Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2006)
M.-P. Cani, J. O'Brien (Editors)

Learning a correlated model of identity and pose-dependent body shape variation for real-time synthesis

Brett Allen[1,2], Brian Curless[1], Zoran Popović[1], and Aaron Hertzmann[3] †

[1]University of Washington [2]Industrial Light & Magic [3]University of Toronto

Abstract

We present a method for learning a model of human body shape variation from a corpus of 3D range scans. Our model is the first to capture both identity-dependent and pose-dependent shape variation in a correlated fashion, enabling creation of a variety of virtual human characters with realistic and non-linear body deformations that are customized to the individual. Our learning method is robust to irregular sampling in pose-space and identity-space, and also to missing surface data in the examples. Our synthesized character models are based on standard skinning techniques and can be rendered in real time.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Curve, surface, solid and object modeling; I.3.7 [Computer Graphics]: Animation

1. Introduction

One of the main challenges in creating animated human characters for computer graphics is the problem of modeling realistic body shapes. Manually modeling a high-resolution shape that will pass as human is very difficult. Furthermore, a statically-modelled shape must be given a "skeleton" and rigged for animation, so that it can be put into different poses. This rigging process includes *enveloping*, where the initial shape is deformed to follow the underlying skeleton, and then modeling further deformations to account for muscle bulges, bones that slide beneath the skin, dimples, and other changing aspects of the underlying tissues. These tasks must be repeated for each different character ("identity") that is modelled.

To avoid this manual effort, it has become popular to use data capture techniques such as 3D scanning to create human models. However, previous approaches have many restrictions, such as requiring complete or near-complete surface examples, or examples where each joint is moved independently at precise angles (such examples must be made by an artist, since real humans cannot meet such stringent constraints). Moreover, no existing model is able to capture the relationship between different identities and their different pose-dependent deformations. Below we will discuss some previous "example-based" approaches, and introduce our novel approach.

1.1. Related work

We begin by considering approaches for modeling pose-dependent deformations, such as muscle bulges and other anatomical effects. Of course, one option is to directly model the anatomy, as in the work of Scheepers et al. [SPCM97] and many others. The anatomical approach is very promising and flexible, however it requires a lot of manual modeling, and is not suitable for real-time interactions with multiple characters due to the overhead of physical simulation.

In this paper we will focus on example-based methods, where the anatomical effects arise from a generic shape-modifying function that is fit to examples. One approach is to learn a linear function of the joint angle that modifies the shape. For example, the multi-weight embedding of Wang and Philips [WP02] uses a set of per-element weights on the coordinate matrices of each joint to alter the final shape. Mohr and Gleicher [MG03] add extra joints to the skeleton to modulate the shape as a function of pose. Anguelov et al.

† e-mail: allen@cs.washington.edu, curless@cs.washington.edu, zoran@cs.washington.edu, hertzman@dgp.toronto.edu

[ASK*05] learn a linear function that is applied within a deformation transfer framework.

For modeling deformations that are non-linear functions of pose, *scattered-data interpolation* techniques are often used. For example, Lewis et al. [LCF00], Sloan et al. [SRC01], and Kry et al. [KJP02] use radial basis functions to create a function of the joint angles that interpolates a set of example shapes. Allen et al. [ACP02] use a similar *k*-nearest-neighbor interpolation approach.

Scattered-data interpolation is useful when the underlying parameters of variation are exposed (e.g., joint angles), but when considering the problem of modeling an entire population of shapes (e.g., all human faces or bodies), it is unclear what the underlying parameters should be. For this reason, *latent variable models* are often employed, such as Principal Component Analysis (PCA). These techniques model shape variation as a projection onto a low dimensional subspace of shape space. The coordinates of each shape in this subspace are the latent variables. PCA has been applied to character modeling problems, such as analyzing variation among faces [BV99], or among bodies [ACP03, SCMT03].

The next logical step is to have a combined model that can generate any identity in any pose. Sloan et al. [SRC01] extend their scattered-data interpolation method to include identity by adding parameter values such as "male-female". However, their approach risks conflating these two axes if the examples are not precisely spaced in parameter-space. For example, it is possible that bending the elbow could make the character become more female.

Another combined model is the SCAPE approach of Anguelov et al. [ASK*05]. SCAPE learns pose-deformation as a completely separate phenomenon from their PCA-based identity-variation model, and then combines the two modalities when a new shape is synthesized. This model is very powerful, but it cannot capture the correlation between the two modes. For example, when a muscular person bends their arm, the shape change will be the same as when a very thin person bends their arm.

Multi-linear approaches, such as Vlasic et al. [VBPP05] use multi-linear algebra to extend latent variable techniques to handle multiple modalities, such as the identity, expression, and viseme of a face. In principle, the same approach could be applied to bodies. However, sampling body poses is much harder than sampling expressions, because many more samples are needed, and it is difficult to control exactly what poses are captured (e.g., it is hard to request that a subject should achieve a precise shoulder rotation angle). In addition, body deformations are very localized (moving one joint affects only a few of the vertices of the body), which would be inefficiently represented as full tensors.

1.2. Overview

Our novel approach is to take the best features of scattered-data interpolation and latent variable models, and combine them into a hybrid model. Rather than building a latent variable model of just shape, we will build a latent variable model that includes the full set of interpolation keys needed to generate a model in any pose. Using this approach, we encapsulate the correlation between pose and identity, while keeping these two modalities from being conflated. We will present a method for learning our model given an arbitrary set of examples of different individuals in different poses. In addition, unlike previous methods, our system can incorporate incomplete surfaces, where only part of the full surface is observed.

To train our model we require a large corpus of 3D data that covers the range of identity and pose variation. We used 44 subjects from the CAESAR project (see Allen et al. [ACP03]) who were captured in a standard standing and seated pose, each with 74 landmark positions labelled. For an additional 5 subjects, we repeated the CAESAR scanning process, but with a total of 16 different poses. These five subjects were selected to cover a variety of height and weight combinations. Finally, we included scans of a single person in 69 poses (as described by Anguelov et al. [ASK*05]).

The contributions of this paper are broken into three parts. First, in Section 2 we describe the *enveloping* problem, and introduce our modifications for *corrective enveloping*. In Section 3, we will discuss the *matching* problem, where a consistent shape representation is generated for each of the examples. Then in Section 4, we will show how we can learn pose deformations and identity variation from the data.

