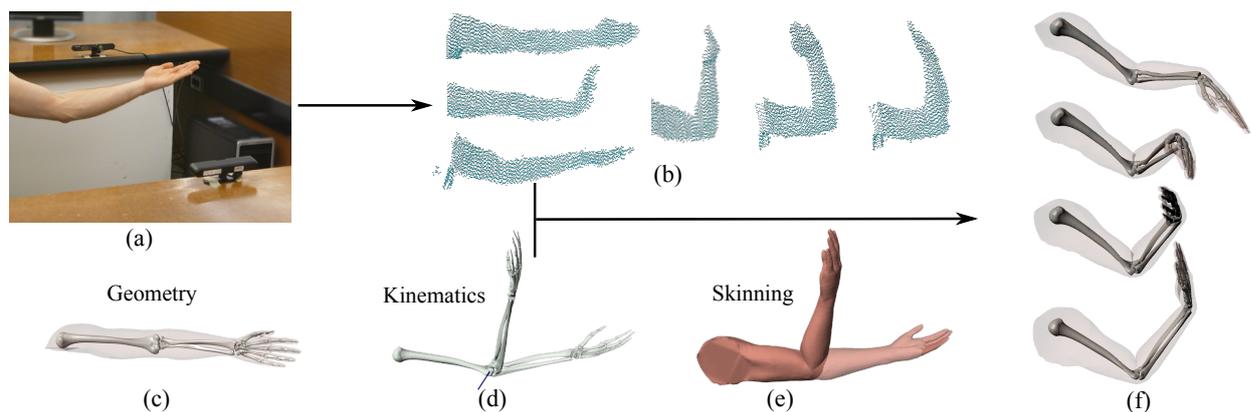


# Adaptable Anatomical Models for Realistic Bone Motion Reconstruction

Lifeng Zhu, Xiaoyan Hu and Ladislav Kavan

University of Pennsylvania



**Figure 1:** Our system starts by capturing animated point clouds using commodity depth-sensors (a). These point clouds (b) are used to learn the subject-specific parameters of our anatomical model, which consists of geometry (c), kinematics (d), and skinning (e). We apply our final personalized model to reconstruct anatomically-plausible bone motion (f).

## Abstract

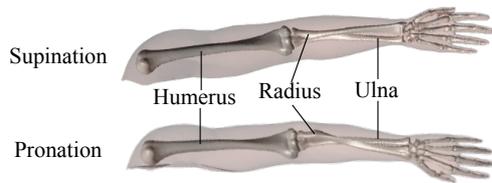
We present a system to reconstruct subject-specific anatomy models while relying only on exterior measurements represented by point clouds. Our model combines geometry, kinematics, and skin deformations (skinning). This joint model can be adapted to different individuals without breaking its functionality, i.e., the bones and the skin remain well-articulated after the adaptation. We propose an optimization algorithm which learns the subject-specific (anthropometric) parameters from input point clouds captured using commodity depth cameras. The resulting personalized models can be used to reconstruct motion of human subjects. We validate our approach for upper and lower limbs, using both synthetic data and recordings of three different human subjects. Our reconstructed bone motion is comparable to results obtained by optical motion capture (Vicon) combined with anatomically-based inverse kinematics (OpenSIM). We demonstrate that our adapted models better preserve the joint structure than previous methods such as OpenSIM or Anatomy Transfer.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Three Dimensional Graphics and Realism—Animation

## 1. Introduction

The human body evolved through millennia to create a fascinating mechanical device. However, the inner workings of bones, muscles, and soft tissues are hidden to the naked eye. During normal motions, the bones behave as rigid bodies. It

is interesting to think about what this means during everyday activities such as twisting the arm when turning a doorknob. The forearm contains two major bones, the radius and the ulna, see Fig. 2. During pronation and supination, the approximately 180° twist of the hand is achieved by the radius revolving around the ulna. Because the radius is rigid, the



**Figure 2:** Bones of the upper extremity during supination and pronation.

same amount of twist must occur also in the elbow joint. We invite the reader to observe their own arm while pronating and supinating. We see that the skin near the elbow barely moves at all, revealing very little about the dramatic rotation of the radius that occurs inside the body.

Anatomical models lead to stunning levels of visual realism in computer graphics, as evidenced by visual effects e.g. in the movie *Avatar* [CJ10]. Unfortunately, anatomically-based modeling is very laborious and requires skilled technical artists, which makes this technology hardly accessible outside of high-budget production. We hope our method can democratize anatomically-based models in computer graphics. Our reconstructed bone motion can be used as boundary conditions driving physics-based musculoskeletal simulations [FLP14]. We hope that our method will pave the way to exciting future applications such as personalized medical visualization or animated medical atlases.

Highly accurate skeletal motion can be observed using imaging devices based on X-ray or MRI technology. While real-time MRI does not suffer from the risks of ionizing radiation, the equipment and its operation are very expensive and available only in major health care centers. Even if budget were not a concern, MRI machines offer only a small workspace which limits the range of captured motions. While anatomically-accurate bone motion capture seems to be out of reach in computer animation, we argue that for many applications in computer graphics and visualization, high accuracy is not necessary as long as the motions look anatomically realistic. This is similar to physics-based animation, which does not aim for physically accurate but only physically plausible results.



**Figure 3:** An animation skeleton compared to a realistic one.

However, skeleton models typically used in computer animation are only very crude approximations of the real human anatomy, see Fig. 3. Our main contribution is an *adaptable* anatomically-based model describing geometry, kinematics,

and deformation behavior of the skin – in a unified way. Our model starts with polygon meshes representing realistic bones and skin in a rest pose (Fig. 1(c)). The motion of the bones is parameterized using standard kinematic chains. In particular, we use the kinematic models developed in the biomechanics community [DAA\*07] (OpenSIM), which describe anatomically-realistic joints (Fig. 1(d)). The geometry and kinematics is complemented by a skinning model, which relates articulations of the bones to the deformations of the skin (Fig. 1(e)). Surprisingly, we found that existing skinning techniques struggle with anatomically-realistic bones, e.g., in the forearm (often approximated by only one “bone” in typical computer graphics models). We propose to resolve this challenge by using custom deformers. In particular, we design a novel deformer for the forearm which works well with realistic radius and ulna kinematics.

Our anatomically based model is parameterized by “anthropomorphic” parameters, which account for the variations of human bodies in terms of size and shape, depending on age, gender, ethnicity, and other anthropological factors. The template anatomy model must be *adapted* to match the size and proportions of a specific individual whose motion we intend to capture. This adaptation must be done carefully because it is very easy to destroy the anatomical realism of the model and produce artifacts such as self-intersecting bones. As input, we use 3D point clouds corresponding to multiple poses of the human performer. Multiple poses help us disambiguate the location and shape of the bones. For example, the humerus head (“pointy elbow”) is much more prominent with a bent elbow than with a straight arm. We present an algorithm which optimizes for pose parameters (joint angles), shape deformation parameters (e.g., bone lengths), and rest pose skin geometry in order to align the adapted template with the example shapes as closely as possible. Our template adaptation can be viewed as a way to approximate bone geometry without any medical imaging (X-ray or MRI), relying solely on consumer-grade depth sensors. The recent Anatomy Transfer paper [DLG\*13] shares a similar goal, but relies on perfect meshes and correspondences provided by the user. In spite of using only unstructured and noisy point clouds, our technique results in higher anatomical fidelity of the reconstructed bones (Fig. 20).

After adapting our anatomical model to a specific human subject, we use it for tracking an input motion sequence, captured with commodity depth sensors. Alternatively, it would also be possible to use traditional marker-based systems in this phase, however, we continue to rely on our depth sensors which are much more easily accessible than professional systems such as Vicon. We experiment with two types of body parts: the upper and lower limb. The bone motions reconstructed using our method look natural and the structure of the moving joints is well preserved. This is possible due to our anatomical models adapted to specific human subjects which combine geometry, kinematics, and skinning. In the

future, we hope that our work will inspire synergies between computer graphics and biomechanics.

## 2. Related Work

Reconstruction of human motion is a long standing problem in graphics, vision, and biomechanics, see [Pop07, MHKS11] for a survey. In this section we focus only on the most closely related work. Discriminative pose estimation methods, such as those used by the Microsoft Kinect [SSK\*13], are very robust but do not achieve high accuracy [HBB\*13]. Using a high-quality template model, impressive results have been demonstrated both for full-body characters [YLH\*12, WZC12] and human hands [BTG\*12]. An accurate template model can be constructed by laser scanning or multi-view stereo reconstruction of the human subject in a rest pose. Corresponding animation skeletons are then created manually. These skeletons are not intended to be anatomically realistic, their role is to provide a model explaining surface deformations. Even highly simplified linearized skeletal models are useful in tracking [SHRB12]. Data-driven approaches devise a statistical model from a database of human body shapes [ASK\*05, HSS\*09], which can be used as an alternative method to obtain a subject-specific model, e.g., in full body tracking [HBB\*13]. However, these databases do not contain anatomically-realistic bone models.

