations using one body shape (the heavy one in this case) and add them to the shape of the target body in Dyna model (Eq. (19)); that is, the shape becomes$(\mathbf{S}_t(\boldsymbol{\beta}^i) + \mathbf{D}_t(f(\mathbf{x}_k, \boldsymbol{\beta}^j)))$ where i and j represent different body shapes.

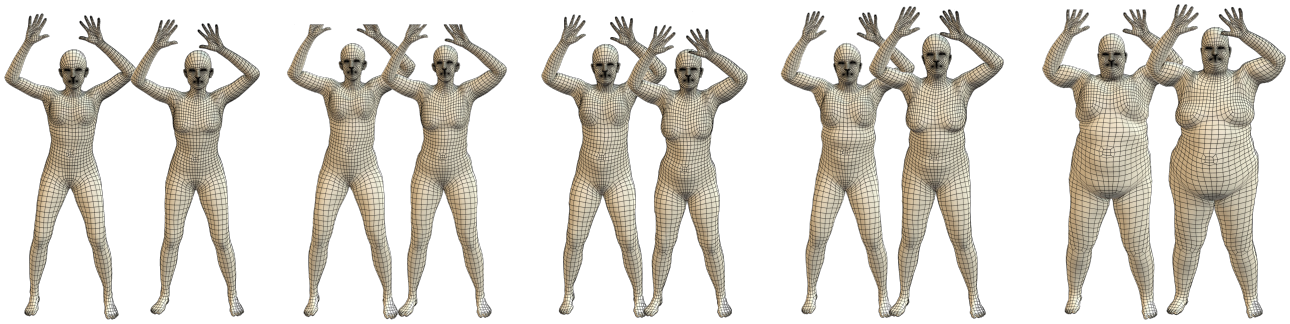

Figure 9: *Dyna can animate subjects where we have only seen their shape. We first predict the BMI of the subject based on their static shape. We then use this to predict the dynamic deformations given a motion sequence. In this example, we animate five subjects from the CAESAR dataset ordered (left to right) from lowest to highest BMI. To further exaggerate the contrast in the deformations we scale the predicted deformations by factors of (from left to right)* $0.9, 1.1, 1.3, 1.5, 1.7$*. Notice how the deformations produced by Dyna differ depending on the subject BMI, see for example how the stomach deformation for the fourth subject is different than the deformation of the left-most subject.*

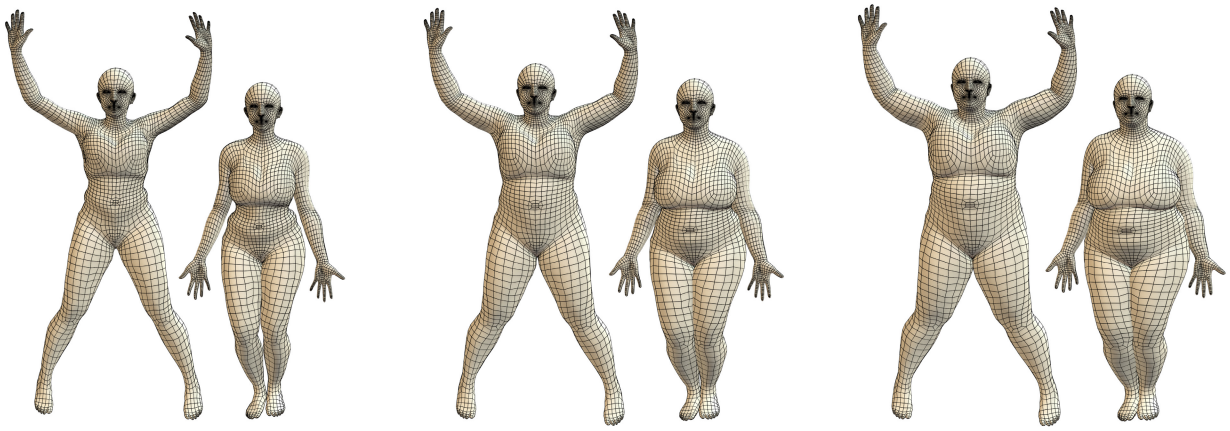

Figure 10: Changing BMI*: Animators can change the BMI of the BlendSCAPE model resulting in a change in body shape and a realistic change in soft-tissue deformations. We start with the subject shape in the center and show a pair of poses. We then change her weight by -30 kg (left bodies) and +30 kg (right bodies). By changing the weight, while keeping height fixed, the BMI changes and therefore the deformations predicted by Dyna change.*

Figure 11 shows two examples in which we transfer the dynamics of a heavy male to an "ogre" and a heavy female to a thin "office worker." To animate a character like the ogre, that cannot be represented in $\mathbf{U}_S$, we simply replace $\mathbf{S}_t(\boldsymbol{\beta})$ in Eq. (19) with the shape we want to animate. This requires that the new shape is in alignment with our template mesh. The **Supplemental Video** illustrates how the ogre deforms plausibly and how putting the dynamics of a heavy woman on a thin character creates interesting effects.