2. Character representation

In this section, we describe our representation for shape, skeleton, enveloping, and corrective enveloping. Our goal is to choose a representation that is expressive enough to capture the phenomena we are trying to model, yet simple enough that it can be learned automatically from examples.

2.1. Shape and skeleton

To begin with, we create a skeleton hierarchy to approximate the human body's articulations. This skeleton $\mathcal{S}$, shown in Figure 1, consists of 30 rotations and 22 translations. These transformations comprise the degrees of freedom (DOFs) that describe any particular skeleton. We divide the skeletal DOFs into two groups. The first group is the skeleton parameters, $\mathbf{b}$, which consist of the DOFs that are intrinsic to an identity and do not typically vary over time, such as the bone lengths. The second group is the pose, $\mathbf{q}$, which consists of the remaining DOFs, primarily the joint angles.

Our skeleton is constrained such that most of the translation elements must occur along a particular axis, as illustrated in Figure 1. These constraints reduce ambiguities in an observed skeletal structure. We found it necessary to add in a couple of transformations that are not typically used by animators: the carrying angles of the elbow and knee. The

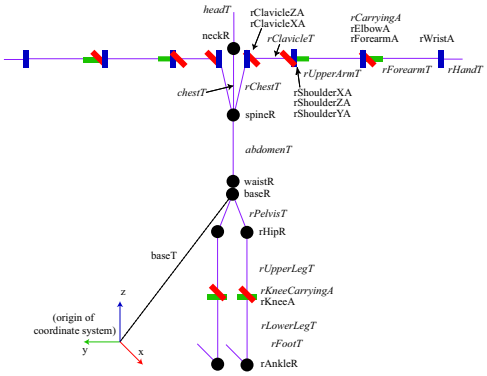

Figure 1: *Articulated skeleton hierarchy. The circles represent free rotations, the bars represent single-axis rotations, and the lines represent translations. The italicized joint names are intrinsic to a particular identity (the skeleton parameters), whereas the values of the other transformations may vary with pose.*

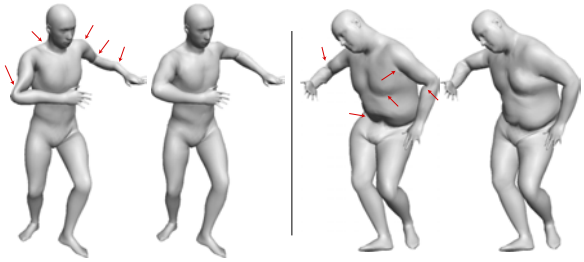

Figure 3: *Comparison between SSD enveloping only (left in each pair), and SSD with corrective enveloping (right in each pair). Notice the increased realism and corrected artifacts, particularly in the regions indicated by arrows.*

carrying angle is a fixed rotation about the axis that is perpendicular to the axes of elbow flexion and twist. That is, if the arms are lowered, the carrying angle bends the arms outward from the body. This angle is intrinsic to an identity and can be different on the left and right sides. Without these degrees of freedom, we were unable to accurately fit our skeletons to the scanned examples.

We assign an index j to each skeletal transformation, and for a particular pose $\mathbf{q}$ and skeleton parameters $\mathbf{b}$ we denote its coordinate frame as a 4×4 matrix $\mathbf{M}_{\mathbf{q},\mathbf{b},j}$.

We represent the shape of a character using a triangle mesh, $\mathcal{M}$, of 7000 vertices, with vertex positions $\mathbf{v}_i$. Our mesh is topologically symmetrical across the sagittal plane; that is, each point on the left side has a corresponding point on the right side.

2.2. Enveloping

The most common approach to enveloping is called Skeleton Subspace Deformation (SSD) [MTT91], also called "Linear blend skinning" [MG03]. The essence of SSD is that each vertex $\mathbf{v}_i$ on the body derives a local position relative to each bone in some canonical pose, called the dress pose, $\bar{\mathbf{q}}$. A weight $\mathbf{s}_{i,j}$ is associated with each vertex and joint. Typically most of the weights are zero, since only a couple of joints influence any particular vertex. To determine the vertex's position in a new pose, one determines that vertex's new position as if it were rigidly attached to each bone, and takes a linear combination of the resulting positions:

$$\mathbf{v}_i = \sum_j \mathbf{s}_{i,j} \mathbf{M}_{\mathbf{q},\mathbf{b},j} \mathbf{M}^{-1}_{\bar{\mathbf{q}},\mathbf{b},j} \bar{\mathbf{v}}_i \qquad (1)$$

Our set of enveloping weights is shown in Figure 2b. We will discuss how these weights are determined in Section 4.1.2, but for now we highlight some of our choices in selecting enveloping regions. We modeled most of the free rotations as a single quaternion joint with 4 DOFs, however we split the shoulder into three Euler angles. Our observation is that elevation and abduction of the shoulder affects vertices near the joint center, but twisting at the shoulder about the humeral axis causes a deformation that is distributed along the upper arm. Therefore, we created one enveloping region for elevation/abduction, and two for the shoulder twist: a 50%-twist region at the top of the upper arm and a 100%-twist region below. We split the forearm twist in the same way. Splitting twist into multiple joints is often used by animators, and is part of the technique published by Mohr and Gleicher [MG03].

The primary advantage of SSD is its simplicity; in fact it can be easily implemented in hardware [KJP02, JT05]. However, SSD suffers from many problems, such as severe volume loss near joints, and inflation in regions far from joints. To a certain extent, these artifacts may be ameliorated by manually adjusting the enveloping weights, however, due to the linear nature of the SSD, there is often no set of enveloping weights that can correct the problems. For this reason, many alternatives to SSD have been proposed, such as free-form deformation lattices [SK00], spherical blend skinning [Kv05], and various proprietary solutions.

Despite the improvements offered by these more complex methods, ultimately no anatomically-naïve enveloping model can suffice to model pose-dependent deformations, due to the complex nature of muscles, bones, and other tissues of the human body. Consequently, it will always be necessary to include corrections to the underlying enveloping model. Since arbitrary corrections are always needed, we will base our method on the simple method: SSD.