An important ingredient of human body models is *skinning*, i.e., a mapping between pose parameters and the resulting deformed skin. Commonly used skinning techniques are linear blend skinning [YLH\*12, BTG\*12] or dual quaternion skinning [LGS\*13]. We found that previous skinning techniques are not sufficient with anatomically-realistic skeletons, especially in challenging areas such as the forearm, where the skin is affected by a complex interplay between two major bones. Inspired by [KS12], we propose to explain skin motion using general deformer. In particular, we propose a novel deformer to describe anatomically-realistic forearm deformations. In contrast to [KS12], our deformer is not joint-based, but applies to the entire forearm region.

In this paper, we assume that our input point clouds correspond to samples of the skin, i.e., the human subjects are not wearing any loose clothing. This assumption has been relaxed in performance capture techniques [VBMP08, DAST\*08, GSDA\*09], which allow the reconstructed surface to depart from the template model in order to better fit the input data. The method due to Li and colleagues [LLV\*12] is even more flexible and allows for changes of topology, enabling effects such as gliding cloth or exposure of previously covered body parts.

In biomechanics and other applications where accuracy is important, professional marker-based systems such as Vicon or Motion Analysis are often the method of choice. Skeletal motion reconstruction from sparse markers is a well stud-

ied problem [LO99], but the accuracy is limited due to the assumption that markers are rigidly attached to the bones. The resulting “soft tissue artifacts” are a prevalent problem in biomechanics [GKM\*07, Tod07]. To overcome these artifacts, some researchers resorted to attaching markers to pins drilled into their own bones [NJL\*07]. In this paper, we propose to combat the soft tissue artifacts by anatomically-based skinning models, which predict the shape of the entire outer surface. In contrast to typical computer animation setups, our skinning model uses anatomically-realistic kinematic models adapted from the OpenSIM platform [DAA\*07]. These kinematic models have been established via cadaver studies, stereo-photogrammetry, and in vivo fluoroscopy. In particular we use models for the upper [HMD05] and lower extremity [DLH\*90]. Our method compares favorably to skeletal motion reconstruction obtained using Vicon and the inverse kinematics module of OpenSIM.

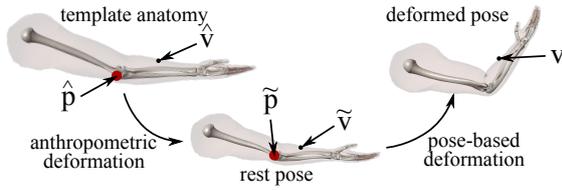
Marker-less motion capture has been studied by Corazza et al. [CMC\*06, CGMA10, CMG\*10, CSC14]. Their system uses a multi-camera setup tracking approximately rigid body parts using iterative closest points; deformable parts such as joints are excluded from tracking. In contrast, our skinning model allows us to track these deformable parts as well, which is very helpful in locating the underlying bones. While Corazza and colleagues achieve impressive full-body tracking results, their work does not model skinning or anatomically-realistic kinematics.

The most accurate way to obtain subject-specific anatomical templates is using medical imaging, such as static MRI [GRP10, DGFP12]. However, MRI data are often unavailable. In this case, the recent work of Dicko and colleagues [DLG\*13] can be used to adapt a template anatomy model to a different body shape specified only by its boundary. However, they assume the target skin is specified by a clean polygon mesh with perfect correspondences. In spite of the fact that our method relies only on unstructured point clouds, our approach results in more realistic joint shapes, see Fig. 20. We achieve this using a new non-uniform bone scaling model, inspired by related methods for non-uniform scaling of man-made objects [KSSCO08].

Another class of previous methods focuses on reconstruction of articulated models solely from input data, without using any template priors. This approach has been explored both for input data without explicit correspondences [PG08, CZ11] as well as for animated meshes [LD14]. Lu et al. [LDAS13] propose a system to reconstruct a high fidelity outer surface of human limbs and compute their volume, however, only for static shapes. In general, it is impossible to discover internal anatomical structures just from the surface information alone. Instead, our method takes advantage of anatomically-based models such as [HMD05].

### 3. Method

Our anatomically-based template model consists of: 1) the geometry of the bones and the skin, 2) bone kinematics, and 3) the skinning function. The role of the kinematic model is to parameterize a pose, i.e., motion due to skeletal articulation. We concatenate all of the pose degrees of freedom into a vector  $\theta$ . For example, our upper extremity model has 10 degrees of freedom (6 in the shoulder, 1 for elbow flexion, 1 for radius rotation, and 2 for the wrist). The skinning function maps the pose vector  $\theta$  to the shape of the deformed skin, as illustrated in Fig. 1(c,d,e). Our current skinning model does not contain dynamics or muscle contractions, even though these would be interesting additions in the future.

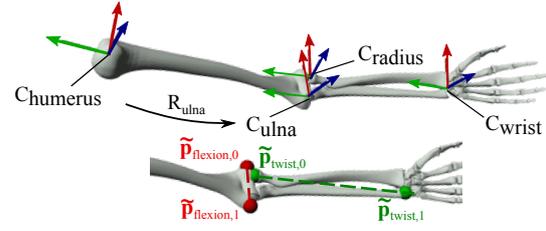


**Figure 4:** The template anatomy model (quantities denoted with a hat) is adapted to personalized anatomy model by anthropometric deformation. After adaptation, we obtain a rest pose (quantities denoted with a tilde) which is articulated using pose-based deformation.

The main feature of our anatomy model is its *adaptability*, i.e., the ability to reshape itself to match the anatomy of a given human subject. The adaptation consists in deforming the template anatomy. This is different than traditional pose-based skinning (Fig. 1(e)), because it accounts for changes in lengths and girths from one individual to another. To avoid confusion, we call this *anthropometric deformation* (Fig. 4). In the following section we discuss the details of our anatomy model; subsequently, in Sec. 3.2 we describe our method of adapting the anatomy model given a vector  $\mathbf{a}$  of anthropometric parameters. Sec. 3.3 presents our algorithm to optimize the parameters  $\mathbf{a}$  against a set of 3D point clouds, corresponding to surface measurements of a given human subject in several different poses. The bone motion reconstruction from captured point clouds is discussed in Sec. 3.4. In Sections 3.5 and 3.6, we discuss the details of our anatomically-based models of the upper and lower extremities, which fit well into our general framework of anatomy adaptation and bone motion reconstruction.

#### 3.1. Anatomy Model

Our anatomy model builds on top of traditional kinematic and skinning techniques. We found that the requirement of anatomical plausibility involves interesting challenges: for example, the forearm contains two major bones, not one, and



**Figure 5:** Kinematics of the upper limb (top). Below, we show landmarks defining the axes of rotation of the ulna (red) and the radius bones (green).

the knee joint does not only rotate, but also translates by non-negligible amounts [LT08]. Our model must be able to adapt itself to different body shapes while remaining functional, i.e., well-articulated and skinned.

**Kinematics.** Bone kinematics is described using standard kinematic chains. Every bone  $b$  is associated with a coordinate frame  $C_b$  which specifies the pivot point and local coordinate system for expressing its motion. For example, the coordinate frames of our upper extremity model are illustrated in Fig. 5.

To enable subject-specific adaptability of our model, the coordinate frames are defined using a vector of landmark points  $\tilde{\mathbf{p}}$ . We use standard landmarks from the biomechanics literature [WSA\*02, WvdHV\*05]. Each axis of rotation is defined by two landmark points, typically on the bone head; the coordinate frame associated with this axis of rotation can be chosen arbitrarily. We denote this (nonlinear) relationship as  $C_b(\tilde{\mathbf{p}})$ . An interesting case is the coordinate frame of the radius, i.e.,  $C_{\text{radius}}(\tilde{\mathbf{p}})$ , because its axis of rotation specifies revolution of the radius around the ulna; this axis is defined by the radial head and the distal part (*styloid process*) of the ulna, see Fig. 5 (bottom). The interesting fact that this axis of rotation depends on the length of the forearm is automatically accounted for in our definition of  $C_{\text{radius}}(\tilde{\mathbf{p}})$ , it is not necessary to implement any special logic.

We denote the relative transformation between coordinate frames as  $R_b(\tilde{\mathbf{p}}) = C_b(\tilde{\mathbf{p}})C_{\text{parent}(b)}^{-1}(\tilde{\mathbf{p}}) \in SE(3)$ , e.g., see  $R_{\text{ulna}}$  in Fig. 5. We denote the current pose of bone  $b$  with respect to its parent as  $P_b(\theta, \tilde{\mathbf{p}}) \in SE(3)$ , where  $\theta$  is the vector of pose parameters. In typical skeleton models used in computer graphics,  $P_b$  would be a pure rotation and would only depend on joint angles. However, in our case we need to support more complicated anatomical joints such as the knee, which requires significant translations. The final coordinate frames are computed by concatenating the entire chain of transformations from the root, i.e.,  $F_b(\theta, \tilde{\mathbf{p}})$  can be written as:

$$P_{\text{root}}(\theta, \tilde{\mathbf{p}}) \cdots R_{\text{parent}(b)}(\tilde{\mathbf{p}})P_{\text{parent}(b)}(\theta, \tilde{\mathbf{p}})R_b(\tilde{\mathbf{p}})P_b(\theta, \tilde{\mathbf{p}}) \quad (1)$$

Note that all coordinate frames are parameterized by our

landmark points  $\tilde{\mathbf{p}}$ , which is the key to achieving subject-specific adaptation.

