SCAPE: Shape Completion and Animation of People

Dragomir Anguelov* Praveen Srinivasan* Daphne Koller* Sebastian Thrun* Jim Rodgers*
Stanford University

James Davis†
University of California, Santa Cruz

Figure 1: Animation of a motion capture sequence taken for a subject, of whom we have a single body scan. The muscle deformations are synthesized automatically from the space of pose and body shape deformations.

Abstract

We introduce the SCAPE method (Shape Completion and Animation for PEople) — a data-driven method for building a human shape model that spans variation in both subject shape and pose. The method is based on a representation that incorporates both articulated and non-rigid deformations. We learn a *pose deformation model* that derives the non-rigid surface deformation as a function of the pose of the articulated skeleton. We also learn a separate model of variation based on body shape. Our two models can be combined to produce 3D surface models with realistic muscle deformation for different people in different poses, when neither appear in the training set. We show how the model can be used for *shape completion* — generating a complete surface mesh given a limited set of markers specifying the target shape. We present applications of shape completion to partial view completion and motion capture animation. In particular, our method is capable of constructing a high-quality animated surface model of a moving person, with realistic muscle deformation, using just a single static scan and a marker motion capture sequence of the person.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Hierarchy and geometric transformations; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: synthetic actors, deformations, animation, morphing

1 Introduction

Graphics applications often require a complete surface model for rendering and animation. Obtaining a complete model of a particular person is often difficult or impossible. Even when the person can be constrained to remain motionless inside of a Cyberware full body scanner, incomplete surface data is obtained due to occlusions. When the task is to obtain a 3D sequence of the person in motion, the situation can be even more difficult. Existing marker-based motion capture systems usually provide only sparse measurements at a small number of points on the surface. The desire is to map such sparse data into a fully animated 3D surface model.

This paper introduces the SCAPE method (Shape Completion and Animation for PEople) — a data-driven method for building a unified model of human shape. Our method learns separate models of body deformation — one accounting for changes in pose and one accounting for differences in body shape between humans. The models provide a level of detail sufficient to produce dense full-body meshes, and capture details such as muscle deformations of the body in different poses. Importantly, our representation of deformation allows the pose and the body shape deformation spaces to be combined in a manner which allows proper deformation scaling. For example, our model can correctly transfer the deformations of a large person onto a small person and vice versa.

The pose deformation component of our model is acquired from a set of dense 3D scans of a single person in multiple poses. A key aspect of our pose model is that it decouples deformation into a rigid and a non-rigid component. The rigid component of deformation is described in terms of a low degree-of-freedom rigid body skeleton. The non-rigid component captures the remaining deformation such as flexing of the muscles. In our model, the deformation for a body part is dependent only on the adjacent joints. Therefore, it is relatively low dimensional, allowing the shape deformation to be learned automatically, from limited training data.

Our representation also models shape variation that occurs across different individuals. This model component can be acquired from a set of 3D scans of different people in different poses. The shape variation is represented by using principal component analysis (PCA), which induces a low-dimensional subspace of body shape deformations. Importantly, the model of shape variation does not get confounded by deformations due to pose, as those are accounted for separately. The two parts of the model form a single unified framework for shape variability of people. The framework can be used to generate a complete surface mesh given only a succinct specification of the desired shape — the angles of the human skeleton and the eigen-coefficients describing the body shape.

We apply our model to two important graphics tasks. The first is partial view completion. Most scanned surface models of humans

*e-mail: {drago,praveens,koller,thrun,jimkr}@cs.stanford.edu
†e-mail: davis@cs.ucsc.edu

have significant missing regions. Given a partial mesh of a person for whom we have no previous data, our method finds the shape that best fits the observed partial data in the space of human shapes. The model can then be used to predict a full 3D mesh. Importantly, because our model also accounts for non-rigid pose variability, muscle deformations associated with the particular pose are predicted well even for unobserved parts of the body.

The second task is producing a full 3D animation of a moving person from marker motion capture data. We approach this problem as a shape completion task. The input to our algorithm is a single scan of the person and a time series of extremely sparse data — the locations of a limited set of markers (usually between 50 and 60) placed on the body. For each frame in the sequence, we predict the full 3D shape of the person, in a pose consistent with the observed marker positions. Applying this technique to sequences of motion capture data produces full-body human 3D animations. We show that our method is capable of constructing high-quality animations, with realistic muscle deformation, for people of whom we have a single range scan.

In both of these tasks, our method allows for variation of the individual body shape. For example, it allows for the synthesis of a person with a different body shape, not present in the original set of scans. The motion for this new character can also be synthesized, either based on a motion capture trajectory for a real person (of similar size), or keyframed by an animator. Thus, our approach makes it possible to create realistic shape completions and dense 3D animations for people whose exact body shape is not included in any of the available data sources.

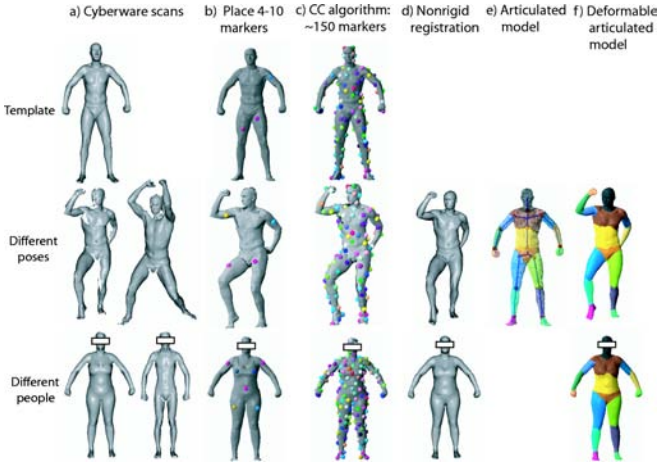

Figure 2: The mesh processing pipeline used to generate our training set. (a) We acquired two data sets spanning the shape variability due to different human poses and different physiques. (b) We select a few markers by hand mapping the template mesh and each of the range scans. (c) We apply the Correlated Correspondence algorithm, which computes numerous additional markers. (d) We use the markers as input to a non-rigid registration algorithm, producing fully registered meshes. (e) We apply a skeleton reconstruction algorithm to recover an articulated skeleton from the registered meshes. (f) We learn the space of deformations due to pose and physique.

2 Related Work

The recent example-based approaches for learning deformable human models represent deformation by point displacements of the example surfaces, relative to a generic template shape. For modeling pose deformation, the template shape is usually assumed to be an articulated model. A popular animation approach called *skinning* (described in [Lewis et al. 2000]) assumes that the point displacements are generated by a weighted set of (usually linear) influences from neighboring joints. A more sophisticated method was presented by Allen *et al.* [2002], who register an articulated model (represented as a posable subdivision template) to scans of a human in different poses. The displacements for a new pose are predicted by interpolating from a set of example scans with similar joint angles. A variety of related methods [Lewis et al. 2000; Sloan et al. 2001; Wang and Phillips 2002; Mohr and Gleicher 2003] differ only in the details of representing the point displacements, and in the particular interpolation method used. Models of pose deformation are learned not only from 3D scans, but also by combining shape-from-silhouette and marker motion capture sequences [Sand et al. 2003]. However, none of the above approaches learn a model of the shape changes between different individuals.