2.3. Corrective enveloping

To overcome the limitations of SSD, an animator will typically find the bad-looking poses and apply *corrective enveloping* [RL99]. The goal of corrective enveloping is to modify the dress shape such that when SSD is applied, the

Figure 2: ***(a)*** *Enveloping weight initializations. We manually sketch out the kinematic influence regions (red in color plate) for each joint. From left to right: neck, clavicle, shoulder elevation/abduction, 50% shoulder twist, 100% shoulder twist, elbow flexion, 50% forearm twist, 100% forearm twist, wrist flexion, back rotation, waist rotation, base rotation, hip rotation, knee flexion, ankle rotation. The left-side regions are determined by symmetry.* ***(b)*** *Optimized enveloping weights. Here we show the enveloping weights for each joint on a scale from white (no influence) to red (full influence). The gray regions are outside the initialization area and therefore have zero weight.* ***(c)*** *Pose-dependent deformation regions. From left to right: neck rotation, clavicle rotation, shoulder rotation, elbow flexion, elbow twist, spine rotation, waist rotation, hip rotation, knee flexion.*

correct shape will result. Each joint j will have a certain number of "key angles" $\mathbf{r}_k$ where the dress shape has been edited. Then, for every vertex that is influenced by j, we store a vector offset $\mathbf{k}_{j,k}$. When posing the character, we adjust the raw dress shape $\bar{\mathbf{v}}'$ by adding a weighted combination of the offsets for that vertex. Thus:

$$\bar{\mathbf{v}}_i = \bar{\mathbf{v}}'_i + \sum_j \sum_k \omega_{j,k} \mathbf{k}_{j,k} \tag{2}$$

In the above equation, j is summed over each joint that influences vertex i, and k is summed over the number of keys for joint j.

Various alternatives have been suggested for how to determine the weights $\omega_{j,k}$. Sloan et al. [SRC01] calculate weights using radial basis functions (RBFs) on the example poses. To create the RBFs, we select a set of joint angles at which we will sample each joint. We chose to populate our set of sampling angles by looking at a corpus of posed skeletons (drawn from our scan database, as described in Section 4.1.1). We automatically choose zero rotation as one sample point, and then greedily add in samples from our pool of poses which are as far as possible from the other samples. We add samples until all observed poses are within 0.2 radians of some sample (or 0.3 radians for the shoulder joint, which would otherwise have too many samples).

Once we establish the key angles, we can state that the corrected dress shape for any pose is found by using RBFs to find the weights $\omega_{j,k}$ for each joint and key, and apply Equation 2. We then apply regular enveloping to the modified dress shape. We summarize this process as a function $f(\mathbf{c}, \mathbf{s}, \mathbf{q})$, where $\mathbf{c}$ includes the original dress shape, the skeleton parameters, and the deformation offsets.

When multiple joints affect the same part of the surface, corrective enveloping becomes difficult. Previous work has sidestepped this problem by combining multiple joints into one sample space, however this means that all combinations of joint values must be sampled. Since we will eventually be building a large model of identity variation, we prefer to create a compact pose model with as few samples as possible. Therefore, we consider overlapping influence regions to be separate, and attempt to learn the overlapping effects of each joint as if they were independent.

Another distinction of our approach is that we do not demand that the body shape is actually observed at the key angles, because it would be nearly impossible to force each of our subjects to strike a precise set of joint angles for scanning. Instead, we will take a data-fitting approach, where we attempt to find the offsets at the key angles that, when interpolated, would best explain the scanned poses that we do observe.

We summarize all of the information needed to put a particular individual into any pose using corrective enveloping into a single vector called the *character vector*, $\mathbf{c}$. It includes the dress shape $\bar{\mathbf{v}}$, skeleton parameters $\mathbf{b}$, and the pose-dependent deformation offsets $\mathbf{k}$. The key angles and enveloping weights we consider to be common to all people, and are not included in the character vector.

We defer discussion of how the character vector and skinning weights are learned until Section 4. The results of our corrective enveloping method are shown in Figure 3.

3. Matching

In order to relate the unstructured range scan meshes to our chosen mesh $\mathcal{M}$, we must first apply a surface matching technique. That is, for each scan α, we would like to summarize the observed shape as a collection of 3D vectors $\mathbf{e}_i^{(\alpha)}$, where i is a vertex index in our canonical surface $\mathcal{M}$.

The matching framework presented by Allen et al. [ACP03] is robust to missing surface data, and has been shown to work well for matching human body scans. We used this algorithm to match the scans that were in a standard standing pose, however this matching method is not suitable when the scan and the template are in extremely different poses (see Figure 4c). A key assumption in this algorithm is that the deformation in any local region is roughly constant. However, if there is a large pose change, such as a bent elbow, then this assumption is violated. Rather than

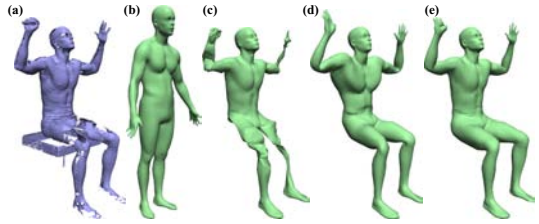

Figure 4: *Mesh matching to a drastically different pose.* ***(a)*** *Target mesh.* ***(b)*** *Standard template.* ***(c)*** *Match using standard initialization.* ***(d)*** *Skinned template.* ***(e)*** *Match using skinned initialization.*

changing the arm direction suddenly at the elbow, this approach prefers to gradually change the angle of the arm over its length.

Therefore, in order to match our template to scans in different poses, we must put our template into the appropriate pose using enveloping. We first determine the pose of the scan using the marker positions (see Section 4.1.1), and approximate enveloping weights (see Section 4.1.2). We then repose the template and apply the shape-matching algorithm as usual, giving the result shown in Figure 4e.

4. Learning

Now that we have a consistent mesh representation for all of the examples, we present a method for automatically learning the enveloping weights and pose deformations. We begin by establishing a probabilistic method for learning pose-dependent deformations of a single character, given an arbitrary set of example surfaces. It is critical to set up this single-character step in a way that will generalize to the multi-character problem in Section 4.2.

4.1. Learning a single character

Suppose we have n_α scans, which have been mapped to our standard surface representation using the algorithm from the previous section. We denote the i^{th} vertex of the α^{th} matched example by $\mathbf{e}_i^{(\alpha)}$.