**Skinning.** Transformations from the rest pose of bone  $b$  to its articulated position are defined as  $T_b(\theta, \tilde{\mathbf{p}}) = F_b(\theta, \tilde{\mathbf{p}})F_b(\mathbf{0}, \tilde{\mathbf{p}})^{-1}$ , assuming the rest pose corresponds to a zero  $\theta$  vector. The resulting matrices  $T_b$  correspond to classical skinning transformations, and can be directly applied in linear blend skinning:

$$\mathbf{v}_i(\theta, \tilde{\mathbf{p}}) = \sum_b w_{b,i} T_b(\theta, \tilde{\mathbf{p}}) \tilde{\mathbf{v}}_i \quad (2)$$

where  $w_{b,i}$  denotes the skinning weight of bone  $b$  corresponding to vertex  $i$  and  $\tilde{\mathbf{v}}_i$  is the rest pose of vertex  $i$ . Unfortunately, linear blend skinning does not always lead to realistic skin shapes. Therefore, we generalize our skinning model by replacing the  $T_b(\theta, \tilde{\mathbf{p}}) \tilde{\mathbf{v}}_i$  term with a more general *deformer* function,  $D(\theta, \tilde{\mathbf{p}}, \tilde{\mathbf{v}}_i)$ , leading to linear deformer blending:

$$\mathbf{v}_i(\theta, \tilde{\mathbf{p}}) = \sum_d w_{d,i} D_d(\theta, \tilde{\mathbf{p}}, \tilde{\mathbf{v}}_i) \quad (3)$$

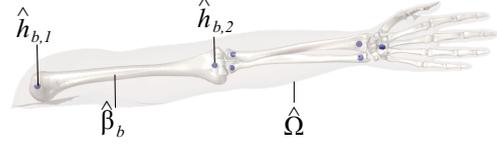
where  $d$  indexes individual deformers, and  $w_d$  corresponds to deformer blending weights. This formulation is more general, linear blend skinning can be easily implemented by using rigid body deformers:  $D_d(\theta, \tilde{\mathbf{p}}, \tilde{\mathbf{v}}_i) = T_d(\theta, \tilde{\mathbf{p}}) \tilde{\mathbf{v}}_i$ . However, more sophisticated deformers are necessary to achieve realistic results with anatomically-realistic bone kinematics, see Sec. 3.5.

### 3.2. Adaptation

Our skinning model discussed in the previous section describes deformations due to articulated motion of the skeleton. In this section, we discuss deformations due to body shape variations, e.g., making limbs longer or shorter. These anthropometric deformations affect all components of our anatomy model: geometry, kinematics, and skinning. In our system, the anthropometric deformations are applied first, before pose-based deformations (skinning). We call the initial anatomy model, i.e., before the application of anthropometric deformations, the “template anatomy” and we denote its quantities with a hat. For example, the skin vertices in our anatomy template are represented by vector  $\hat{\mathbf{v}}$ . After anthropometric deformation, the model is in its rest pose, ready to be deformed with pose-based deformation. The rest pose quantities are denoted with a tilde, such as the  $\tilde{\mathbf{v}}$  and  $\tilde{\mathbf{p}}$ ; see Fig. 4 for a visual overview.

We start from a template anatomy model corresponding to an average male. We tetrahedralize its bone shapes  $\hat{\beta}_b$  and the entire domain  $\hat{\Omega}$ , see Fig. 6. Our template adaptation proceeds in two phases. In the first phase, we reshape the bones and the surrounding soft tissues as a whole, i.e., assuming that the shapes of the skin and the bones are perfectly correlated. In the second phase, we assume the shape of the bones

is final and we update only the shape of the skin, accounting for individual variations in muscle mass and fat tissues.



**Figure 6:** Anthropometric deformation handles of our upper limb model.

In the **first phase**, we use the following deformation model. We define two *anthropometric deformation handles*,  $\hat{h}_{b,1}$  and  $\hat{h}_{b,2}$  for each bone  $b$ , corresponding to its proximal and distal heads, see Fig. 6. This deformation model is inspired by growth mechanisms of human bones [KL85]. In particular, bone heads (epiphyses) do not differ much between individuals, except for uniform scaling. Bone length is determined by the length of the bone shaft (diaphysis). Therefore, we deform each bone as follows. We pre-compute point-based bounded biharmonic weights [JBPS11] for each of the two handles in  $\hat{\beta}_b$ , and we deform the bone by applying two affine transformations at  $\hat{h}_{b,1}$  and  $\hat{h}_{b,2}$ . We allow only translation and uniform scaling, because rotational motion is accounted for in pose-based deformation (Sec. 3.1). This way, we ensure that the bone heads maintain their shape, while allowing the bones to change their length. The deformations applied to individual bones are then naturally extended to the entire shape  $\hat{\Omega}$ , by using pre-computed bounded biharmonic weights for the entire domain  $\hat{\Omega}$ .

We denote the vector of parameters of our handle transformations as  $\mathbf{a}$  (the *anthropometric* parameters). Specifically, the vector  $\mathbf{a}$  stacks the uniform scale coefficients and the 3D translation vectors for all handles  $\hat{h}_{b,j}$ . However, we need to impose certain constraints on  $\mathbf{a}$  in order to ensure anatomically-realistic results. In particular, a joint can be only scaled as a whole, i.e., all bone heads meeting at a given joint must be scaled by the same amount. For example, for the elbow joint, we have the following constraints:  $s_{\text{humerus},2} = s_{\text{ulna},1} = s_{\text{radius},1}$ , where  $s > 0$  denotes the scale factor. Similarly, every joint can be only translated as a whole. Incompatible translations would correspond to joint dislocation, which we do not model. For example, for the elbow joint we have the following constraint:

$$(s - 1)(\hat{\mathbf{h}}_{\text{ulna},1} - \hat{\mathbf{h}}_{\text{humerus},2}) = \hat{\mathbf{t}}_{\text{ulna},1} - \hat{\mathbf{t}}_{\text{humerus},2} \quad (4)$$

where  $s = s_{\text{humerus},2} = s_{\text{ulna},1}$ ,  $\hat{\mathbf{h}}$  represents the (constant) 3D locations of the handles and  $\hat{\mathbf{t}}$  are their translations. This formula accounts for the interplay between scaling and translation; an analogous constraint also applies for the radius. We put together all of these equality constraints, obtaining an affine set  $\mathcal{A}$ , because each individual constraint is affine. In Sec. 3.3, we explain how to find the individual-specific parameters  $\mathbf{a}$  subject to the constraint  $\mathbf{a} \in \mathcal{A}$ . Note that the

constraints  $\mathcal{A}$  are defined only by general anatomical considerations and therefore apply equally to all human subjects.

To deform our anatomy template for a given vector of anthropometric parameters  $\mathbf{a}$ , we apply bounded biharmonic weights to interpolate the transformations of individual handles. Because this interpolation is linear, the rest pose skin vertices  $\hat{\mathbf{v}}$  are an affine function of  $\mathbf{a}$  and  $\hat{\mathbf{v}}$ ; to keep the notation simple, we write this affine dependence as  $\tilde{\mathbf{v}}(\mathbf{a}, \hat{\mathbf{v}})$ . The rest-pose landmark points  $\tilde{\mathbf{p}}$  are obtained in a similar way, using an affine function  $\tilde{\mathbf{p}}(\mathbf{a})$ . Note that here we do not need the second argument because our template-pose landmarks are assumed to be constant.

In the **second phase**, we adapt the template vertices  $\hat{\mathbf{v}}$  in order to account for subject-specific soft tissue shape variations. From a modeling perspective, this is straightforward: we directly move  $\hat{\mathbf{v}}$  to the desired locations, as dictated by the input data. However, care must be taken to minimize the effect of noise, which we elaborate on in Sec. 3.3.