To model body shape variation across different people, Allen *et al.* [2003] morph a generic template shape into 250 scans of different humans in the same pose. The variability of human shape is captured by performing principal component analysis (PCA) over the displacements of the template points. The model is used for hole-filling of scans and fitting a set of sparse markers for people captured in the standard pose. Another approach, by Seo and Thalmann [2003], decomposes the body shape deformation into a rigid and a non-rigid component, of which the latter is also represented as a PCA over point displacements. Neither approach learns a model of pose deformation. However, they demonstrate preliminary animation results by using expert-designed skinning models. Animation is done by bringing the space of body shapes and the skinning model into correspondence (this can be done in a manual or semi-automatic way [Hilton et al. 2002]), and adding the point displacements accounting for pose deformation to the human shape. Such skinning models are part of standard animation packages, but since they are usually not learned from scan data, they usually don't model muscle deformation accurately.

An obvious approach for building a data-driven model of pose and body shape deformation would be to integrate two existing methods in a similar way. The main challenge lies in finding a good way to combine two distinct deformation models based on point displacements. Point displacements cannot be multiplied in a meaningful way; adding them ignores an important notion of scale. For example, pose displacements learned on a large individual cannot be added to the shape of a small individual without undesirable artifacts. This problem has long been known in the fields of deformation transfer and expression cloning [Noh and Neumann 2001]. In order to address it, we take an inspiration in the deformation transfer method of Sumner and Popović [2004]. It shows how to retarget the deformation of one mesh to another, assuming point-to-point correspondences between them are available. The transfer maintains proper scaling of deformation, by representing the deformation of each polygon using a 3×3 matrix. It suggests a way of mapping pose deformations onto a variety of human physiques. However, it does not address the task of representing and learning a deformable human model, which is tackled in this paper.

Multilinear models, which are closely related to our work, have been applied for modeling face variation in images [Vasilescu and Terzopoulos 2002]. A generative model of human faces has to address multiple factors of image creation such as illumination, expression and viewpoint. The face is modeled as a product of linear appearance models, corresponding to influences of the various factors. Ongoing work is applying multilinear approaches to model 3D face deformation [Vlasic et al. 2004]. Our method adapts the idea to the space of human body shapes, which exhibits articulated structure that makes human body modeling different from face modeling. In particular, we directly relate surface deformations to the underlying body skeleton. Such a model would not be sufficient to address face deformation, because a significant part of the deformation is purely muscle-based, and is not correlated with the skeleton.

Our shape-completion application is related to work in the area of hole-filling. Surfaces acquired with scanners are typically incomplete and contain holes. A common way to complete these holes is to fill them with a smooth surface patch that meets the boundary conditions of the hole [Curless and Levoy 1996; Davis et al. 2002; Liepa 2003]. These approaches work well when the holes are small compared to the geometric variation of the surface. Our application,

by contrast, requires the filling of huge holes (e.g., in some experiments more than half of the surface was not observed; in others we are only provided with sparse motion capture data) and we address it with a model-based method. Other model-based solutions for hole filling were proposed in the past. Kähler *et al.* [2002] and Szeliski and Lavallée [1996] use volumetric template-based methods for this problem. These approaches work well for largely convex objects, such as a human head, but are not easily applied to objects with branching parts, such as the human body. While the work of Allen *et al.* [2003] can be used for hole-filling of human bodies, it can only do so if the humans are captured in a particular pose.

Marker motion capture systems are widely available, and can be used for obtaining high-quality 3D models of a moving person. Existing animation methods (e.g. [Allen et al. 2002; Seo and Magnenat-Thalmann 2003]) do not utilize the marker data and assume the system directly outputs the appropriate skeleton angles. They also do not handle body shape variation well, as previously discussed. Both of these limitations are lifted in our work.

3 Acquiring and Processing Data Meshes

The SCAPE model acquisition is data driven, and all the information about the shape is derived from a set of range scans. This section describes the basic pipeline for data acquisition and pre-processing of the data meshes. This pipeline, displayed in in Fig. 2, consists largely of a combination of previously published methods. The specific design of the pipeline is inessential for the main contribution of this paper; we describe it first to introduce the type of data used for learning our model.

Range Scanning We acquired our surface data using a Cyberware WBX whole-body scanner. The scanner captures range scans from four directions simultaneously and the models contain about 200K points. We used this scanner to construct full-body instance meshes by merging the four scan views [Curless and Levoy 1996] and subsampling the instances to about 50,000 triangles [Garland and Heckbert 1997].

Using the process above, we obtained two data sets: a *pose data set*, which contains scans of 70 poses of a particular person in a wide variety of poses, and a *body shape data set*, which contains scans of 37 different people in a similar (but not identical) pose. We also added eight publicly available models from the CAESAR data set [Allen et al. 2003] to our data set of individuals.

We selected one of the meshes in the pose data set to be the *template mesh*; all other meshes will be called *instance meshes*. The function of the template mesh is to serve as a point of reference for all other scans. The template mesh is hole-filled using an algorithm by Davis *et al.*[2002]. In acquiring the template mesh, we ensured that only minor holes remained mostly between the legs and the armpits. The template mesh and some sample instance meshes are displayed in Fig. 2(a). Note that the head region is smoothed in some of the figures, in order to hide the identity of the scan subjects; the complete scans were used in the learning algorithm.

Correspondence The next step in the data acquisition pipeline brings the template mesh into *correspondence* with each of the other mesh instances. Current non-rigid registration algorithms require that a set of corresponding markers between each instance mesh and the template is available (the work of Allen *et al.* [2003] uses about 70 markers for registration). We obtain the markers using an algorithm called Correlated Correspondence (*CC*) [Anguelov et al. 2005]. The CC algorithm computes the consistent embedding of each instance mesh into the template mesh, which minimizes deformation, and matches similar-looking surface regions. To break the scan symmetries, we initialize the CC algorithm by placing 4–10 markers by hand on each pair of scans. The result of the algorithm is a set of 140–200 (approximate) correspondence markers between the two surfaces, as illustrated in Fig. 2(c).

Non-rigid Registration Given a set of markers between two meshes, the task of non-rigid registration is well understood and a variety of algorithms exist [Allen et al. 2002; Hähnel et al. 2003; Sumner and Popović 2004]. The task is to bring the meshes into close alignment, while simultaneously aligning the markers. We apply a standard algorithm [Hähnel et al. 2003] to register the template mesh with all of the meshes in our data set. As a result, we obtain a set of meshes with the same topology, whose shape approximates well the surface in the original Cyberware scans. Several of the resulting meshes are displayed in Fig. 2(d).