Our goal in this section is to find the optimal character vector values, given our example data set. We also estimate the pose of each of the scans, $\mathbf{q}_\alpha$, and the optimal enveloping weights $\mathbf{s}$. Using the corrective enveloping method developed in Section 2.3, we can determine where in 3D space we would expect $\mathbf{e}_i^{(\alpha)}$ to appear for any particular value of $\mathbf{c}$, $\mathbf{q}_\alpha$, and $\mathbf{s}$. We call this reconstructed point $\mathbf{v}_i^{(\alpha)}$:

$$\mathbf{v}_i^{(\alpha)} = f(\mathbf{c},\mathbf{s},\mathbf{q}_\alpha)_i \tag{3}$$

Now we couch our problem in probabilistic terms. For any observed point $\mathbf{e}_i^{(\alpha)}$, we expect to find it nearby $\mathbf{v}_i^{(\alpha)}$, subject to some observation noise ν:

$$\mathbf{e}_i^{(\alpha)} = \mathbf{v}_i^{(\alpha)} + \nu \qquad \nu \sim N(\mathbf{0},\sigma_\mathbf{v}^2\mathbf{I}) \tag{4}$$

We assume that the observation noise is drawn from an isotropic Gaussian with variance $\sigma_\mathbf{v}^2$. Therefore, the probability of $\mathbf{e}_i^{(\alpha)}$, given a particular set of parameter values is:

$$p(\mathbf{e}_i^{(\alpha)}|\mathbf{c},\mathbf{s},\mathbf{q}_\alpha) = \frac{1}{(2\pi\sigma_\mathbf{v})^{3/2}} \exp\left(-\frac{1}{2\sigma_\mathbf{v}^2}||\mathbf{e}_i^{(\alpha)} - \mathbf{v}_i^{(\alpha)}||^2\right) \tag{5}$$

Armed with these probabilities, we can find the optimal parameters using maximum a posteriori (MAP) estimation. The probability of our parameter values given the data is:

$$P = p(\mathbf{c},\mathbf{s},\{\mathbf{q}_\alpha\}\,|\,\{\mathbf{e}_i^{(\alpha)}\}) \tag{6}$$

Here, the set on the right-hand side includes all of our point observations from all scans. Using Bayes' rule, we can re-express Equation 6 in terms of the posterior beliefs (Equation 5), multiplied by the prior probability of the parameters:

$$P = \left[\prod_{\alpha=1}^{n_\alpha}\left[\prod_{i=1}^{n_\mathbf{v}} p(\mathbf{e}_i^{(\alpha)}|\mathbf{c},\mathbf{s},\mathbf{q}_\alpha)\right]\right] p(\mathbf{c})\; p(\mathbf{s})\; p(\{\mathbf{q}_\alpha\}) \tag{7}$$

The prior probability terms $p(\mathbf{c})\; p(\mathbf{s})\; p(\{\mathbf{q}_\alpha\})$ reflect our assumptions about what parameter values are more likely, without considering the data. For example, we would expect the enveloping weights to vary smoothly across the surface; a set of enveloping weights that contains a sudden change in weight is improbable considering the fleshy nature of a human. (We will describe the form of these prior terms in the following subsections.)

To apply MAP estimation, we find the parameter values that minimize the negative log likelihood. We ignore the terms that do not depend on the parameter values (e.g., the Gaussian normalization constants), and split $p(\mathbf{c})$ into the product of the parameters that make up the character vector:

$$\begin{aligned} -\log P \;=\; & n_\alpha n_\mathbf{v} 1.5\log(2\pi\sigma_\mathbf{v}^2) + \sum_{\alpha=1}^{n_\alpha}\sum_{i=1}^{n_\mathbf{v}} \frac{1}{2\sigma_\mathbf{v}^2}||\mathbf{e}_i^{(\alpha)} - \mathbf{v}_i^{(\alpha)}||^2 - \\ & \log p(\bar{\mathbf{v}}) - \log p(\mathbf{s}) - \log p(\mathbf{b},\{\mathbf{q}_\alpha\}) - \log p(\mathbf{k}) \end{aligned} \tag{8}$$

Due to the non-linearities in the skeletal transformations, Equation 8 is too complicated to solve analytically. Therefore, we use a standard optimization package [ZBLN97]. Because there are thousands of variables to optimize and many local minima, it is critical to find a good initialization for the parameter values.

In the following subsections, we will address each parameter value individually, describing the initialization process and also the prior used for each parameter. We will address these parameters in the order in which they must be initialized: first the bones $\mathbf{b}$ and poses $\mathbf{q}_\alpha$ in Section 4.1.1, then the enveloping weights $\mathbf{s}$ in Section 4.1.2, then the dress vertices $\bar{\mathbf{v}}$ in Section 4.1.3, and finally the pose-dependent deformations $\mathbf{k}$ in Section 4.1.4. Afterwards, we will address issues of symmetry (Section 4.1.5), and tuning the sigma values (Section 4.1.6).

4.1.1. Initializing and prior on the skeleton parameters

Here we consider how to initialize the bones $\mathbf{b}$ and the poses $\mathbf{q}_\alpha$ based on our examples. We start with the labelled markers from each scan. This set of markers is similar to motion capture data, and can be optimized using inverse kinematics (IK) while also optimizing the bone DOFs. (One such optimization technique is discussed in detail by Silaghi et al. [SPB*98].)

Unlike motion capture, we have a relatively small set of poses, sometimes as few as two, which is clearly insufficient to determine the skeletal parameters. Indeed, we found that only for our subject who was scanned in 69 poses could we reliably determine a skeleton using markers alone. Therefore, we supplement IK with a heuristic technique that is inspired by how biomechanicists estimate of joint centers from surface landmarks. Using a pose example which is in the standard CAESAR standing pose, we can estimate the joint centers as a linear function of landmark positions on the body (e.g., the knee joint center is approximated by the midpoint between two landmarks on either side of the leg). We used the 69-pose example to find good surface landmark positions for estimating the joint centers, and then find the corresponding landmark points on other individuals using surface matching (Section 3).

We employ this heuristic in the form of a prior term, $p(\mathbf{b}, \{\mathbf{q}_\alpha\}))$, to be included both during the IK initialization, and during the optimization of Equation 8.