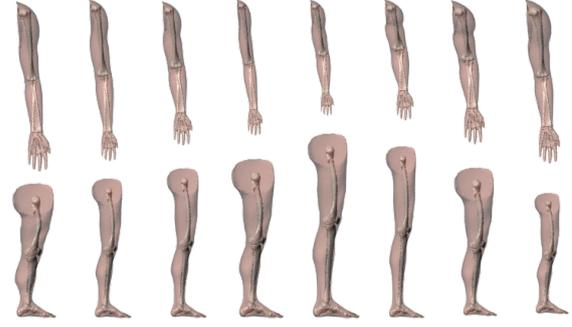
Our complete deformation model, accounting for both anthropometric and pose-based deformations, is obtained by substituting the anthropometric deformation into Eq. (3). This leads to:

$$\mathbf{v}_i(\theta, \mathbf{a}, \hat{\mathbf{v}}_i) = \sum_d w_{d,i} D_d(\theta, \tilde{\mathbf{p}}(\mathbf{a}), \tilde{\mathbf{v}}_i(\mathbf{a}, \hat{\mathbf{v}}_i)) \quad (5)$$

This formula looks simple but encompasses a lot of functionality. For example, changing the anthropometric parameters  $\mathbf{a}$  results in a change of  $\tilde{\mathbf{p}}$  which in turn changes the joint coordinate frames (Fig. 5), resulting in a change of the kinematic chain and, consequently, changes in the deformer matrices. See Fig. 7 for a visual demonstration of the adaptability of our model, noting that all of the models are functional both in terms of kinematics and skinning. Animated examples can be seen in the accompanying video. In Sec. 3.3, we will need the partial derivatives of  $\mathbf{v}_i(\theta, \mathbf{a}, \hat{\mathbf{v}}_i)$ . As an implementation note, we recommend composing Eq. (5) from elementary building blocks, allowing for unit testing. For each of these elementary functions, we implement both the function itself and its partial derivatives; the partial derivatives are composed according to the chain rule.

### 3.3. Data-driven Anatomy Fitting

In this section we describe our algorithm to optimize the parameters of our model for given input data. The input data consists of 3D point clouds corresponding to a given human subject in various poses. The input point clouds are extracted from depth images captured by two PrimeSense Carmine 1.09 sensors. We use OpenNI2 to convert depth images to point clouds. We manually build a bounding box to cull the human subject from the background. In order to register the points from two depth sensors, we use a calibration box with known dimensions. Each capture session starts by capturing the calibration box by both of our depth sensors. The relative



**Figure 7:** Synthesized limb models from our adaptable anatomy model.

rigid body transformation between the two depth sensors is computed by registering two point clouds to a common box with the same size as the calibration box. We denote the final point clouds corresponding to the human performance as  $\mathcal{P}_k$ , where  $k$  indexes individual frames of our input animation. Every pose corresponds to different pose parameters  $\theta_k$ , which we need to calculate. At the same time, we also need to compute the anthropometric deformation parameters  $\mathbf{a}$  and the template skin vertex positions  $\hat{\mathbf{v}}$ . The parameters  $\mathbf{a}$  and  $\hat{\mathbf{v}}$  are the same for all poses, i.e., they depend only on the human body shape but not its current pose.

Our algorithm starts by optimizing the pose parameters  $(\theta_1, \theta_2, \dots)$  and the anthropometric parameters  $\mathbf{a}$ . Upon convergence, we optimize the template vertices  $\hat{\mathbf{v}}$ . The idea is that we first deform the bones and the skin simultaneously, i.e., as one volumetric ensemble. When no further progress can be made, we accept the current shape of the bones and optimize only for  $\hat{\mathbf{v}}$ , accounting for subject-specific differences in soft tissues.

One complication is that we do not have any *a priori* correspondences between the skin vertices and the point clouds  $\mathcal{P}_k$ ; in fact, every  $\mathcal{P}_k$  represents a completely different set of points. Therefore, we proceed as in non-rigid iterative closest point algorithms, with Eq. (5) serving as the underlying deformation model. In particular, we implement a routine `find_corres(v, P_k)` which finds correspondences between mesh vertices  $\mathbf{v}$  and the point cloud  $\mathcal{P}_k$ , similar to [LAGP09] [CZ11]. This routine returns a selector matrix  $\mathbf{S}_k$  and a vector of target values  $\mathbf{t}_k$ , such that  $\|\mathbf{S}_k \mathbf{v} - \mathbf{t}_k\|^2$  is a data term attracting selected vertices from  $\mathbf{v}$  to their desired locations in  $\mathcal{P}_k$ . Note that this data term allows us to weigh each vertex by an estimate of correspondence “trustworthiness”, or even drop a vertex entirely if no suitable correspondence has been found. Specifically, we discard correspondences which are further than 4cm away, have negative dot product of their estimated normals, or lie on the scan boundary. The rows of  $\mathbf{S}_k$  and  $\mathbf{t}_k$  can encode either point-to-point or point-to-plane distances. We use both, with point-to-plane

weight  $10\times$  stronger than point-to-point, as suggested by Li and colleagues [LAGP09]. This allows us to compensate for the fact that our point cloud  $\mathcal{P}_k$  is only a sparse sampling of a smooth surface.

---

**Algorithm 1:**  $\text{optimize\_}\theta((\theta_1, \theta_2, \dots)_{\text{init}}, \mathbf{a}, \hat{\mathbf{v}})$

---

```

1 for each example  $k$  do
2    $\theta_k \leftarrow \theta_{k,\text{init}}$ 
3   repeat
4      $(\mathbf{S}_k, \mathbf{t}_k) \leftarrow \text{find\_corres}(\mathbf{v}(\theta_k, \mathbf{a}, \hat{\mathbf{v}}), \mathcal{P}_k)$ 
5      $\theta_k \leftarrow \arg \min_{\mathbf{x}} \|\mathbf{S}_k \mathbf{v}(\mathbf{x}, \mathbf{a}, \hat{\mathbf{v}}) - \mathbf{t}_k\|^2$ 
6   until convergence
7 end
```

---



---

**Algorithm 2:**  $\text{optimize\_}\mathbf{a}((\theta_1, \theta_2, \dots), \mathbf{a}_{\text{init}}, \hat{\mathbf{v}})$

---

```

1  $\mathbf{a} \leftarrow \mathbf{a}_{\text{init}}$ 
2 repeat
3   for each example  $k$  do
4      $(\mathbf{S}_k, \mathbf{t}_k) \leftarrow \text{find\_corres}(\mathbf{v}(\theta_k, \mathbf{a}, \hat{\mathbf{v}}), \mathcal{P}_k)$ 
5   end
6    $\mathbf{a} \leftarrow \arg \min_{\mathbf{x}} \sum_k \|\mathbf{S}_k \mathbf{v}(\theta_k, \mathbf{x}, \hat{\mathbf{v}}) - \mathbf{t}_k\|^2$  subj. to  $\mathbf{x} \in \mathcal{A}$ 
7 until convergence
```

---



---

**Algorithm 3:**  $\text{optimize\_}\hat{\mathbf{v}}((\theta_1, \theta_2, \dots), \mathbf{a}, \hat{\mathbf{v}}_{\text{init}})$

---

```

1  $\hat{\mathbf{v}} \leftarrow \hat{\mathbf{v}}_{\text{init}}$ 
2 repeat
3   for each example  $k$  do
4      $(\mathbf{S}_k, \mathbf{t}_k) \leftarrow \text{find\_corres}(\mathbf{v}(\theta_k, \mathbf{a}, \hat{\mathbf{v}}), \mathcal{P}_k)$ 
5   end
6    $\hat{\mathbf{v}} \leftarrow \arg \min_{\mathbf{x}} \mathbf{x}^T \mathbf{L} \mathbf{x} + \sum_k \|\mathbf{S}_k \mathbf{v}(\theta_k, \mathbf{a}, \mathbf{x}) - \mathbf{t}_k\|^2$ 
7   /*  $\mathbf{L}$  is a regularization term */
8 until convergence
```

---

Our individual optimization routines are summarized in Algorithms 1, 2, and 3; the final template adaptation process can be found in Algorithm 4. Currently, we rely on manual initialization of  $\mathbf{a}_{\text{init}}$  and  $\theta_{k,\text{init}}$  parameters;  $\hat{\mathbf{v}}_{\text{init}}$  is given by our template geometry. In Algorithms 1 and 2, we solve the nonlinear least squares problems using the Levenberg-Marquardt algorithm, which converges very well in our settings – typically in no more than 5 iterations. Note that aside from  $\mathcal{A}$  in Algorithm 2 we do not use any other constraints. The constraints  $\mathcal{A}$  are only equality constraints which can be implemented easily. In Algorithm 3, the regularization matrix  $\mathbf{L}$  is a standard cotangent Laplacian, used to suppress the effect of noise in the input data and regularize skin vertices that do not have any corresponding points in the input data. We observe that the function  $\mathbf{v}$  is linear with respect to its last argument and therefore the optimization problem in line (6) of Algorithm 3 reduces to solving a single linear system, because  $S_k$  is constant.