Recovering the Articulated Skeleton As discussed in the introduction, our model uses a low degree-of-freedom skeleton to model the articulated motion. We construct a skeleton for our template mesh automatically, using only the meshes in our data set. We applied the algorithm of [Anguelov et al. 2004], which uses a set of registered scans of a single subject in a variety of configurations. The algorithm exploits the fact that vertices on the same skeleton joint are spatially contiguous, and exhibit similar motion across the different scans. It automatically recovers a decomposition of the object into approximately rigid parts, the location of the parts in the different object instances, and the articulated object skeleton linking the parts. Based on our pose data set, the algorithm automatically constructed a skeleton with 18 parts. The algorithm broke both the crotch area and the chest area into two symmetric parts, resulting in a skeleton which was not tree-structured. To facilitate pose editing, we combined the two parts in each of these regions into one. The result was a tree-structured articulated skeleton with 16 parts.

Data Format and Assumptions The resulting data set consists of a model mesh X and a set of instance meshes $Y = \{Y^1, \ldots, Y^N\}$. The model mesh $X = \{V_X, P_X\}$ has a set of vertices $V_X = \{x_1, \ldots, x_M\}$ and a set of triangles $P_X = \{p_1, \ldots, p_P\}$. The instance meshes are of two types: scans of the same person in various poses, and scans of multiple people in approximately the same pose.

As a result of our pre-processing, we can assume that each instance mesh has the same set of points and triangles as the model mesh, albeit in different configurations. Thus, let $Y^i = \{y_1^i, \ldots, y_M^i\}$ be the set of points in instance mesh Y^i. As we also mapped each of the instance meshes onto our articulated model in the pre-processing phase, we also have, for each mesh Y^i, a set of absolute rotations R^i for the rigid parts of the model, where R_ℓ^i is the rotation of joint ℓ in instance i.

The data acquisition and pre-processing pipeline provides us with exactly this type of data; however, any other technique for generating similar data will also be applicable to our learning and shape completion approach.

4 Pose Deformation

This and the following sections describe the SCAPE model, which is the main contribution of this paper. In the SCAPE model, deformations due to changes in pose and body shape are modeled separately. In this section, we focus on learning the pose deformation model.

4.1 Deformation Process

We want to model the deformations which align the template with each mesh Y^i in the data set containing different poses of a human. The deformations are modeled for each triangle p_k of the template. We use a two-step, translation invariant representation of triangle deformations, accounting for a non-rigid and a rigid component of the deformation. Let triangle p_k contain the points $x_{k,1}, x_{k,2}, x_{k,3}$. We apply our deformations in terms of the triangle's local coordinate system, obtained by translating point $x_{k,1}$ to the global origin. Thus, the deformations will be applied to the triangle edges $\hat{v}_{k,j} = x_{k,j} - x_{k,1}$, $j = 2, 3$.

First, we apply a 3×3 linear transformation matrix Q_k^i to the triangle. This matrix, which corresponds to a non-rigid pose-induced deformation, is specific to each triangle p_k and each pose Y_i. The deformed polygon is then rotated by R_ℓ^i, the rotation of its rigid part in the articulated skeleton. The same rotation is applied to all triangles that belong to that part. Letting $\ell[k]$ be the body part associated

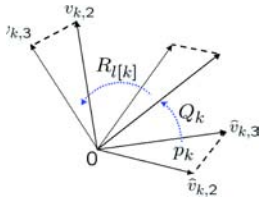

Figure 3: An illustration of our model for triangle deformation.

with triangle p_k, we can write:

$$v^i_{k,j} = R^i_{\ell[k]} Q^i_k \hat{v}_{k,j}, \;\; j = 2,3 \tag{1}$$

The deformation process is sketched in Fig. 3. A key feature of this model is that it combines an element modeling the deformation of the rigid skeleton, with an element that allows for arbitrary local deformations. The latter is essential for modeling muscle deformations.

Given a set of transformation matrices Q and R associated with a pose instance, our method's predictions can be used to synthesize a mesh for that pose. For each individual triangle, our method makes a prediction for the edges of p_k as $R_k Q_k \hat{v}_{k,j}$. However, the predictions for the edges in different triangles are rarely consistent. Thus, to construct a single coherent mesh, we solve for the location of the points $y_1, \ldots, y_M$ that minimize the overall least squares error:

$$\underset{y_1,\ldots,y_M}{\operatorname{argmin}} \sum_k \sum_{j=2,3} \| R^i_{\ell[k]} Q^i_k \hat{v}_{j,k} - (y_{j,k} - y_{1,k}) \|^2 \tag{2}$$

Note that, as translation is not directly modeled, the problem has a translational degree of freedom. By anchoring one of the points y (in each connected component of the mesh) to a particular location, we can make the problem well-conditioned, and reconstruct the mesh in the appropriate location. (See [Sumner and Popović 2004] for a very related discussion on mesh reconstruction from a set of deformation matrices.)

4.2 Learning the Pose Deformation Model

We showed how to model pose-induced deformations using a set of matrices Q^i_k for the template triangles p_k. We want to predict these deformations from the articulated human pose, which is represented as a set of relative joint rotations. If R_{ℓ_1} and R_{ℓ_2} are the absolute rotation matrices of the two rigid parts adjacent to some joint, the relative joint rotation is simply $R^T_{\ell_1} R_{\ell_2}$.

Joint rotations are conveniently represented with their twist coordinates. Let M denote any 3×3 rotation matrix, and let m_{ij} be its entry in i-th row and j-th column. The twist t for the joint angle is a 3D vector, and can be computed from the following formula [Ma et al. 2004]:

$$\begin{aligned} t &= \frac{\|\theta\|}{2 \sin \|\theta\|} \begin{bmatrix} m_{32} - m_{23} \\ m_{13} - m_{31} \\ m_{21} - m_{12} \end{bmatrix} \\ \text{with } \theta &= \cos^{-1}\left(\frac{\operatorname{tr}(M) - 1}{2} \right). \end{aligned}$$

The direction of the twist vector represents the axis of rotation, and the magnitude of the twist represents the rotation amount.

We learn a regression function for each triangle p_k which predicts the transformation matrices Q^i_k as a function of the twists of its two nearest joints $\triangle r^i_{\ell[k]} = (\triangle r^i_{\ell[k],1}, \triangle r^i_{\ell[k],2})$. By assuming that a matrix Q^i_k can be predicted in terms of these two joints only, we greatly reduce the dimensionality of the learning problem.

Each joint rotation is specified using three parameters, so altogether $\triangle r^i_{\ell[k]}$ has six parameters. Adding a term for the constant bias, we associate a 7×1 regression vector $\mathbf{a}_{k,lm}$ with each of the 9 values of the matrix Q, and write:

$$q^i_{k,lm} = \mathbf{a}^T_{k,lm} \cdot \begin{bmatrix} \triangle r^i_{\ell[k]} \\ 1 \end{bmatrix} \qquad l,m = 1,2,3 \tag{3}$$

Thus, for each triangle p_k, we have to fit 9×7 entries $\mathbf{a}_k = (\mathbf{a}_{k,lm} \;:\; l,m = 1,2,3)$. With these parameters, we will have $Q^i_k = \mathscr{Q}_{\mathbf{a}_k}(\triangle r^i_{\ell[k]})$.