This prior states that for the CAESAR standing poses in which we applied our heuristic techniques, the joint centers calculated from our skeleton hierarchy should be close to the heuristic-estimated joint centers, which we call $\mathbf{h}_j^{(\alpha)}$. By "close to," we mean the distance has a Gaussian distribution with mean 0 and variance $\sigma_{\mathbf{b}}^2$:

$$-\log p(\mathbf{b}, \{\mathbf{q}_\alpha\}) \approx \sum_j \frac{1}{2\sigma_{\mathbf{b}}^2} ||\mathbf{h}_j^{(\alpha)} - \mathbf{M}_{\mathbf{q}_\alpha, \mathbf{b}, j} [0\ 0\ 0\ 1]^{\mathrm{T}}||^2 \tag{9}$$

In the above equation, α is the index of the CAESAR standing pose, and j is summed over the joint centers found by our heuristic method. By using this prior, we can avoid the noise and local minima that IK would provide when few poses are available.

4.1.2. Initializing and prior on the enveloping weights

Given a large enough sample of poses, we could, in principle, learn all of the enveloping weights automatically, by determining which joint angles affect which surface points. However, even with a large number of examples, one could imagine that there could be some accidental correlation between distant body parts that would introduce spurious weights. Therefore, we manually label the approximate influence regions of each joint, in a one-time process, as shown in Figure 2a. The labellings identify the maximum extent of each joint's influence. Outside each joint's influence region, its corresponding enveloping weight must be zero.

To obtain a reasonable enveloping result, we need to create a smooth transition between the influence regions. We do so by introducing a prior on the enveloping weights, based on the squared umbrella operator $U^2(\mathbf{s}_i)$ [KCVS98]. To minimize the curvature of our skinning weight function (in the mesh domain), we introduce a zero-centered Gaussian term for each weight-curvature estimate with variance $\sigma_{\mathbf{s}}^2$:

$$-\log p(\mathbf{s}) \approx \sum_{i=1}^{n_{\mathbf{v}}} \frac{1}{2\sigma_{\mathbf{s}}^2} ||U^2(\mathbf{s}_i)||^2 \tag{10}$$

Our initial values for the enveloping weights are found by starting with the weights in Figure 2a, normalizing them, and then minimizing this prior term only (ignoring the actual data). The resulting enveloping weights are very similar to those shown in Figure 2b.

4.1.3. Initializing and prior on the dress shape

We can trivially initialize the dress shape by using one of our matching results (e.g., for the CAESAR standing pose), and determine the dress location of each vertex using the initial skeletons and enveloping weights.

We choose to use a uniform prior for the dress shape (i.e., all shapes are equally likely in the absense of data), as our initialization brings us quite close to the correct value, and so there was no need for additional regularization. Therefore, $\log p(\overline{\mathbf{v}})$ is a constant and can be dropped from Equation 8.

4.1.4. Initialization and prior on the pose-dependent deformations

We initialize the pose-dependent deformation offsets to be zero, which is equivalent to using SSD without corrective enveloping.

Unlike the dress vertices, we will introduce a prior on the pose-dependent deformation offsets. As mentioned in the previous subsection, we could reliably obtain a reasonable dress shape from the standing CAESAR pose. This is because we chose to trust all of the data in this pose, even where the scanned shape had holes, because our matching algorithm works quite well in this pose. However, in other poses, we are using a skinned template which has all of the bad artifacts such as volume loss and a rubbery appearance. Moreover, the other poses tend to have more occlusions or grazing angle views, resulting in very large holes. Since we do not have good data in these regions, we do not include those example points $\mathbf{e}_i^{(\alpha)}$ in Equation 8. In fact, we scale the weight of each observation in accordance with the scanner confidence value, so that less-certain observations contribute less to our model.

This solution is intuitively reasonable: we want to fit more closely to good data than bad. However, it causes a problem near the boundaries of good and missing data. Suppose we notice that a bicep bulges, but there is a small hole in the flexed arm scan. Our system would assume that the bicep does not bulge in the hole, since there is no data to indicate

any change. This runs counter to our intuition that the deformations are locally consistent, that is, shape changes at nearby points should be very similar.

To include this intuition in our model, we supply the following prior on the pose-dependent deformation offsets, which applies to all neighboring vertices in the mesh:

$$-\log p(\mathbf{k}) \approx \sum_{J}^{n_\mathbf{j}} \sum_{\{i_1,i_2|\{\mathbf{v}_{i_1},\mathbf{v}_{i_2}\}\in edges(\mathcal{M})\}} \frac{1}{2\sigma_\mathbf{k}^2} ||\mathbf{k}_{i_1,j} - \mathbf{k}_{i_2,j}||^2 \tag{11}$$

This prior has an additional benefit. We specify an influence region for each pose-dependent deformation offset, as shown in Figure 2c. We then force all offsets outside the influence region to be zero. Our regularization term will then cause a smooth fall-off at the boundary of the influence region. Without this regularization, we would observe seams at the boundaries, where spurious pose-dependent deformations developed near the boundary.

4.1.5. Symmetry

By and large, humans are bilaterally symmetrical across the sagittal plane. We exploit this fact in our learning step in order to reduce the number of variables in our model by approximately half, by implicitly stating that the left-side positions and displacements are the mirror-image of the right-side values. We also implicitly make the left and right bone DOFs the same. We make a concession to asymmetry when it comes to carrying angles, and allow those to be unequal. The reason is based on the quite high variation of these angles, and the high mismatch that would arise from not respecting them.

4.1.6. Estimating variances

So far, we have introduced many variance values which we assume have been provided manually: $\sigma_\mathbf{v}^2$, $\sigma_\mathbf{b}^2$, $\sigma_\mathbf{s}^2$, and $\sigma_\mathbf{k}^2$. We made initial estimate for these values of $(1\text{ mm})^2$, $(1\text{ mm})^2$, $(0.01)^2$, and $(1\text{ cm})^2$ respectively. However, we do not want to have to tweak all of these parameters to get the ideal values.