It would be straightforward to use all of our captured frames as input to our final template adaptation process (Algo-

---

**Algorithm 4:**  $\text{template\_adaptation}(\mathbf{a}_{\text{init}}, (\theta_1, \theta_2, \dots)_{\text{init}}, \hat{\mathbf{v}}_{\text{init}})$

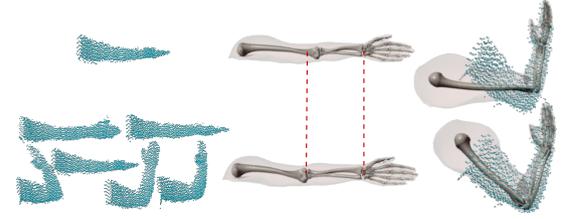
---

```

1  $(\theta_1, \theta_2, \dots) \leftarrow (\theta_1, \theta_2, \dots)_{\text{init}}, \mathbf{a} \leftarrow \mathbf{a}_{\text{init}}, \hat{\mathbf{v}} \leftarrow \hat{\mathbf{v}}_{\text{init}}$ 
2 repeat
3   repeat
4      $(\theta_1, \theta_2, \dots) \leftarrow \text{optimize\_}\theta((\theta_1, \theta_2, \dots), \mathbf{a}, \hat{\mathbf{v}})$ 
5      $\mathbf{a} \leftarrow \text{optimize\_}\mathbf{a}((\theta_1, \theta_2, \dots), \mathbf{a}, \hat{\mathbf{v}})$ 
6   until convergence
7    $\hat{\mathbf{v}} \leftarrow \text{optimize\_}\hat{\mathbf{v}}((\theta_1, \theta_2, \dots), \mathbf{a}, \hat{\mathbf{v}})$ 
8 until convergence
```

---

gorithm 4). However, as an optimization, we found that a much smaller number of representative poses ranging from 4 to 8 is typically sufficient (as shown in Fig. 1(b)). We pick these representative poses manually. We experimented with different strategies how to choose the representative poses, but we found that the results are not particularly sensitive to this choice as long as all important bone landmarks appear in at least one of the poses. For example, it is very difficult to find the exact location of the elbow from the straight arm, see Fig. 8. When this basic principle is taken into account, we observed that more example poses produce little improvement and only slow down the computation.



**Figure 8:** Using only one pose corresponding to straight arm, our template adaptation may produce inaccurate joint location estimates (top); increasing the number of input poses solves this problem (bottom). From left to right: input poses, adapted anatomy models and reconstructed bone motions for arm flexion. The top model fails in tracking.

### 3.4. Bone Motion Reconstruction

After our anatomy model has been adapted to a given human subject, we use it to reconstruct the corresponding bone motions. The input is a sequence of point clouds  $\{\mathcal{P}^f\}_{f=1,2,\dots}$  and the output is a sequence of pose parameters  $\{\theta^f\}_{f=1,2,\dots}$ . We use a small modification of Algorithm 1 to reconstruct bone motions from the input point clouds. In frame  $f$ , we use  $\theta^{f-1}$  as an initial guess; the first frame is initialized manually. Even with well-adapted models, Algorithm 1 may occasionally converge to the wrong local minimum and lose the tracking. This typically happens due to occlusions and insufficient resolution of the point clouds. In our current system we resolve these issues by user interaction, similar to the post-processing routinely performed with

professional motion capture systems such as Vicon. It would certainly be possible to apply more advanced tracking techniques, e.g., the probabilistic method combined with robust pose detection [WZC12]. However, we observed the tracking errors were typically infrequent and easily correctable by user interaction.

### 3.5. Upper Limb Model



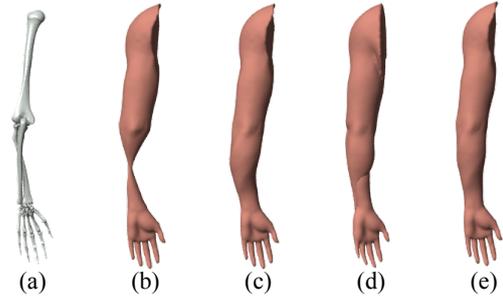
**Figure 9:** Bounded biharmonic weights (BBW) lead to undesirable influence regions. We redistribute their influence to match realistic deformations of the forearm.

The kinematics of our upper extremity model is based on the Stanford VA Upper Limb Model [HMD05]. We encountered certain interesting challenges when developing skinning compatible with this model. The first issue is relatively benign: bounded biharmonic weights [JBPS11] give us undesirable weights for the ulna and radius bones. We fix this by summing the radius and ulna weights together and modulating them with a linear ramp function along the length of the forearm, similar to stretchable-twistable bones [JS11], see Fig. 9.

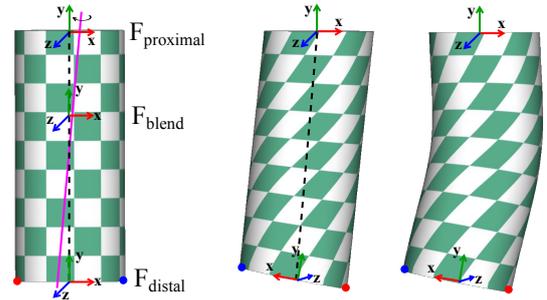
Unfortunately, even after correcting the weights, it is difficult to produce realistic deformation of the forearm. Linear blend skinning leads to a typical candy-wrapper artifact, see Fig. 10(b). Dual quaternion skinning removes the candy-wrapper, but unnaturally bends the forearm, see Fig. 10(c). The reason for the failure of dual quaternion skinning is the displacement of the wrist induced by the “carrying angle”, i.e., translation of the hand away from the body, allowing us to carry objects without colliding with the hips, see the inset figure to the right. Physics-based skinning using corotated elasticity [MZS\*11] leads to “pinching” of the soft tissue between the two bones (see Fig. 10(d)). This is because this model treats the body as a whole and does not account for sliding between the tendons and bones. Algorithms that take collision and contact into account exist [FLP14] but are very complex and would require the modeling of individual muscles. Another way to produce realistic forearm deformations would be using Pose Space Deformation [LCF00], however this would require additional data (corrective shapes).



We propose a light-weight deformer which leads to natural skin shapes. We start by defining two auxiliary coordinate



**Figure 10:** Challenges in skinning the forearm: pronation pose (a), linear blend skinning (b), dual quaternion skinning (c), corotated elasticity (d), our method (e).



**Figure 11:** Forearm deformer applied to a cylinder: rest-pose (left), 180° twist with our deformer (middle) and dual quaternion skinning (right). Note that the axis of rotation is not aligned with the axis of symmetry of the cylinder.

frames in the forearm,  $F_{\text{proximal}}$  and  $F_{\text{distal}}$ , see Fig. 11 (left). The proximal frame is parented to the ulna:  $F_{\text{proximal}}(\theta, \tilde{\mathbf{p}}) = F_{\text{ulna}}(\theta, \tilde{\mathbf{p}})R_{\text{proximal}}(\tilde{\mathbf{p}})$ , while the distal frame is parented to the radius:  $F_{\text{distal}}(\theta, \tilde{\mathbf{p}}) = F_{\text{radius}}(\theta, \tilde{\mathbf{p}})R_{\text{distal}}(\tilde{\mathbf{p}})$ . The key to realistic deformations is to combine spherical blending of the twist with linear shearing due to the carrying angle translation, see Fig. 11 (middle). We define a blended frame  $F_{\text{blend}}(\theta, \tilde{\mathbf{p}}, t)$  for  $t \in [0, 1]$ , interpolating between  $F_{\text{proximal}}(\theta, \tilde{\mathbf{p}})$  and  $F_{\text{distal}}(\theta, \tilde{\mathbf{p}})$  in the following way: the origins are blended linearly and the orientations (rotation matrices) are blended using SLERP. The forearm deformer then applies the transformation specified by  $F_{\text{blend}}$ :

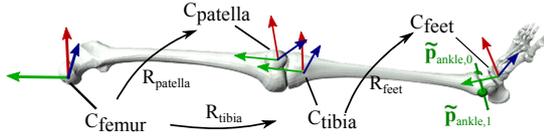
$$D_{\text{forearm}}(\theta, \tilde{\mathbf{p}}, \tilde{\mathbf{v}}_i) = F_{\text{blend}}(\theta, \tilde{\mathbf{p}}, t(\tilde{\mathbf{v}}_i)) F_{\text{blend}}(\mathbf{0}, \tilde{\mathbf{p}}, t(\tilde{\mathbf{v}}_i))^{-1} \tilde{\mathbf{v}}_i$$

where  $t(\tilde{\mathbf{v}}_i)$  assigns 1 to vertices near the elbow, 0 to vertices near the wrist, with linear transition in between. Note that while this deformer does not produce a smooth transition in the upper arm or the hand regions, smooth blending between individual body parts is achieved by smooth deformer blending weights  $w_d$  (see Eq. (3)). We believe that our  $D_{\text{forearm}}$  is the simplest deformer capable of achieving realistic forearm shapes, see Fig. 10(e).

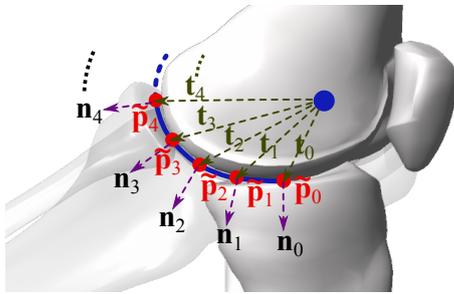
It is interesting to note that our method can still successfully track the pronation and supination motions even though the

forearm is mostly cylindrical and does not contain many visible features. This success stems from the fact that the radius bone drives the rotations of the hand, which helps our algorithm to disambiguate the orientation of the radius. This works reliably only with our proposed deformer; with dual quaternions we encountered hard-to-correct tracking problems due to the systematic error in the resulting skin shapes.