Our goal now is to learn these parameters $a_{k,lm}$. If we are given the transformation Q^i_k for each instance Y^i and the rigid part rotations R^i, solving for the regression values (using a quadratic cost function) is straightforward. It can be carried out for each triangle k and matrix value $q_{k,lm}$ separately:

$$\underset{\mathbf{a}_{k,lm}}{\operatorname{argmin}} \sum_i \left([\triangle r^i \;\; 1] \mathbf{a}_{k,lm} - q^i_{k,lm} \right)^2. \tag{4}$$

In practice, we can save on model size and computation by identifying joints which have only one or two degrees of freedom. Allowing those joints to have three degrees of freedom can also cause overfitting in some cases. We performed PCA on the observed angles of the joints $\triangle r^i$, removing axes of rotation whose eigenvalues are smaller than 0.1. The associated entries in the vector $\mathbf{a}_{k,lm}$ are then not estimated. The value 0.1 was obtained by observing a plot of the sorted eigenvalues. We found that the pruned model minimally increased cross-validation error, while decreasing the number of parameters by roughly one third.

As discussed, the rigid part rotations are computed as part of our preprocessing step. Unfortunately, the transformations Q^i_k for the individual triangles are not known. We estimate these matrices by fitting them to the transformations observed in the data. However, the problem is generally underconstrained. We follow Sumner *et al.* [2004] and Allen *et al.* [2003], and introduce a smoothness constraint which prefers similar deformations in adjacent polygons that belong to the same rigid part. Specifically, we solve for the correct set of linear transformations with the following equation for each mesh Y^i:

$$\begin{aligned} \underset{\{Q^i_1,\ldots,Q^i_P\}}{\operatorname{argmin}} \quad & \sum_k \sum_{j=2,3} \| R^i_k Q^i_k \hat{v}_{k,j} - v^i_{k,j} \|^2 + \\ & w_s \sum_{k_1,k_2 \text{ adj}} I(\ell_{k_1} = \ell_{k_2}) \cdot \| Q^i_{k_1} - Q^i_{k_2} \|^2, \end{aligned} \tag{5}$$

where $w_s = 0.001\rho$ and ρ is the resolution of the model mesh X. Above, $I(\cdot)$ is the indicator function. The equation can be solved separately for each rigid part and for each row of the Q matrices.

Given the estimated Q matrices, we can solve for the (at most) 9×7 regression parameters $\mathbf{a}_k$ for each triangle k, as described in Eq. (4).

4.3 Application to Our Data Set

We applied this method to learn a SCAPE pose deformation model using the 70 training instances in our pose data set. Fig. 4 shows examples of meshes that can be represented by our learned model. Note that these examples do not correspond to meshes in the training data set; they are new poses synthesized completely from a vector of joint rotations R, using Eq. (3) to define the Q matrices, and Eq. (2) to generate the mesh.

The model captures well the shoulder deformations, the bulging of the biceps and the twisting of the spine. It deals reasonably well with the elbow and knee joints, although example (g) illustrates a small amount of elbow smoothing that occurs in some poses. The model exhibits an artifact in the armpit, which is caused by hole-filling in the template mesh.

Generating each mesh given the matrices takes approximately 1 second, only 1.5 orders of magnitude away from real time, opening the possibility of using this type of deformation model for real-time animation of synthesized or cached motion sequences.

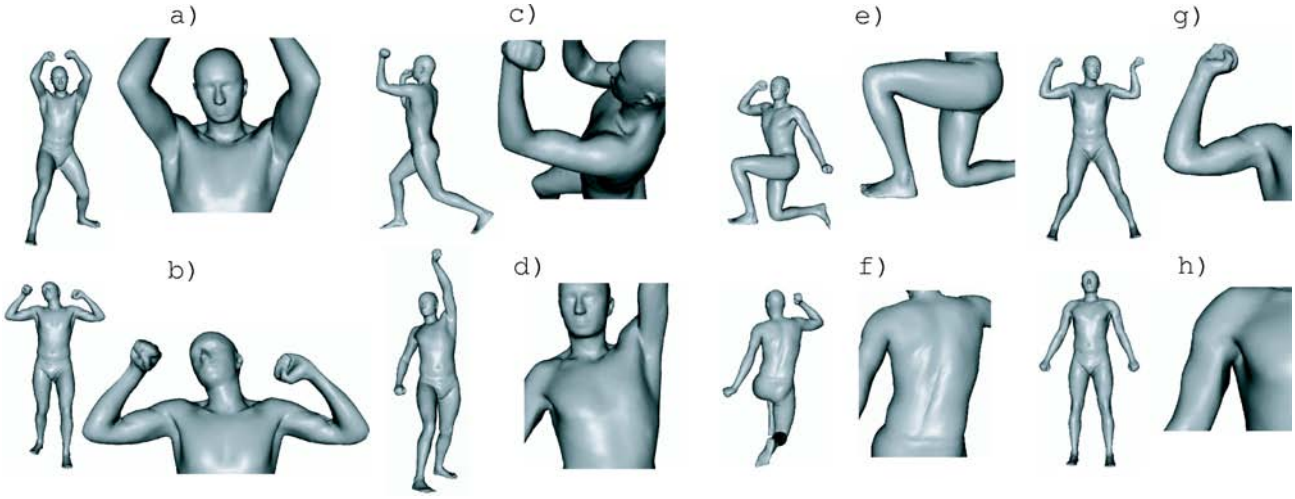

Figure 4: Examples of muscle deformations that can be captured in the SCAPE pose model.

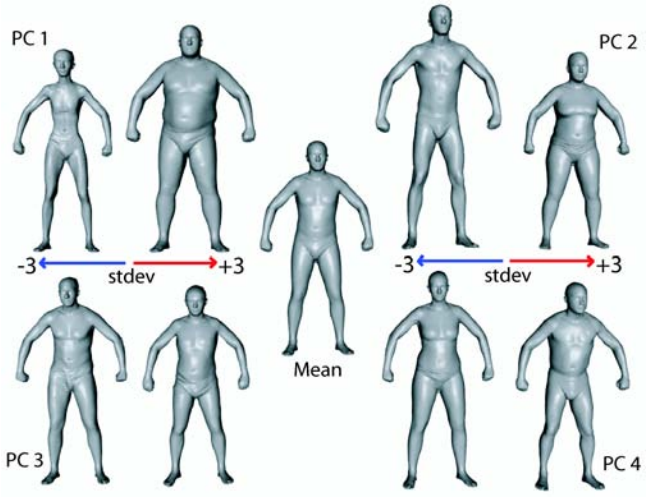

Figure 5: The first four principal components in the space of body shape deformation

5 Body-Shape Deformation

The SCAPE model also encodes variability due to body shape across different individuals. We now assume that the scans of our training set Y^i correspond to different individuals.