Instead, after running our optimization for several iterations, we re-estimate these parameters by optimizing Equation 8 in closed form for the best sigma values, and then alternate back to the main optimization. (This technique has been applied to a similar learning problem by Torresani and Hertzmann [TH04].) Each of the aforementioned sigma values are based on a collection of Gaussian distributions. If the number of Gaussians involved is n, the dimension of each vector is d, and the distance from each point to the Gaussian center is e_i, then the optimal sigma value is:

$$\sigma^2 = (\sum_{i}^{n} ||e_i||^2)/nd \tag{12}$$

4.2. Learning all identities

Next, we consider the problem of learning variation in both pose and body shape. Thanks to the formulation of the character vector $\mathbf{c}$ we now have a convenient way to represent this variation. Recall that the character vector encapsulates all of the information needed to reconstruct a particular identity in any pose, implicitly storing such parameters as the individual's height, girth, and muscle tone. Previous work, such as Allen et al. [ACP03] and Seo et al. [SCMT03], has characterized the space of all human body shapes as a distribution within all shape-vectors. The key idea here is that instead of finding a distribution over shape vectors, we will find a distribution over character vectors.

In principle, if we had a large number of character vectors, for example, if we had captured hundreds of individuals each in many poses and applied the technique of Section 4.1, then we could run PCA on those character vectors and have our model. However, this approach would require a very large number of scans, which would be very expensive to acquire, store, and process. It is much more appealing if we can just use whatever data samples we are given to build our model. For instance, if we have a lot of pose data for one body shape, we should be able to estimate the pose variation for another, similar body shape, even if we just have one or two scans of that person.

4.2.1. Learning character vectors

To model all character distributions, we assume that all character vectors are drawn from a latent variable distribution of the form: $\mathbf{c}^{(\beta)} = \mathbf{W}\mathbf{x} + \bar{\mathbf{c}}$, where $\mathbf{x}$ has a Gaussian distribution with unit covariance. We want to solve for the components $\mathbf{W}$ and the mean character vector $\bar{\mathbf{c}}$. Unlike PCA, we will not require that the components are orthonormal. If we had a large set of character vectors, then we could use conventional PCA. One problem with this approach is that different parts of the character vector have different scales (depending on whether they are vertex positions, angles, or offsets), and so our analysis will be biased. A more serious issue is that we cannot actually observe the character vectors directly; we can only observe the shapes that they produce in a particular pose. In fact, if we only see an individual in a couple of poses, we may not have enough information to reliably know any elements of $\mathbf{c}$.

Therefore, we propose the following generative model for our vertex observations, based on our corrective enveloping function f:

$$\mathbf{e}_i^{(\alpha,\beta)} = f(\mathbf{W}\mathbf{x}_\beta + \bar{\mathbf{c}}, \mathbf{s}, \mathbf{q}_\alpha)_i + \nu; \quad \mathbf{x}_\beta \sim N(\mathbf{0},\mathbf{I}); \quad \nu \sim N(\mathbf{0},\sigma_\mathbf{v}^2\mathbf{I}) \tag{13}$$

We model the observation noise as an isotropic Gaussian variable ν. Notice that Equation 13 is exactly the same as Equation 4 in the previous subsection, except that we have replaced $\mathbf{c}$ with $\mathbf{W}\mathbf{x}_\beta + \bar{\mathbf{c}}$. That is, we now use a character vector that we have reconstructed from the components instead of using a fixed character vector.

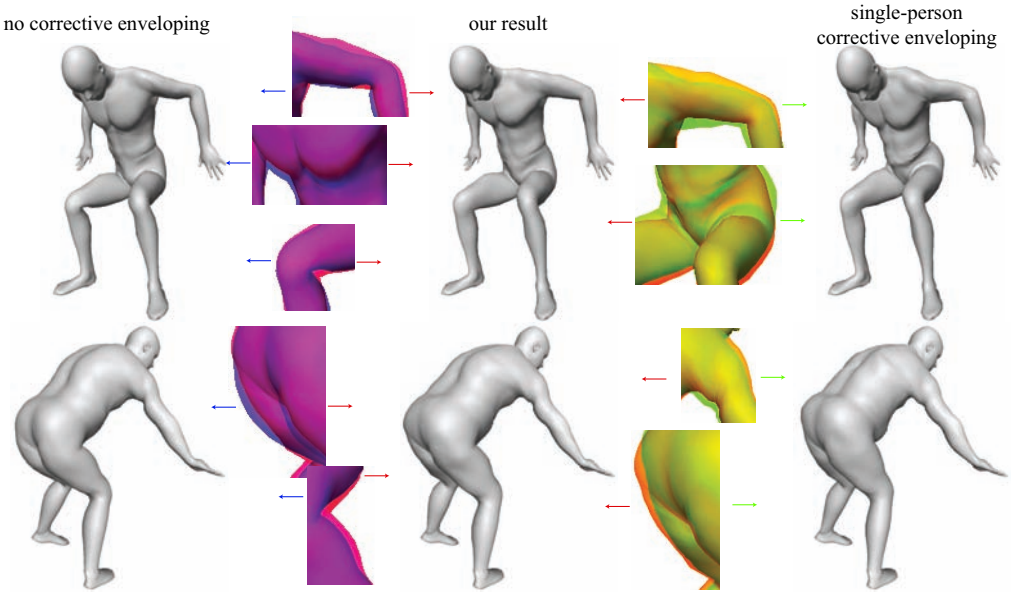

Figure 5: *Results of meshes generated with our hybrid model are in the middle column. Each of these individuals was only observed in a standard standing and seated pose, and then put into a novel pose using our method. We compare with using enveloping alone, in the left column. In the right column, we show the result of transferring just one typical person's corrective enveloping onto the new character. Our learned model is able to generate corrections that are more suitable to the new individuals' body type. The color-tinted details (see color plate) compare our result with the left and right alternatives.*

Our generative model is based on the Probabilistic Principal Component Analysis (PPCA) algorithm introduced by Tipping and Bishop [TB99] and by Roweis [Row98]. To find the best $\mathbf{W}$ and $\bar{\mathbf{c}}$ values that explain our data, we could apply the Expectation-Maximization (EM) algorithm [DLR77], which alternates between estimating a distribution for each $\mathbf{x}_\beta$, and then finding the maximum expected likelihood values of $\mathbf{W}$ and $\bar{\mathbf{c}}$. However, unlike PPCA, we are observing our data through the lens of corrective enveloping, a complex and non-linear process. Therefore, the estimated distributions for $\mathbf{x}_\beta$ will not be Gaussian, making a full EM optimization very difficult. Instead, we alternate between optimizing for a fixed value of each $\mathbf{x}_\beta$, and then optimizing $\mathbf{W}$ and $\bar{\mathbf{c}}$ using the MAP approach introduced in Section 4.1. Neal and Hinton [NH98] refer to this approximation as a "winner-take-all" variant of EM, and suggest that although the guarantees of convergence that EM endows may no longer apply, this approach will make progress towards the minimum.