6.5 Exaggeration

An animator might want to exaggerate the dynamic deformations. This can be easily achieved in our model by uniformly scaling the output deformations predicted by Dyna. Figure 12 shows the result of exaggerating deformations $\boldsymbol{\delta}$ by a factor of $\lambda = 2.5$.

6.6 Localized exaggeration

Finally, our model allows local editing of the deformations. An animator can exaggerate certain parts of the body by locally scaling the components of the PCA basis vectors. The artist defines weights, for example by painting them on the mesh, and these correspond to elements in the PCA vectors in the Dyna deformation space. We exaggerate deformations by scaling the corresponding elements of the shape-deformation vector non-uniformly according to the weights. Figure 13 illustrates this exaggeration in comparison with BlendSCAPE and Dyna without exaggeration. The weights here increase the deformation of the chest region while keeping the soft-tissue deformations of the other regions unchanged. Notice how the weighted region undergoes more deformation.

7 Conclusion and Future Work

Dyna provides a new method for automatically animating characters with realistic soft-tissue deformations. It significantly advances the state of the art in terms of realism. This is enabled by novel methods for capturing and registering 3D meshes over time. We train the model from men and women of different BMI and learn a mixture of different models that is able to generalize to new subjects and motions. We develop a training method that allows Dyna to generalize despite a limited set of body shapes and motions. Dyna can be applied for realistic human animation and admits various animator controls for varying the soft-tissue motion.

This technology opens up many directions for future research.

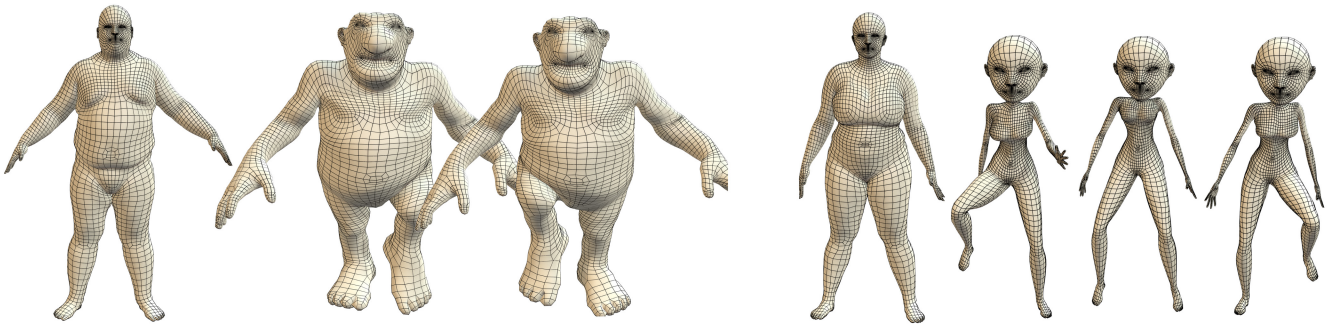

Figure 11: Soft-tissue retargeting*: We can retarget Dyna to new body shapes as long as they share the same topology as our template mesh. This can be used to bring soft-tissue motions to stylized characters. Here we retarget a heavy man to an ogre and a heavy woman to a thin female character. We simply animate the characters and apply the deformations predicted by Dyna.*

Figure 12: Global exaggeration. *We can exaggerate the dynamic deformations predicted by Dyna by scaling $\boldsymbol{\delta}$ by a constant factor. Left: Dyna. Middle: Dyna exaggerated. Right: Dyna exaggerated and unposed. The unposed results show the predicted dynamic deformations in fixed T-pose.*

Figure 13: Localized exaggeration*: Deformations can be exaggerated locally in regions selected by the animator. Weights are illustrated by the color coding (right). The hotter the weight, the larger the exaggeration of the deformation. Left: BlendSCAPE. Middle: Dyna. Right: Local exaggeration.*

Dyna captures only the visible surface effects of body motions on soft-tissue. It would be interesting (though challenging) to relate these visible surface changes to the underlying surface material properties and physical forces. This would require a computationally manageable model of the body's soft tissues. Given such a physics-based model, one might be able to infer the parameters of the model, enabling greater generalization through FEM. It would also be interesting to relate these parameters to individual health; for example, to the distribution and consistency of fat on the body. We could also learn more localized models of soft-tissue deformation using a model like the one in [Neumann et al. 2013b]. Some properties of soft tissue may be local while others are more global so it is unclear whether this approach will be valuable.