5.1 Deformation Process

We model the body-shape variation independently of the pose variation, by introducing a new set of linear transformation matrices S_k^i, one for each instance i and each triangle k. We assume that the triangle p_k observed in the instance mesh i is obtained by first applying the pose deformation Q_k^i, then the body shape deformation S_k^i, and finally the rotation associated with the corresponding joint $R_{\ell[k]}^i$. The application of consecutive transformation matrices maintains proper scaling of deformation. We obtain the following extension to Eq. (1):

$$v_{k,j}^i = R_{\ell[k]}^i S_k^i Q_k^i \hat{v}_{k,j}. \tag{6}$$

The body deformation associated with each subject i can thus be modeled as a set of matrices $S^i = \{S_k^i \,:\, k = 1,\ldots,P\}$.

5.2 Learning the Shape Deformation Model

To map out the space of body shape deformations, we view the different matrices S^i as arising from a lower dimensional subspace. For each example mesh, we create a vector of size $9 \times N$ containing the parameters of matrices S^i. We assume that these vectors are generated from a simple linear subspace, which can be estimated by using PCA:

$$S^i = \mathscr{S}_{U,\mu}(\beta^i) = \overline{U\beta^i + \mu} \tag{7}$$

where $U\beta^i + \mu$ is a (vector form) reconstruction of the $9 \times N$ matrix coefficients from the PCA, and $\overline{U\beta^i + \mu}$ is the representation of this vector as a set of matrices. PCA is appropriate for modeling the matrix entries, because body shape variation is consistent and not too strong. We found that even shapes which are three standard deviations from the mean still look very much like humans (see Fig. 5).

If we are given the affine matrices S_k^i for each i,k we can easily solve for the PCA parameters U, μ, and the mesh-specific coefficients β^i. However, as in the case of pose deformation, the individual shape deformation matrices S_k^i are not given, and need to be estimated. We use the same idea as above, and solve directly for S_k^i, with the same smoothing term as in Eq. (5):

$$\underset{S^i}{\operatorname{argmin}} \sum_k \sum_{j=2,3} \|R_k^i S_k^i Q_k^i \hat{v}_{k,j} - v_{k,j}^i\|^2 + w_s \sum_{k_1,k_2 \text{ adj}} \|S_{k_1}^i - S_{k_2}^i\|^2. \tag{8}$$

Importantly, recall that our data preprocessing phase provides us with an estimate R^i for the joint rotations in each instance mesh, and therefore the joint angles $\triangle r^i$. From these we can compute the predicted pose deformations $Q_k^i = \mathscr{Q}_{\mathbf{a}_k}(\triangle r_{\ell[k]}^i)$ using our learned pose deformation model. Thus, the only unknowns in Eq. (8) are the shape deformation matrices S_k^i. The equation is quadratic in these unknowns, and therefore can be solved using a straightforward least-squares optimization.

5.3 Application to Our Data Set

We applied this method to learn a SCAPE body shape deformation model using the 45 instances in the body shape data set, and taking as a starting point the pose deformation model learned as described in Sec. 4.3. Fig. 5 shows the mean shape and the first four principal components in our PCA decomposition of the shape space. These components represent very reasonable variations in weight and height, gender, abdominal fat and chest muscles, and bulkiness of the chest versus the hips.

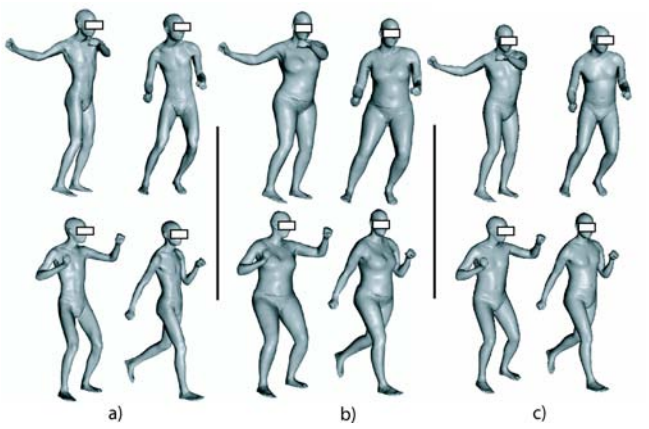

Figure 6: Deformation transfer by the SCAPE model. The figure shows three subjects, each in four different poses. Each subject was seen in a single reference pose only

Our PCA space spans a wide variety of human body shapes. Put together with our pose model, we can now synthesize realistic scans of various people in a broad range of poses. Assume that we are given a set of rigid part rotations R and person body shape parameters β. The joint rotations R determine the joint angles $\triangle R$. For a given triangle p_k, the pose model now defines a deformation matrix $Q_k = \mathscr{Q}_{\mathbf{a}_k}(\triangle r_{\ell[k]})$. The body shape model defines a deformation matrix $S_k = \mathscr{S}_{U,\mu}(\beta)$. As in Eq. (2), we solve for the vertices Y that minimize the objective:

$$E_H[Y] = \sum_k \sum_{j=2,3} \|R_k \mathscr{S}_{U,\mu}(\beta) \mathscr{Q}_{\mathbf{a}_k}(\triangle r_{\ell[k]}) \hat{v}_{j,k} - (y_{j,k} - y_{1,k})\|^2 \quad (9)$$

The objective can be solved separately along each dimension of the points y.

Using this approach, we can generate a mesh for any body shape in our PCA space in any pose. Fig. 6 shows some examples of different synthesized scans, illustrating variation in both body shape and pose. The figure shows that realistic muscle deformation is achieved for very different subjects, and for a broad range of poses.

6 Shape Completion

So far, we have focused on the problem of constructing the two components of our SCAPE model from the training data: the regression parameters $\{\mathbf{a}_k : k = 1,\ldots,P\}$ of the pose model, and the PCA parameters U, μ of our body shape model. We now show how to use the SCAPE model to address the task of shape completion, which is the main focus of our work. We are given sparse information about an instance mesh, and wish to construct a full mesh consistent with this information; the SCAPE model defines a prior on the deformations associated with human shape, and therefore provides us with guidance on how to complete the mesh in a realistic way.

Assume we have a set of markers $Z = z_1,\ldots,z_L$ which specify known positions in 3D for some points $x_1,\ldots,x_L$ on the model mesh. We want to find the set of points Y that best fits these known positions, and is also consistent with the SCAPE model. In this setting, the joint rotations R and the body shape parameters β are also not known. We therefore need to solve simultaneously for Y, R, and β minimizing the objective:

$$E_H[Y] + w_Z \sum_{l=1}^{L} \|y_l - z_l\|^2, \quad (10)$$

where $E_H[Y]$ was defined in Eq. (9) and w_Z is a weighting term that trades off the fit to the markers and the consistency with the model.