4.2.2. Initializing latent variables

In PPCA, the latent variables $\mathbf{x}_\beta$ can be initialized randomly, due to the convergence guarantees. However, since our optimization is less robust than the full EM approach, we would do well to use a good initialization. The bone parameters are a good choice to guide this initialization, because the skeletons are very important to obtaining an accurate fit, and they can be estimated without running our full optimization. We find the bone parameters $\mathbf{b}$ for each individual in our dataset (Section 4.1.1), and run conventional PCA on these parameters. We then use the reconstruction weights provided by PCA to initialize each $\mathbf{x}_\beta$.

Variable	#
Dress shape $\bar{\mathbf{v}}$	106,920
Bone DOFs $\mathbf{b}$	210
Pose-dependent deformations $\mathbf{k}$	314,220
Pose DOFs $\mathbf{q}$	12,561
Skinning weights $\mathbf{s}$	17,820
Reconstruction weights $\mathbf{x}$	450

Table 1: *Summary of the total number of variables in our optimization for all scans and components.*

4.2.3. Summary of optimization steps

To model all poses for all identities, we need to solve for a lot of variables (see Table 1), and take care to optimize them in the correct order. In this section we will describe our procedure for learning all variables from the range scans.

We begin with the 69-pose data set, and estimate $\mathbf{b}$ and $\mathbf{q}$ from the markers (§4.1.1). We then initialize the skinning weights smoothly (§4.1.2), and match all of the surfaces using the skinning initialization (§3). We then further optimize $\mathbf{q}$ and $\bar{\mathbf{v}}$, then add $\mathbf{s}$ and $\mathbf{b}$, and finally include $\mathbf{k}$. When the optimization starts to converge, we update the

variances (§4.1.6) and then optimize further. This gives us a single character vector to start with.

To move on to the multi-identity problem, we estimate skeletons for the full dataset (§4.1.1), and use our learned skinning model to initialize the surface matching (§3). We also run PCA on the bone DOFs to initialize $\mathbf{x}$ (§4.2.2). We first optimize for $\mathbf{q}$ and $\mathbf{b}$ using only the skeleton prior, and then initialize the rest of $\bar{\mathbf{c}}$ with the single-person variables. Now we can optimize all variables as in the single-person case, but we also alternate with optimizing for $\mathbf{x}$. The end-to-end learning process takes about a day to run.

We chose to use 9 components (plus the mean $\bar{\mathbf{c}}$) to represent body shape. However, we only have six pose-dense individuals. Therefore, it is unreasonable to expect to be able to learn 9 components for $\mathbf{k}$. Indeed, even with six components, we found that there was serious overfitting. Therefore, we reduced the number of components for $\mathbf{k}$ to just 3 (i.e., the other 6 character vector components will have $\mathbf{k}$ that are forced to be zero). By doing so, we eliminate overfitting and force our optimization to find a correlation between $\mathbf{k}$ and body shape.

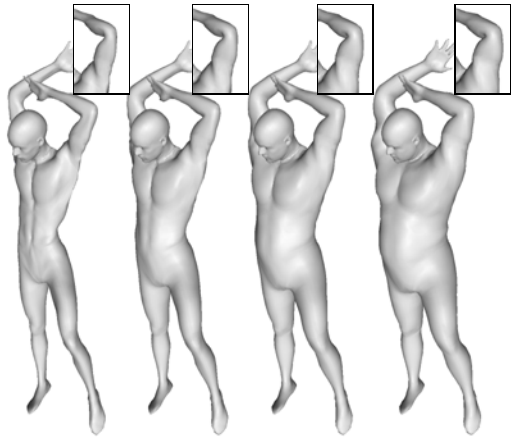

Figure 6: *Editing body weight. We edit the weight of one of the subjects (second from the left) using trends from the population. The height is kept constant. The insets show the re-posed left arm without corrective enveloping for comparison.*

5. Results

The overall root-mean-squared (RMS) reconstruction error of our learned model with regard to the training set is 4.9 mm on each vertex. We also fit our model to five additional scans of subjects who were not part of the training set, and obtained an RMS error of 8.1 mm. Some of this error is due to the difficulty in determining the pose and PCA weights of these novel characters (which is done through an optimization process).

Figure 3 shows our learned corrective enveloping model applied to two of the characters in our multi-pose training set. The novel poses were drawn from a motion capture sequence. Notice that the "joint-collapse" artifacts of pure enveloping are compensated for, and anatomical effects such as the pointiness of the elbow, and the shape change of the larger man's upper arm are accounted for.

In Figure 5, we demonstrate additional results where our model is applied to characters in the CAESAR set (who were only observed in two poses), and new poses are applied. Our results are much better that skinning applied alone: notice the rubbery look in the arms and legs, and the lack of muscle bulging in the triceps and pectoral muscles. In addition, we claim that a single pose-deformation model is not sufficient. To prove this claim, we also compare our result with using corrective enveloping learned from just a single, average individual. Our model is able to automatically generate corrective enveloping that is particular to a body type.

Using our latent variable model, we can perform analysis tasks similar to previous work [BV99, ACP03]. For example, we can learn a trend between recorded attributes about each example and the latent variables. Figure 6 demonstrates a learned trend between height, weight, and body shape. We are able to edit the weight of one of the subjects, while controlling his height independently.

6. Conclusion

We have presented an algorithm for learning character models from observations of human body shape. Our algorithm is robust in the face of sparse, irregular, and incomplete data. By incorporating other information beyond the base shape in our latent variable model, we have created a fundamentally more expressive model for modeling the interdependence of pose-dependent deformation and individual variation. We have shown how our model is useful for synthesizing and editing animated characters; in the future, we envision other applications, e.g., providing a shape prior for computer-vision and recognition applications.

A primary advantage of our approach is speed. Our synthesized models can be posed by evaluating a few RBF values, then taking a linear interpolation of keys, and then apply standard enveloping. Our unoptimized software implementation can generate a posed shape in just 13 ms on a 2.8 GHz PC. In contrast, methods based on deformation transfer [ASK*05] take around one second per body.