### 3.6. Lower Limb Model



**Figure 12:** Kinematics of the lower limb. We show the landmarks defining the axis of rotation of the foot (green).



**Figure 13:** The kinematics of the knee. The vectors  $\mathbf{n}_i$  are normals at landmark points  $\tilde{\mathbf{p}}_i$ .

The kinematics of the lower limb used in our system is shown in Fig. 12. Different from the upper limb, there is no twist between the tibia and fibula and therefore, both bones share one common transformation  $C_{\text{tibia}}$ . The ankle can be represented as a simple hinge joint, but the knee requires a more elaborate model. Because the head of the femur bone is not cylindrical, the knee joint introduces translations to the tibia. We use the standard spline model from OpenSIM [DAA\*07] to model the kinematics of the knee. In this section we explain how our adaptable anatomy models support these more complicated joint models. Let us focus on the kinematics of the femurotibial joint. The total transformation of the tibia  $P_{\text{tibia}}(\theta, \tilde{\mathbf{p}}) \in SE(3)$  consists of a translation  $\mathbf{t}(\theta)$  and rotation with angle  $\alpha(\theta)$ . Note that the knee flexion parameter in  $\theta$  is not the angle of rotation of the lower leg; the actual angle and the translation depend nonlinearly on  $\theta$ . This nonlinear relationship is approximated using a cubic spline. Specifically,  $\mathbf{t}(\theta) = \sum_i \mathbf{t}_i b_i(\theta)$ , where  $\mathbf{t}_i$  are vectors pointing from the center of rotation to landmark points  $\tilde{\mathbf{p}}_i$  (see Fig. 13);  $b_i(\theta)$  are standard cubic spline basis functions. Similarly, the angle  $\alpha(\theta) = \sum_i \alpha_i b_i(\theta)$ , where  $\alpha_i$  are the angles between normals  $\mathbf{n}_i$  and  $\mathbf{n}_0$  (see Fig. 13). The control

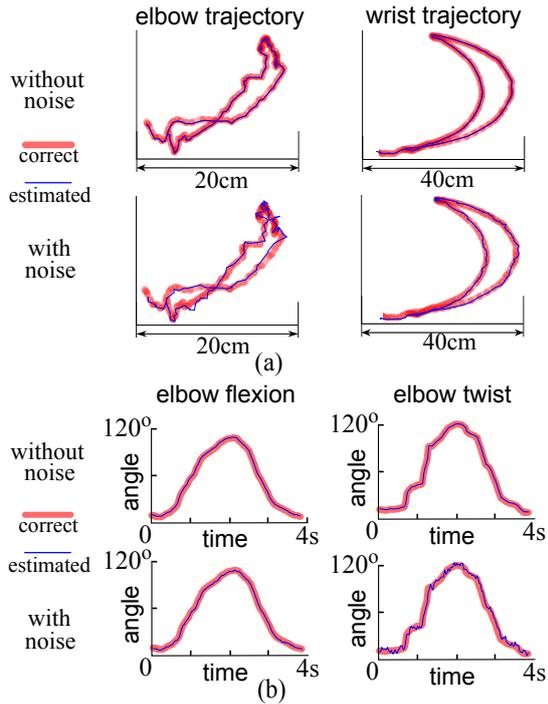
points  $\tilde{\mathbf{p}}_i$  of this spline are copying the shape of the joint and we include them in the vector of landmark points  $\tilde{\mathbf{p}}$ . This way, the adjustments of the spline due to anthropometric deformations are handled automatically in our framework: the spline control points are scaled just like any other landmarks, which consequently updates the knee kinematics. This fits our general framework beautifully, there is no need for any special logic.

## 4. Results

To prepare our template, we started with the bone and skin meshes of a 50th percentile male model, commercially available from the Zygote company. We took kinematic models from OpenSIM [DAA\*07] and adapted them to our bones. Subsequently, we calculated bounded biharmonic weights [JBPS11] to obtain the skinning weights of our template model. Model preparation took about 30 minutes for each limb.

Fig. 1(a) shows the setup of our data capture, which starts by converting two depth image sequences into animated point clouds. We use two PrimeSense Carmine 1.09 sensors and a single thread implementation of our algorithms running on a 2.5GHz processor. It takes around 1 minute to manually initialize the anthropometric parameters  $\mathbf{a}$  and 1 minute to initialize the pose parameters  $\theta_k$  for each pose, assuming 6 input poses. Our data-driven anatomy fitting (Algorithm 4) usually takes 2 to 3 iterations in the outer loop and 5 to 8 iterations in the inner loop to converge. In pose parameter optimization and anthropometric parameter optimization, vertex correspondences are dynamically updated. In Algorithms 1 and 2 we repeat correspondence finding and optimization for 10 to 15 iterations. For an upper limb model with 8175 skin vertices and 6 example poses, solving the nonlinear least squares optimization (line 5 of Algorithm 1) takes less than 0.7 seconds (per pose) and solving for the optimal anthropometric parameters (line 6 in Algorithm 2) takes less than 6 seconds (for all 6 poses simultaneously). The template vertex optimization (line 6 in Algorithm 3) costs around 0.5 seconds and is iterated with correspondences updated 3 times. Each example pose is represented as a point cloud with about 3000 to 4000 points. Our lower limb model has 8359 skin vertices. Pose parameter optimization exhibits performance similar to the upper limb case. For 6 input poses, each with about 4000 to 5000 points, each iteration of anthropometric parameter optimization and template vertex optimization takes around 2.8 seconds and 0.3 seconds, respectively. Overall, the entire adaptation process for upper and lower limbs takes around 10 to 30 minutes, depending on the accuracy of the initialization.

We first validate our algorithm with synthetic motions. In Fig. 14, we study a bend-and-twist motion of a manually adapted arm, created by a ground truth keyframed sequence, with point clouds artificially sampled on the animating skin



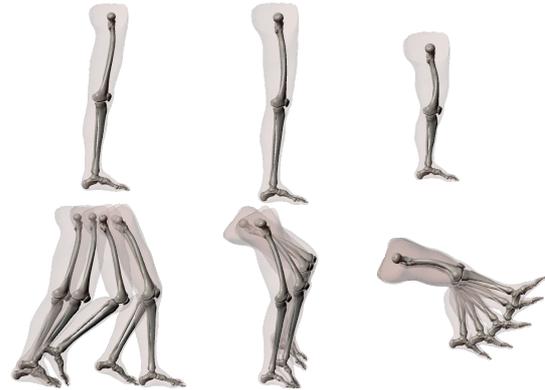
**Figure 14:** Results (thin blue curves) on synthetic arm bend-and-twist data without (upper row) and with (lower row) white noise, compared with the ground truth (thick red curves). (a) Trajectory of the elbow and wrist joints, (b) joint angles of elbow flexion and twist.

meshes. We add white noise to simulate the measurement inaccuracies. We select 6 point clouds as example poses for the template adaptation. We compare the locations of elbow and wrist to verify that bones are correctly adapted. For this comparison, we reconstruct bone motions and project the trajectory of elbow and wrist onto the frontal plane. The distance between the reconstructed joint trajectory and the ground truth is plotted in Fig. 14(a). Specifically, we plot the origin of  $C_{radius}$  as the elbow and the origin of  $C_{wrist}$  as wrist. We also graph the elbow flexion angle and elbow twist angle in Fig. 14(b) to show that our algorithm comes close to the ground truth with synthetic data.

Our experiments with real data involve three different subjects (an Indian female, an Asian male, and a Caucasian



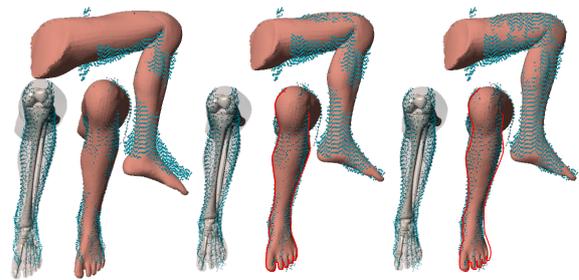
**Figure 15:** Adapted upper limb models (top) and reconstructed bone motions (bottom) for three different subjects.



**Figure 16:** Adapted lower limb models (top) and reconstructed bone motions (bottom) for three different subjects.

male). For each subject, we captured a range of motions of both the upper and lower limbs. The top rows of Fig. 15 and Fig. 16 display the resulting adapted anatomy models for each subject. Sample motions are shown in the bottom rows of Fig. 15 and Fig. 16; please see the accompanying video for more example motions. Due to the low resolution of our depth sensors, we cannot reliably capture the details of the hand. Therefore, we decided not to model the hand articulation, assuming the fingers are not moving significantly in our test motions.

Fig. 17 demonstrates how our adaptable anatomy model (left) fits to the input data. In the middle, we can see that the tibia is shortened to fit the captured data. In this phase (phase one), the skin is deformed in unison with the bones. To capture the individual differences in soft tissues, we subsequently optimize the template vertices (Algorithm 3). The result after this step (phase two) is shown in Fig. 17 (right).