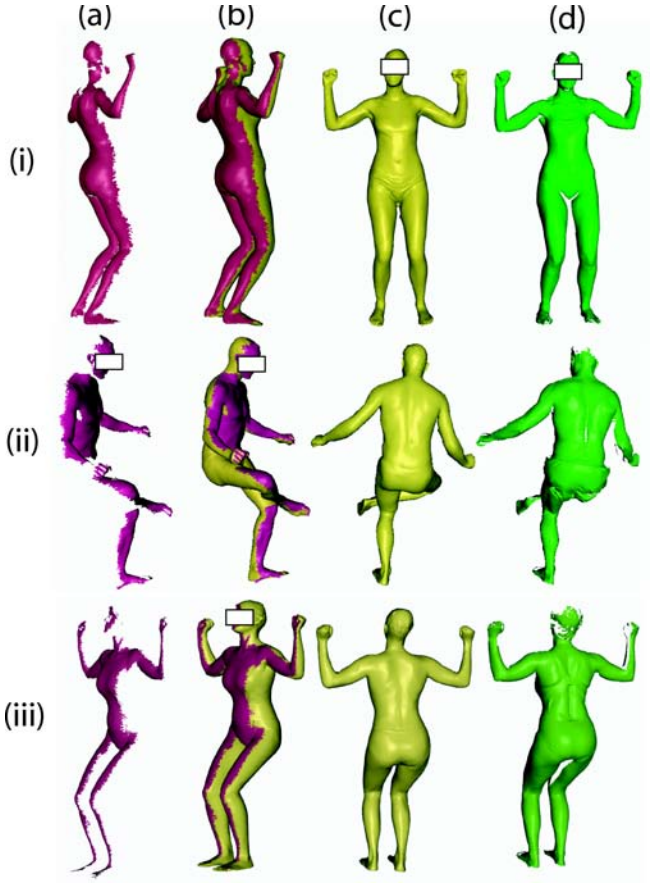

Figure 7: Examples of view completion, where each row represents a different partial view scan. Subject (i) is in our data set but not in the this pose; neither subjects (ii) and (iii) nor their poses are represented in our data set. (a) The original partial view. (b) The completed mesh from the same perspective as (a), with the completed portion in yellow. (c) The completed mesh from a view showing the completed portion. (d) A true scan of the same subject from the view in (c).

A solution to this optimization problem is a *completed mesh* $Y[Z]$ that both fits the observed marker locations and is consistent with the predictions of our learned SCAPE model. It also produces a set of joint rotations R and shape parameters β. Note that these parameters can also be used to produce a predicted mesh $\tilde{Y}[Z]$, as in Sec. 4.3. This predicted mesh is (by definition) constrained to be within our PCA subspace of shapes; thus it generally does not encode some of the details unique to the new (partial) instance mesh to be completed. As we shall see, the predicted mesh $\tilde{Y}[Z]$ can also be useful for smoothing certain undesirable artifacts.

Eq. (10) is a general non-linear optimization problem to which a number of existing optimization techniques can be applied. Our specific implementation of the optimization is intended to address the fact that Eq. (10) is non-linear and non-convex, hence is subject to local minima. Empirically, we find that care has to be taken to avoid local minima. Hence, we devise an optimization routine that slows the adaptation of certain parameters in the optimization, thereby avoiding the danger of converging to sub-optimal shape completions. In particular, optimizing over all of the variables in this equation using standard non-linear optimization methods is not a good idea. Our method uses an iterative process, where it optimizes each of the three sets of parameters (R, β, and Y) separately, keeping the others fixed.

The resulting optimization problem still contains a non-linear optimization step, due to the correlation between the absolute part ro-

tations R and the joint rotations $\triangle R$, both of which appear in the objective of Eq. (10). We use an approximate method to deal with this problem. Our approach is based on the observation that the actual joint rotations R influence the point locations much more than their (fairly subtle) effect on the pose deformation matrices via $\triangle R$. Thus, we can solve for R while ignoring the effect on $\triangle R$, and then update $\triangle R$ and the associated matrices $\mathscr{Q}_{\mathbf{a}}(\triangle R)$. This approximation gives excellent results, as long as the value of $\triangle R$ does not change much during each optimization step. To prevent this from happening, we add an additional term to the objective in Eq. (10). The term penalizes steps where adjacent parts (parts that share a joint) move too differently from each other.

Specifically, when optimizing R, we approximate rotation using the standard approximation $R^{new} \approx (I+\hat{\mathbf{t}})R^{old}$, where $\mathbf{t} = (t_1, t_2, t_3)$ is a *twist vector*, and

$$\hat{\mathbf{t}} = \begin{pmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{pmatrix} \tag{11}$$

Let $\mathbf{t}_\ell$ denote the twist vector for a part ℓ. The term preventing large joint rotations then is simply $\sum_{\ell_1,\ell_2 \text{ adj}} \|\mathbf{t}_{\ell_1} - \mathbf{t}_{\ell_2}\|^2$.

We are now ready to state the overall optimization technique applied in our work. This techniques iteratively repeats three steps:

- We update R, resulting in the following equation:

$$\underset{\mathbf{t}}{\operatorname{argmin}} \sum_k \sum_{j=2,3} \|(I+\hat{\mathbf{t}}_{\ell_k})R^{old}SQ\hat{v}_{j,k} - (y_{j,k} - y_{1,k})\|^2 + w_T \sum_{\ell_1,\ell_2 \text{ adj}} \|\mathbf{t}_{\ell_1} - \mathbf{t}_{\ell_2}\|^2$$

  Here $S = \mathscr{S}_{U,\mu}(\beta)$ according to the current value of β, $Q = \mathscr{Q}_{\mathbf{a}_k}(\triangle r_{\ell[k]})$ where $\triangle R$ is computed from R^{old}, and w_T is an appropriate trade-off parameter.

  After each update to R, we update $\triangle R$ and Q accordingly.

- We update Y to optimize Eq. (10), with R and β fixed. In this case, the S and Q matrices are determined, and the result is a simple quadratic objective that can be solved efficiently using standard methods.

- We update β to optimize Eq. (10). In this case, R and the Q matrices are fixed, as are the point positions Y, so that the objective reduces to a simple quadratic function of β:

$$\sum_k \sum_{j=2,3} \|R_k\overline{(U\beta+\mu)}_k Q\hat{v}_{j,k} - (y_{j,k} - y_{1,k})\|^2 \tag{12}$$

This optimization process converges to a local optimum of the objective Eq. (10).

7 Partial View Completion

An obvious application of our shape completion method is to the task of partial view completion. Here, we are given a partial scan of a human body; our task is to produce a full 3D mesh which is consistent with the observed partial scan, and provides a realistic completion for the unseen parts.

Our shape completion algorithm of Sec. 6 applies directly to this task. We take the partial scan, and manually annotate it with a small number of markers (4–10 markers, 7 on average). We then apply the CC algorithm [Anguelov et al. 2004] to register the partial scan to the template mesh. The result is a set of 100–150 markers, mapping points on the scan to corresponding points on the template mesh. This number of markers is sufficient to obtain a reasonable initial hypothesis for the rotations R of the rigid skeleton. We then iterate between two phases. First, we find point-to-point correspondences between the partial view and our current estimate of the surface $Y[Z]$. Then we use these correspondences as markers and solve Eq. (10) to obtain a new estimate $Y[Z]$ of the surface. Upon convergence, we obtain a completion mesh $Y[Z]$, which fits the partial view surface as well as the SCAPE model.