Our current capture system supports a wide range of motions but is not able to capture many sports activities that require a larger volume. Extending these scanning methods to large volumes remains an open problem. Modeling people with compression garments and how such garments change soft-tissue motion is an important problem for the garment industry. Finally, we would like to capture the static effects of gravity and other forces using our methods.

8 Acknowledgements

We thank Matthew Loper and Federica Bogo for help with mesh registration and 3D body modeling; Andrea Keller, Senya Polikovsky, and Stephan Streuber for help with data acquisition and technical support; Jon Anning for voice recording, technical support, and web development; Alejandra Quiros Ramirez for web development; and 3dMD for help with the 4D scanning system.

Disclosure: MJB is a founder, shareholder, and member of the board of Body Labs Inc., which is commercializing body shape technology.

References

ALEXA, M., AND MÜLLER, W. 2000. Representing animations by principal components. *Comp. Graph. Forum 19*, 3, 411–418.

ALLEN, B., CURLESS, B., AND POPOVIĆ, Z. 2002. Articulated body deformation from range scan data. *ACM Trans. Graph. 21*, 3 (July), 612–619.

ALLEN, B., CURLESS, B., AND POPOVIĆ, Z. 2003. The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Transactions on Graphics (TOG) 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.

ASSASSI, L., BECKER, M., AND MAGNENAT-THALMANN, N. 2012. Dynamic skin deformation based on biomechanical modeling. In *Proc. 25th Conf. Comp. Anim. and Social Agents*.

AUBEL, A., AND THALMANN, D. 2001. Interactive modeling of the human musculature. In *Proc. Comp. Anim.*, 167–255.

BICKEL, B., BÄCHER, M., OTADUY, M. A., MATUSIK, W., PFISTER, H., AND GROSS, M. 2009. Capture and modeling of non-linear heterogeneous soft tissue. *ACM Trans. Graph. 28*, 3 (July), 89:1–89:9.

BOGO, F., ROMERO, J., LOPER, M., AND BLACK, M. J. 2014. FAUST: Dataset and evaluation for 3D mesh registration. In *CVPR*, 3794 –3801.

CAPELL, S., GREEN, S., CURLESS, B., DUCHAMP, T., AND POPOVIĆ, Z. 2002. Interactive skeleton-driven dynamic deformations. *ACM Trans. Graph. 21*, 3 (July), 586–593.

CAPELL, S., BURKHART, M., CURLESS, B., DUCHAMP, T., AND POPOVIĆ, Z. 2007. Physically based rigging for deformable characters. *Graph. Models 69*, 1 (Jan.), 71–87.

CHADWICK, J. E., HAUMANN, D. R., AND PARENT, R. E. 1989. Layered construction for deformable animated characters. *SIGGRAPH Comput. Graph. 23*, 3 (July), 243–252.

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

DE AGUIAR, E., AND UKITA, N. 2012. Representing mesh-based character animations. *Computers & Graphics 38* (Feb.), 10–17.

DE AGUIAR, E., STOLL, C., THEOBALT, C., AHMED, N., SEIDEL, H.-P., AND THRUN, S. 2008. Performance capture from sparse multi-view video. *ACM Trans. Graph. 27*, 3 (Aug.), 98:1–98:10.

DE AGUIAR, E., SIGAL, L., TREUILLE, A., AND HODGINS, J. K. 2010. Stable spaces for real-time clothing. *ACM Trans. Graph. 29*, 4 (July), 106:1–106:9.

FAN, Y., LITVEN, J., AND PAI, D. K. 2014. Active volumetric musculoskeletal systems. *ACM Trans. Graph. 33*, 4 (July), 152:1–152:9.

GUAN, P., REISS, L., HIRSHBERG, D., WEISS, A., AND BLACK, M. J. 2012. Drape: Dressing any person. *ACM Trans. Graph. 31*, 4 (July), 35:1–35:10.

HAHN, F., MARTIN, S., THOMASZEWSKI, B., SUMNER, R., COROS, S., AND GROSS, M. 2012. Rig-space physics. *ACM Trans. Graph. 31*, 4 (July), 72:1–72:8.

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

HIRSHBERG, D. A., LOPER, M., RACHLIN, E., AND BLACK, M. J. 2012. Coregistration: Simultaneous alignment and modeling of articulated 3D shape. In *ECCV*, vol. 7577 of *LNCS*. Springer, 242–255.