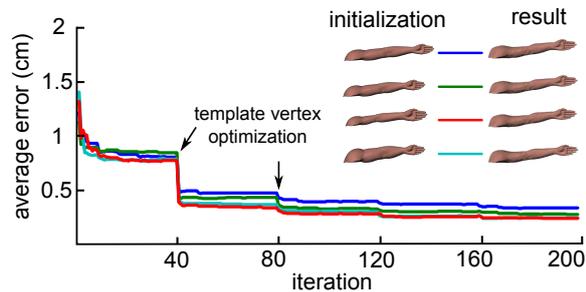
**Figure 17:** Template leg anatomy (left), the result of our adaptation algorithm before template vertex optimization (middle) and after template vertex optimization (right). We overlay the silhouette of the skin before template vertex optimization in the final result (right) to highlight the differences.

In Fig. 18 we study the convergence of the entire template adaptation algorithm (Algorithm 4). The graph plots an average distance between the target point clouds and our model

motions \ angles	elbow flexion		elbow twist	
	mean	SD	mean	SD
bend	3.306	1.677	10.114	5.470
twist	5.839	2.710	9.930	6.501
bend&twist	4.106	3.670	6.229	4.481

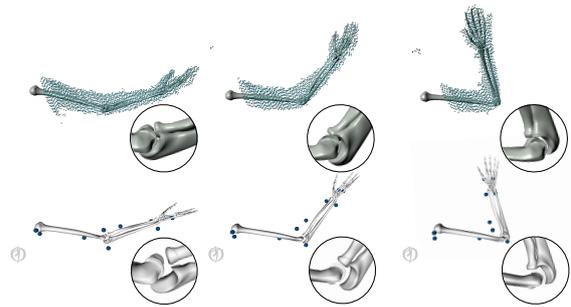
**Table 1:** Comparison of our system with OpenSIM + Vicon. For each joint angle and test motion we report the mean and standard deviation (SD) of the absolute value of the difference (in degrees).

during the iterative optimization. We examine four different starting points. Even though each of them converges to a slightly different local minimum due to the iterative correspondence updates, the final results are similar in shape.



**Figure 18:** Convergence of our data-driven anatomy fitting model with different initializations. Five iterations in the outer loops of Algorithm 4 are taken in this experiment. The sudden drops of the error are due to template vertex optimization (Algorithm 3).

An ideal way to validate our system would be using medical imaging, such as X-ray fluoroscopy. Unfortunately, the use of such devices is problematic and hard to justify with healthy human subjects. A radiation-free alternative is MRI, but these machines are very expensive and offer only a limited workspace. As a more viable alternative, we compare against traditional motion capture workflows used in biomechanics. In particular, we use the open source OpenSIM platform [DAA\*07]. While the main strength of OpenSIM is dynamic simulation, it also implements state-of-the-art inverse kinematics which uses pre-recorded trajectories of motion capture markers. While it would be ideal to run OpenSIM directly on our point clouds, OpenSIM does not provide any skinning models or correspondence finding algorithms. Therefore, we capture the same performance simultaneously with a 12-camera Vicon and our depth sensors. This puts our method at a slight disadvantage, because the Vicon markers introduce systematic errors in our point clouds – visible small balls corresponding to the markers. We did not attempt to remove these points. We process the marker trajectories captured by Vicon using the OpenSIM inverse kinematics module. In Fig. 19, we compare the results of our system

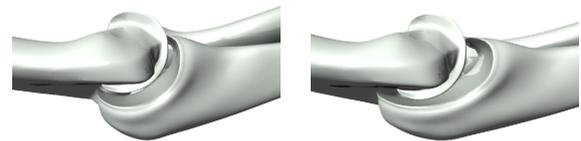


**Figure 19:** A bend-and-twist motion reconstructed with our system (top) and OpenSIM (bottom).

and OpenSIM visually and, in Table 1, we measure the difference of two joint angles in three example motions.

The chief advantage of our method is that our model better fits the human subject and models not only bone kinematics but also skin deformations. In contrast, OpenSIM uses only axis-aligned scaling of the bones and assumes the Vicon markers are rigidly attached to the bones. The higher fidelity of our joints is visible in the close up views in Fig. 19.

In Fig. 20 we compare our results to Anatomy Transfer [DLG\*13], which assumes a clean target mesh with perfect correspondences as input. The unnatural shapes of the bone heads are due to the fact that Anatomy Transfer applies one affine transformation to the entire bone. Our algorithm obtains a more natural joint structure as shown in Fig. 20 (left). Note that we achieve a better result in spite of the fact that our input data are noisy and do not come with user-specified correspondences.



**Figure 20:** Adapted bone heads with our method (left) and anatomy transfer [DLG\*13] (right).

## 5. Limitations and Future Work

A major topic of future work is to develop full-body adaptable anatomical models. While the kinematics of body parts such as the shoulder, pelvis, or spine is well studied [DAA\*07], developing efficient and anatomically-realistic skinning models for the corresponding body parts will be non-trivial. Our forearm deformer is the first step in this direction. More accurate skinning functions should also model effects such as muscle activations and dynamics: the effects of inertia will be particularly important in modeling passive organs such as fat tissues. Our current methodology may not

be adequate for highly muscular or obese subjects. One possibility to handle these cases would be using pose space deformation [LCF00].

Another interesting direction would be to consider pathologies such as joint sprain or dislocation, which may open up new and exciting opportunities to study, e.g., sports injuries. For example, it is conceivable that in the future, a fencing match would be recorded by multiple sensors to allow for after-the-fact analysis of the eventual mishaps. Our system assumes healthy human subjects, because our template adaptation mechanism is unable to account for anatomical abnormalities such as malformed bones; subtle abnormalities may be only detectable via medical imaging.

This paper focuses on adaptable anatomical models and does not aspire to compete with state-of-the-art motion capture solutions in terms of generality and robustness. The complex optimization routines and correspondence finding employed by our method currently hamper its use in real-time motion reconstruction. In the future, we are planning to incorporate automatic recovery from tracking failures. In our current system we do not model realistic articulation of hands and feet, because our sensors are not sufficiently accurate to resolve the finger motion. However, this might be possible with next generation depth sensors, perhaps already with Kinect for Xbox One.

## 6. Conclusions

We presented an adaptable anatomical model for realistic bone motion reconstruction. The main feature of our model is that all of its three components (geometry, kinematics, and skinning) remain functional and well-articulated after the model is adapted to different human subjects. We presented an algorithm that derives the anthropometric parameters of our model from a sequence of input point clouds, and we used the resulting personalized models to reconstruct bone motions of human subjects. In the near term, we hope that our work will help to promote anatomically-based modeling in computer graphics and visualization. In the long run, we foresee new technologies combining computer graphics, biomechanics, and orthopedic surgery.

## 7. Acknowledgements

We are indebted to Marianne Augustine, Norm Badler, Scott Delp, Bryan Dunham, James Gee, Alec Jacobson, Diego Jaramillo, Kang Li, Dimitris Metaxas, Benedict Brown, Aline Normoyle, James O'Brien, Saba Pascha, Mark Pauly, Eftychios Sifakis, Sofien Bouaziz, Leonid Sigal, Daniel Sykora, C.J. Taylor, Eo Trueblood, and Tomas Vondrak for numerous insightful discussions. Our special thanks belong to Spiro Metaxas, Vimanyu Jain, Nathan Marshak, and Sibi Vijayakumar for help with acquisition, data processing, and video editing. Aline Normoyle and Brittany Binler helped to

narrate the accompanying video. This research has been supported by NSF CAREER Award IIS-1350330. Xiaoyan Hu has been partially supported by NSFC (No.61103086 and 61170186).