Fig. 7 shows the application of this algorithm to three partial views. Row (i) shows partial view completion results for a subject who is present in our data set, but in a pose that is not in our data set. The prediction for the shoulder blade deformation is very realistic; a similar deformation is not present in the training pose for this subject. Rows (ii) and (iii) show completion for subjects who are not in our data set, in poses that are not in our data set. The task in row (ii) is particularly challenging, both because the pose is very different from any pose in our data set, and because the subject was wearing pants, which we cut out, (see Fig. 7(ii)-(d)), leading to the large hole in the original scan. Nevertheless, the completed mesh contains realistic deformations in both the back and the legs.

8 Motion Capture Animation

Our shape completion framework can also be applied to produce animations from marker motion capture sequences. In this case, we have a sequence of frames, each specifying the 3D positions for some set of markers. We can view the set of markers observed in each frame as as our input Z to the algorithm of Sec. 6, and use the algorithm to produce a mesh. The sequence of meshes produced for the different frames can be strung together to produce a full 3D animation of the motion capture sequence.

Note that, in many motion capture systems, the markers protrude from the body, so that a reconstructed mesh that achieves the exact marker positions observed may contain unrealistic deformations. Therefore, rather than using the completed mesh $Y[Z]$ (as in our partial view completion task), we use the predicted mesh $\tilde{Y}[Z]$. As this mesh is constrained to lie within the space of body shapes encoded by our PCA model, it tends to avoid these unrealistic deformations.

We applied this data to two motion capture sequences, both for the same subject S. Notably, our data set only contains a single scan for subject S, in the standard position shown in the third row of Fig. 2(a). Each of the sequences used 56 markers per frame, distributed over the entire body. We took a 3D scan of subject S with the markers, and used it to establish the correspondence between the observed markers and points on the subject's surface. We then applied the algorithm of Sec. 6 to each sequence frame. In each frame, we used the previous frame's estimated pose R as a starting point for the optimization. The animation was generated from the sequence of predicted scans $\tilde{Y}[Z_f]$. Using our (unoptimized) implementation, it took approximately 3 minutes to generate each frame. Fig. 8 demonstrates some of our results. We show that realistic muscle deformation was obtained for subject S (Fig. 8(c)). Additionally, we show that motion transfer can be performed onto a different subject in our data set (Fig. 8(d)) and that the subject can be changed during the motion sequence (Fig. 8(e)).

9 Discussion and Limitations

This paper presents the SCAPE model, which captures human shape deformation due to both pose variation and to body shape variation over different subjects. Our results demonstrate that the model can generate realistic meshes for a wide range of subjects and poses. We showed how the SCAPE model can be used for shape completion, and cast two important graphics tasks — partial view completion and motion capture animation — as applications of our shape completion algorithm.

The SCAPE model decouples the pose deformation model and the body shape deformation model. This design choice greatly simplifies the mathematical formulation, improves the identifiability of the model from data, and allows for more efficient learning algorithms. However, it also prevents us from capturing phenomena where there is a strong correlation between body shape and muscle deformation. For example, as the same muscle deformation model

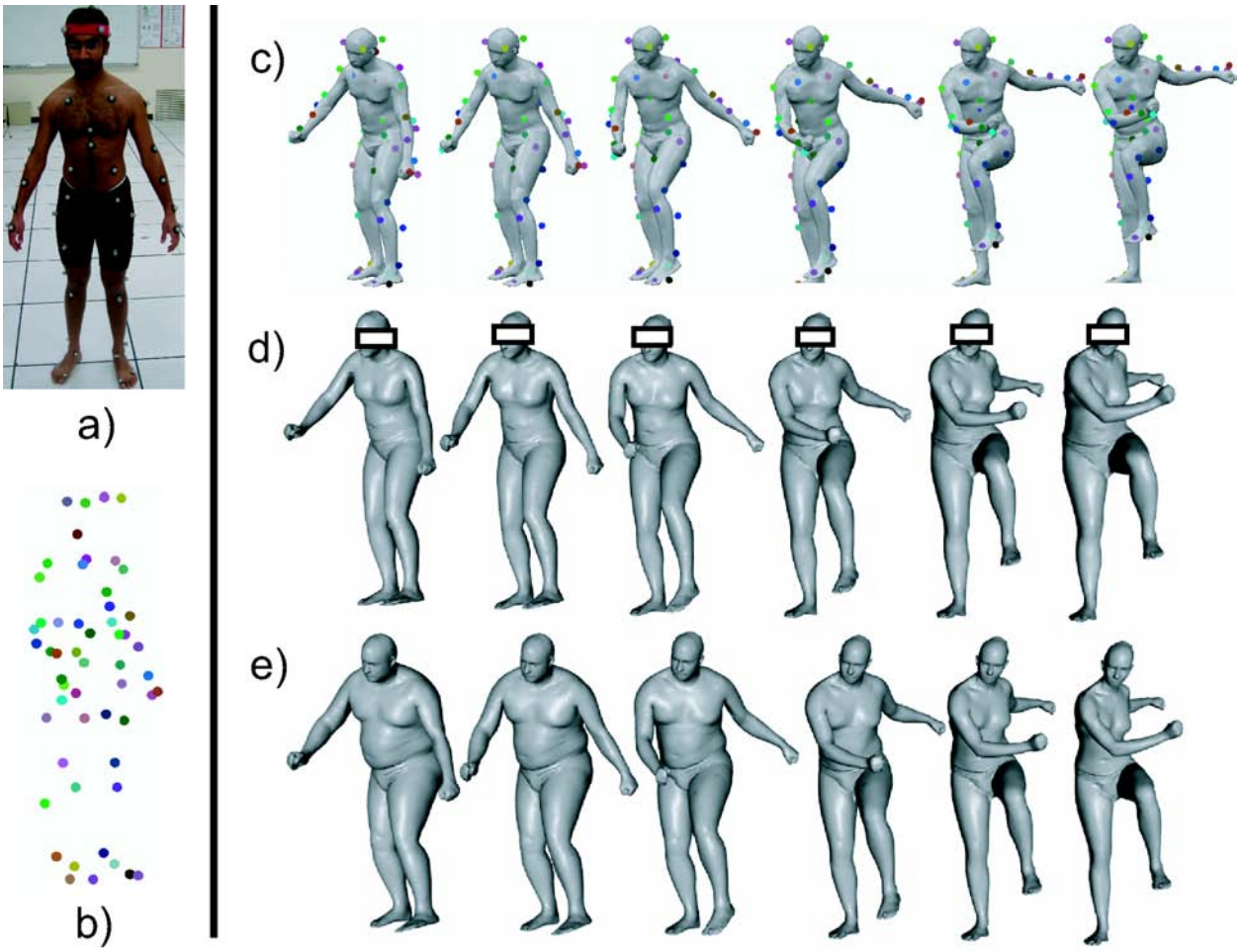

Figure 8: Motion capture animation. (a) Subject wearing motion capture markers (b) Motion capture markers in a single frame (c) An animation of a subject based on a motion capture sequence, with the markers from which the animation was derived superimposed on the meshes. (d) An example of motion transfer to a different subject in our data set. (e) Animation based on motion capture, but where we change the body shape parameters in PCA space as we move through the sequence.

is used for all people, we do not capture the fact that more muscular people are likely to exhibit greater muscle deformation than others, and, conversely, that muscle deformation may be obscured in people with significant body fat. To capture these correlations, a more expressive model would be needed.