Our method could be further improved by including more data. Currently, our 16-pose datasets do not sample some regions of pose-space very well (e.g., there are very few raised arms and bent elbows). This causes some problems in poorly sampled parts of pose- and identity-space; notice, for instance, shoulder inflation for certain characters in the final part of the accompanying video. In addition, it would be nice to sample non-pose related DOFs, such as breathing

and muscle load. Of course with a suitable female dataset, we could also build an animatable female model.

One limitation of our approach is that some poses are very difficult to capture, because of occlusions, or because they are difficult to hold for a scanner. Self-collisions in the body are particularly troublesome; not only can they not be captured, but they are very hard to model using a smooth function such as RBFs, because self-collisions cause a sharp discontinuity in the shape. To address such issues, it may be necessary to eventually couple our method with a simulation framework. Such a framework would also be able to model dynamic phenomena, such as jiggling flesh, that we are unable to capture.

References

[ACP02] ALLEN B., CURLESS B., POPOVIĆ Z.: Articulated body deformation from range scan data. *ACM Transactions on Graphics (ACM SIGGRAPH 2002) 21*, 3 (2002), 612–619.

[ACP03] ALLEN B., CURLESS B., POPOVIĆ Z.: The space of human body shapes: reconstruction and parameterization from range scans. *ACM Transactions on Graphics (ACM SIGGRAPH 2003) 22*, 3 (2003), 587–594.

[ASK*05] ANGUELOV D., SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: SCAPE: shape completion and animation of people. *ACM Transactions on Graphics (ACM SIGGRAPH 2005) 24*, 3 (2005), 408–416.

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3D faces. In *Proceedings of ACM SIGGRAPH 99* (New York, 1999), Rockwood A., (Ed.), Computer Graphics Proceedings, Annual Conference Series, ACM Press/Addison-Wesley Publishing Co., pp. 187–194.

[DLR77] DEMPSTER A. P., LAIRD N. M., RUBIN D. B.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society (Series B) 39*, 1 (1977), 1–38.

[JT05] JAMES D. L., TWIGG C. D.: Skinning mesh animations. *ACM Transactions on Graphics (SIGGRAPH 2005) 24*, 3 (Aug. 2005).

[KCVS98] KOBBELT L., CAMPAGNA S., VORSATZ J., SEIDEL H.-P.: Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of ACM SIGGRAPH 98* (New York, 1998), ACM Press, pp. 105–114.

[KJP02] KRY P. G., JAMES D. L., PAI D. K.: Eigenskin: real time large deformation character skinning in hardware. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2002), ACM Press, pp. 153–159.

[Kv05] KAVAN L., ŽÁRA J.: Spherical blend skinning: a real-time deformation of articulated models. In *SI3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2005), ACM Press, pp. 9–16.

[LCF00] LEWIS J. P., CORDNER M., FONG N.: Pose space deformations: A unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of ACM SIGGRAPH 2000* (2000), Akeley K., (Ed.), Computer Graphics Proceedings, Annual Conference Series, ACM Press / ACM SIGGRAPH / Addison Wesley Longman, pp. 165–172.

[MG03] MOHR A., GLEICHER M.: Building efficient, accurate character skins from examples. *ACM Transactions on Graphics (ACM SIGGRAPH 2003)* (2003).

[MTT91] MAGNENAT-THALMANN N., THALMANN D.: Human body deformations using joint-dependent local operators and finite-element theory. 243–262.

[NH98] NEAL R. M., HINTON G. E.: A new view of the EM algorithm that justifies incremental, sparse and other variants. In *Learning in Graphical Models*, Jordan M. I., (Ed.). Kluwer Academic Publishers, 1998, pp. 355–368.

[RL99] ROUET C., LEWIS J.: Method and apparatus for creating lifelike digital representations of computer animated objects by providing corrective enveloping. US Patent 5,883,638, 1999.

[Row98] ROWEIS S.: EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems* (1998), Jordan M. I., Kearns M. J., Solla S. A., (Eds.), vol. 10, The MIT Press.

[SCMT03] SEO H., CORDIER F., MAGNENAT-THALMANN N.: Synthesizing body models with parameterized shape modifications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), pp. 120–125.

[SK00] SINGH K., KOKKEVIS E.: Skinning characters using surface-oriented free-form deformations. In *Graphics Interface* (2000), pp. 35–42.

[SPB*98] SILAGHI M.-C., PLÄNKERS R., BOULIC R., FUA P., THALMANN D.: Local and global skeleton fitting techniques for optical motion capture. In *Proceedings of the International Workshop on Modelling and Motion Capture Techniques for Virtual Environments (CAPTECH-98)* (Berlin, Nov. 26–27 1998), Magnenat-Thalmann N., Thalmann D., (Eds.), vol. 1537 of *LNAI*, Springer, pp. 26–40.

[SPCM97] SCHEEPERS F., PARENT R. E., CARLSON W. E., MAY S. F.: Anatomy-based modeling of the human musculature. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1997), ACM Press/Addison-Wesley Publishing Co., pp. 163–172.

[SRC01] SLOAN P.-P., ROSE C., COHEN M. F.: Shape by example. In *Proceedings of 2001 Symposium on Interactive 3D Graphics* (2001), pp. 135–143.

[TB99] TIPPING M. E., BISHOP C. M.: Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B 61*, 3 (1999), 611–622.

[TH04] TORRESANI L., HERTZMANN A.: Automatic non-rigid 3D modeling from video. In *ECCV (2)* (2004), pp. 299–312.

[VBPP05] VLASIC D., BRAND M., PFISTER H., POPOVIĆ J.: Face transfer with multilinear models. *ACM Transactions on Graphics (ACM SIGGRAPH 2005) 24*, 3 (2005), 426–433.

[WP02] WANG X. C., PHILLIPS C.: Multi-weight enveloping: Least-squares approximation techniques for skin animation. In *Proceedings of the 2002 ACM SIGGRAPH Symposium on Computer Animation* (2002), pp. 129–138.

[ZBLN97] ZHU C., BYRD R. H., LU P., NOCEDAL J.: Algorithm 778. L-BFGS-B: Fortran subroutines for Large-Scale bound constrained optimization. *ACM Transactions on Mathematical Software 23*, 4 (Dec. 1997), 550–560.