JAMES, D. L., AND PAI, D. K. 2002. Dyrt: Dynamic response textures for real time deformation simulation with graphics hardware. *ACM Trans. Graph. 21*, 3 (July), 582–585.

KARNI, Z., AND GOTSMAN, C. 2004. Compression of soft-body animation sequences. *Computers & Graphics 28*, 25–34.

KIM, T., AND JAMES, D. L. 2009. Skipping steps in deformable simulation with online model reduction. *ACM Trans. Graph. 28*, 5 (Dec.), 123:1–123:9.

KIM, T., AND JAMES, D. L. 2011. Physics-based character skinning using multi-domain subspace deformations. In *Proc. ACM SIGGRAPH/Eurographics Symp. Comp. Anim.*, 63–72.

KRY, P. G., JAMES, D. L., AND PAI, D. K. 2002. EigenSkin: Real time large deformation character skinning in hardware. In *Proc. 2002 ACM SIGGRAPH/Eurographics Symp. Comp. Anim.*, 153–159.

LARBOULETTE, C., CANI, M.-P., AND ARNALDI, B. 2005. Dynamic skinning: Adding real-time dynamic effects to an existing character animation. In *Proc. 21st Spring Conf. Comp. Graph.*, ACM, SCCG '05, 87–93.

LEE, S.-H., SIFAKIS, E., AND TERZOPOULOS, D. 2009. Comprehensive biomechanical modeling and simulation of the upper body. *ACM Trans. Graph. 28*, 4 (Sept.), 99:1–99:17.

LEWIS, J. P., CORDNER, M., AND FONG, N. 2000. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *Proc. SIGGRAPH*, ACM, 165–172.

LOPER, M. M., MAHMOOD, N., AND BLACK, M. J. 2014. MoSh: Motion and shape capture from sparse markers. *ACM Trans. Graph. 33*, 6 (Nov.), 220:1–220:13.

MAUREL, W., WU, Y., MAGNENAT-THALMANN, N., AND THALMANN, D. 1998. *Biomechanical Models for Soft Tissue Simulation*. Springer-Verlag, Berlin.

METAXAS, D., AND TERZOPOULOS, D. 1993. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Trans. Pattern Anal. Mach. Intell. 15*, 6 (June), 580–591.

NEUMANN, T., VARANASI, K., HASLER, N., WACKER, M., MAGNOR, M., AND THEOBALT, C. 2013. Capture and statistical modeling of arm-muscle deformations. *Computer Graphics Forum 32*, 2 (May), 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 (Nov.), 179:1–179:10.

PARK, S. I., AND HODGINS, J. K. 2006. Capturing and animating skin deformation in human motion. *ACM Trans. Graph. 25*, 3 (July), 881–889.

PARK, S. I., AND HODGINS, J. K. 2008. Data-driven modeling of skin and muscle deformation. *ACM Trans. Graph. 27*, 3 (Aug.), 96:1–96:6.

POWELL, M. 1970. A hybrid method for nonlinear equations. In *Numerical Methods for Nonlinear Algebraic Equations*, Gordon and Breach Science, London, P. Rabinowitz, Ed., 87–144.

PRATSCHER, M., COLEMAN, P., LASZLO, J., AND SINGH, K. 2005. Outside-in anatomy based character rigging. In *Proc. 2005 ACM SIGGRAPH/Eurographics Symp. Comp. Anim.*, 329–338.

ROBINETTE, K., BLACKWELL, S., DAANEN, H., BOEHMER, M., FLEMING, S., BRILL, T., HOEFERLIN, D., AND BURNSIDES, D. 2002. Civilian American and European Surface Anthropometry Resource (CAESAR) final report. Tech. Rep. AFRL-HE-WP-TR-2002-0169, US Air Force Research Laboratory.

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

SHI, X., ZHOU, K., TONG, Y., DESBRUN, M., BAO, H., AND GUO, B. 2008. Example-based dynamic skinning in real time. *ACM Trans. Graph. 27*, 3 (Aug.), 29:1–29:8.

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 (July), 417–425.

STARK, J., AND HILTON, A. 2007. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications 27*, 3, 21–31.

SUMNER, R. W., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. *ACM Trans. Graph. 23*, 3, 399–405.

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. and Comp. Graph. 11*, 3 (May), 317–328.

TERZOPOULOS, D., AND WATERS, K. 1990. Physically-based facial modelling, analysis, and animation. *J. Vis. and Comp. Anim. 1*, 2 (Dec.), 73–80.

TSOLI, A., MAHMOOD, N., AND BLACK, M. J. 2014. Breathing life into shape: Capturing, modeling and animating 3D human breathing. *ACM Trans. Graph. 33*, 4 (July), 52:1–52:11.

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