Our current approach requires a set of scans of a single person in different poses to learn the space of pose deformations. Once we have done this, we can use scans of different people in different poses to learn the space of body shapes. We are currently not providing a method to learn both spaces from a random mix of scans from different people in different poses. Our assumption on the training set structure is not particularly restrictive, and it simplifies our data collection and learning procedures. We could try to learn our model from a non-uniform data set, by iterating between estimating either the pose or the body shape model while keeping the other one fixed. This process would result in a local minimum in the joint space of deformations. We cannot predict how good this local minimum would be; it depends specifically on the training data we are given, and on the search method used.

The pose deformation in our model is determined by linear regression from adjacent joint angles. We found that this assumption provides surprisingly good animation results, and simplifies the task of shape completion. For many instances of partial view completion, a more accurate model may not be necessary, because our solution is allowed to deform outside of SCAPE space in order to fit the observed surface. Thus, partial view data can correct some of the (fairly small) errors resulting from the assumption of a linear regression model. When the SCAPE model is used purely for animation, a more complex model may be required in some cases. Extending our approach with a non-linear regression method is an area of our future work.

The SCAPE model is focused on representing muscle deformations resulting from articulated body motion. Deformations resulting from other factors are not encoded. One such factor is deformation resulting from pure muscle activity. Thus, the model is not expressive enough to distinguish between a flexed bicep muscle and a lax one in cases where the joint angle is the same. For the same reason, it is not appropriate to deal with faces, where most of the motion is purely muscle-based. Another factor leading to muscle deformation is tissue perturbations due to motion (e.g., fat wiggling), which our model also does not represent.

Currently, our framework includes no prior over poses. Thus, when encountering occlusions, we cannot use the observed position of some body parts to constrain the likely location of others. Our model can easily be extended to encompass such a prior, in a modular way. For example, in the case of static scans, a kinematic prior such as that of Popović *et al.*[2004] could simply be introduced as an additional term into our optimization. When animating dynamic sequences, we can use a tracking algorithm (e.g., a Kalman filter) to generate a pose prior for any frame given all or part of the observa-

tion sequence.

Finally, we note that our approach is purely data driven, generating the entire model from a set of data scans. Human intervention is required only for placing a small set of markers on the scans, as a starting point for registration. Thus, the model can easily be applied to other data sets, allowing us to generate models specific to certain types of body shapes or certain poses. Moreover, the framework applies more generally to cases where surface deformation is derived from articulated motion. Thus, if we could solve the data acquisition problem (e.g., using shape from silhouette [Cheung et al. 2003]), we could use this framework to learn realistic deformation models for creatures other than humans.

Acknowledgements

This work was supported by by the ONR Young Investigator (PECASE) grant N00014-99-1-0464, and ONR Grant N00014-00-1-0637 under the DoD MURI program. We would like to acknowledge the indispensable help of Lars Mündermann, Stefano Corazza and the Stanford Biomechanics group, who provided us with the scan data.

References

ALLEN, B., CURLESS, B., AND POPOVIĆ, Z. 2002. Articulated body deformation from range scan data. *ACM Transactions on Graphics*, 21(3), 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*, 22(3), 587–594.

ANGUELOV, D., KOLLER, D., PANG, H., SRINIVASAN, P., AND THRUN, S. 2004. Recovering articulated object models from 3d range data. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, 18–26.

ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., PANG, H., AND DAVIS, J. 2005. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Advances in Neural Information Processing Systems 17*, 33–40.

CHEUNG, K. M., BAKER, S., AND KANADE, T. 2003. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 77–84.

CURLESS, B., AND LEVOY, M. 1996. A volumetric method of building complex models from range images. *Proceedings of SIGGRAPH 1996*, 303–312.

DAVIS, J., MARSCHNER, S., GARR, M., AND LEVOY, M. 2002. Filling holes in complex surfaces using volumetric diffusion. In *Symposium on 3D Data Processing, Visualization, and Transmission*.

GARLAND, M., AND HECKBERT, P. S. 1997. Surface simplification using quadric error metrics. In *Proceedings of SIGGRAPH 97*, 209–216.

HÄHNEL, D., THRUN, S., AND BURGARD, W. 2003. An extension of the ICP algorithm for modeling nonrigid objects with mobile robots. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

HILTON, A., STARCK, J., AND COLLINS, G. 2002. From 3d shape capture to animated models. In *First International Symposion on 3D Data Processing, Visualization and Transmission (3DVPT2002)*.

KÄHLER, K., HABER, J., YAMAUCHI, H., AND SEIDEL, H.-P. 2002. Head shop: generating animated head models with anatomical structure. In *ACM SIGGRAPH Symposium on Computer Animation*, 55–64.

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

LIEPA, P. 2003. Filling holes in meshes. In *Proc. of the Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 200–205.

MA, Y., SOATTO, S., KOSECKA, J., AND SASTRY, S. 2004. *An Invitation to 3D Vision*. Springer Verlag.

MOHR, A., AND GLEICHER, M. 2003. Building efficient, accurate character skins from examples. *ACM Transactions on Graphics*, 22(3), 562–568.

NOH, J., AND NEUMANN, U. 2001. Expression cloning. *Proceedings of ACM SIGGRAPH 2001*, 277–288.

POPOVIĆ, Z., GROCHOW, K., MARTIN, S. L., AND HERTZMANN, A. 2004. Style-based inverse kinematics. *ACM Transactions on Graphics*, 23(3), 522–531.

SAND, P., MCMILLAN, L., AND POPOVIĆ, J. 2003. Continuous capture of skin deformation. *ACM Transactions on Graphics*, 22(3), 578–586.

SEO, H., AND MAGNENAT-THALMANN, N. 2003. An automatic modeling of human bodies from sizing parameters. In *ACM Symposium on Interactive 3D Graphics*, 19–26.

SLOAN, P.-P. J., ROSE, C. F., AND COHEN, M. F. 2001. Shape by example. In *2001 Symposium on Interactive 3D Graphics*, 135–144.

SUMNER, R. W., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. *Proceedings of ACM SIGGRAPH 2004*, 23(3), 399–405.

SZELISKI, R., AND LAVALLEE, S. 1996. Matching 3-d anatomical surfaces with non-rigid deformations using using octree-splines. *International Journal of Computer Vision 18*, 2, 171–186.

VASILESCU, M., AND TERZOPOULOS, D. 2002. Multilinear analysis of image ensembles: Tensorfaces. In *European Conference on Computer Vision (ECCV)*, 447–460.

VLASIC, D., PFISTER, H., BRAND, M., AND POPOVIĆ, J. 2004. Multilinear models for facial synthesis. In *SIGGRAPH Research Sketch*.

WANG, X. C., AND PHILLIPS, C. 2002. Multi-weight enveloping: least-squares approximation techniques for skin animation. In *ACM SIGGRAPH Symposium on Computer Animation*, 129–138.