## References

- [ASK\*05] ANGUELOV D., SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: Scape: shape completion and animation of people. In *ACM Trans. Graph.* (2005). 3
- [BTG\*12] BALLAN L., TANEJA A., GALL J., VAN GOOL L., POLLEFEYS M.: Motion capture of hands in action using discriminative salient points. In *ECCV*. Springer, 2012. 3
- [CGMA10] CORAZZA S., GAMBARETTO E., MÜNDERMANN L., ANDRIACCHI T. P.: Automatic generation of a subject-specific model for accurate markerless motion capture and biomechanical applications. *IEEE Transactions on Biomedical Engineering* 57, 4 (2010), 806–812. 3
- [CJ10] CLUTTERBUCK S., JACOBS J.: A physically based approach to virtual character deformation. In *ACM SIGGRAPH 2010 talks* (2010). 2
- [CMC\*06] CORAZZA S., MÜNDERMANN L., CHAUDHARI A., DEMATTIO T., COBELLI C., ANDRIACCHI T.: A markerless motion capture system to study musculoskeletal biomechanics: Visual hull and simulated annealing approach. *Annals of biomedical engineering* 34, 6 (2006), 1019–1029. 3
- [CMG\*10] CORAZZA S., MÜNDERMANN L., GAMBARETTO E., FERRIGNO G., ANDRIACCHI T. P.: Markerless motion capture through visual hull, articulated icp and subject specific model generation. *IJCV* 87, 1-2 (2010), 156–169. 3
- [CSC14] CESERACCIU E., SAWACHA Z., COBELLI C.: Comparison of markerless and marker-based motion capture technologies through simultaneous data collection during gait: Proof of concept. *PLoS one* 9, 3 (2014), e87640. 3
- [CZ11] CHANG W., ZWICKER M.: Global registration of dynamic range scans for articulated model reconstruction. *ACM Trans. Graph.* 30, 3 (2011), 26. 3, 6
- [DAA\*07] DELP S. L., ANDERSON F. C., ARNOLD A. S., LOAN P., HABIB A., JOHN C. T., GUENDELMAN E., THELEN D. G.: Opensim: open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering* 54, 11 (2007), 1940–1950. 2, 3, 9, 11
- [DAST\*08] DE AGUIAR E., STOLL C., THEOBALT C., AHMED N., SEIDEL H.-P., THRUN S.: Performance capture from sparse multi-view video. In *ACM Trans. Graph.* (2008), vol. 27. 3
- [DGFP12] DICKO A., GILLES B., FAURE F., PALOMBI O.: From generic to specific musculoskeletal simulations using an ontology-based modeling pipeline. In *Intelligent Computer Graphics 2012*. Springer, 2012, pp. 227–242. 3
- [DLG\*13] DICKO A.-H., LIU T., GILLES B., KAVAN L., FAURE F., PALOMBI O., CANI M.-P.: Anatomy transfer. *ACM Trans. Graph.* 32, 6 (2013). 2, 3, 11
- [DLH\*90] DELP S. L., LOAN J. P., HOY M. G., ZAJAC F. E., TOPP E. L., ROSEN J. M.: An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Transactions on Biomedical Engineering* 37, 8 (1990). 3
- [FLP14] FAN Y., LITVEN J., PAI D. K.: Active volumetric musculoskeletal systems. *ACM Trans. Graph.* 33, 4 (2014). 2, 8
- [GKM\*07] GARLING E., KAPTEIN B., MERTENS B., BARENDREGT W., VEEGER H., NELISSEN R., VALSTAR E.: Soft-tissue artefact assessment during step-up using fluoroscopy and

- skin-mounted markers. *Journal of Biomechanics* 40 (2007), S18–S24. 3
- [GRP10] GILLES B., REVERET L., PAI D.: Creating and animating subject-specific anatomical models. In *Computer Graphics Forum* (2010), vol. 29, pp. 2340–2351. 3
- [GSDA\*09] GALL J., STOLL C., DE AGUIAR E., THEOBALT C., ROSENHAHN B., SEIDEL H.-P.: Motion capture using joint skeleton tracking and surface estimation. In *CVPR* (2009). 3
- [HBB\*13] HELTEN T., BAAK A., BHARAJ G., MULLER M., SEIDEL H.-P., THEOBALT C.: Personalization and evaluation of a real-time depth-based full body tracker. In *3DTV-Conference, 2013* (2013), IEEE, pp. 279–286. 3
- [HMD05] HOLZBAUR K. R., MURRAY W. M., DELP S. L.: A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control. *Annals of biomedical engineering* 33, 6 (2005), 829–840. 3, 8
- [HSS\*09] HASLER N., STOLL C., SUNKEL M., ROSENHAHN B., SEIDEL H.-P.: A statistical model of human pose and body shape. In *Computer Graphics Forum* (2009), vol. 28. 3
- [JBPS11] JACOBSON A., BARAN I., POPOVIĆ J., SORKINE O.: Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.* 30, 4 (2011), 78. 5, 8, 9
- [JS11] JACOBSON A., SORKINE O.: Stretchable and twistable bones for skeletal shape deformation. *ACM Trans. Graph.* 30, 6 (2011), 165:1–165:8. 8
- [KL85] KUMMER B., LOHSCHIEDT K.: Mathematical model of the longitudinal growth of long bones. *Anatomischer Anzeiger* 158, 5 (1985), 377. 5
- [KS12] KAVAN L., SORKINE O.: Elasticity-inspired deformers for character articulation. *ACM Trans. Graph.* 31, 6 (2012). 3
- [KSSCO08] KRAEVOY V., SHEFFER A., SHAMIR A., COHEN-OR D.: Non-homogeneous resizing of complex models. *ACM Trans. Graph.* 27, 5 (2008), 111:1–111:9. 3
- [LAGP09] LI H., ADAMS B., GUIBAS L. J., PAULY M.: Robust single-view geometry and motion reconstruction. *ACM Trans. Graph.* 28, 5 (2009), 175. 6, 7
- [LCF00] LEWIS J. P., CORDNER M., FONG N.: Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proc. ACM SIGGRAPH* (2000), pp. 165–172. 8, 12
- [LD14] LE B. H., DENG Z.: Robust and accurate skeletal rigging from mesh sequences. *ACM Trans. Graph.* 33, 4 (2014). 3
- [LDAS13] LU G., DESOUZA G., ARMER J., SHYU. B. A. C.-R.: A system for limb-volume measurement using 3d models from an infrared depth sensor. In *IEEE Symposium Series on Computational Intelligence, Symposium on Computational Intelligence for Healthcare and e-Health (CICARE)* (2013). 3
- [LGS\*13] LIU Y., GALL J., STOLL C., DAI Q., SEIDEL H., THEOBALT C.: Markerless motion capture of multiple characters using multi-view image segmentation. *TPAMI* (2013). 3
- [LLV\*12] LI H., LUO L., VLASIC D., PEERS P., POPOVIĆ J., PAULY M., RUSINKIEWICZ S.: Temporally coherent completion of dynamic shapes. *ACM Trans. Graph.* 31, 1 (2012), 2. 3
- [LO99] LU T.-W., O’CONNOR J.: Bone position estimation from skin marker co-ordinates using global optimisation with joint constraints. *Journal of biomechanics* 32, 2 (1999), 129–134. 3
- [LT08] LEE S.-H., TERZOPOULOS D.: Spline joints for multi-body dynamics. In *ACM Trans. Graph.* (2008), vol. 27. 4
- [MHKS11] MOESLUND T., HILTON A., KRUGER V., SIGAL L.: *Visual analysis of humans: looking at people*. Springer, 2011. 3
- [MZS\*11] MCADAMS A., ZHU Y., SELLE A., EMPEY M., TAMSTORF R., TERAN J., SIFAKIS E.: Efficient elasticity for character skinning with contact and collisions. In *ACM Trans. Graph.* (2011), vol. 30, p. 37. 8
- [NJL\*07] NESTER C., JONES R., LIU A., HOWARD D., LUNDBERG A., ARNDT A., LUNDGREN P., STACOFF A., WOLF P.: Foot kinematics during walking measured using bone and surface mounted markers. *Journal of biomechanics* 40, 15 (2007), 3412–3423. 3
- [PG08] PEKELNY Y., GOTSMAN C.: Articulated object reconstruction and markerless motion capture from depth video. In *Computer Graphics Forum* (2008), vol. 27, pp. 399–408. 3
- [Pop07] POPPE R.: Vision-based human motion analysis: An overview. *Computer vision and image understanding* 108, 1 (2007). 3
- [SHRB12] STRAKA M., HAUSWIESNER S., RUETHER M., BISCHOF H.: Simultaneous shape and pose adaptation of articulated models using linear optimization. In *ECCV* (2012). 3
- [SSK\*13] SHOTTON J., SHARP T., KIPMAN A., FITZGIBBON A., FINOCCHIO M., BLAKE A., COOK M., MOORE R.: Real-time human pose recognition in parts from single depth images. *Communications of the ACM* 56, 1 (2013), 116–124. 3
- [Tod07] TODOROV E.: Probabilistic inference of multi-joint movements, skeletal parameters and marker attachments from diverse sensor data. *IEEE Trans. on Biomed. Eng.* (2007). 3
- [VBMP08] VLASIC D., BARAN I., MATUSIK W., POPOVIĆ J.: Articulated mesh animation from multi-view silhouettes. In *ACM Trans. Graph.* (2008), vol. 27, p. 97. 3
- [WSA\*02] WU G., SIEGLER S., ALLARD P., KIRTLEY C., LEARDINI A., ROSENBAUM D., WHITTLE M., DIMA D. D., CRISTOFOLINI L., WITTE H., SCHMID O., STOKES I.: Isb recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion part i: ankle, hip, and spine. *Journal of Biomechanics* 35, 4 (2002), 543 – 548. 4
- [WvdHV\*05] WU G., VAN DER HELM F. C., VEEGER H. D., MAKHSOUS M., ROY P. V., ANGLIN C., NAGELS J., KARDUNA A. R., MCQUADE K., WANG X., WERNER F. W., BUCHHOLZ B.: Isb recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion part ii: shoulder, elbow, wrist and hand. *Journal of Biomechanics* 38, 5 (2005), 981 – 992. 4
- [WZC12] WEI X., ZHANG P., CHAI J.: Accurate realtime full-body motion capture using a single depth camera. *ACM Trans. Graph.* 31, 6 (2012), 188. 3, 8
- [Ylh\*12] YE G., LIU Y., HASLER N., JI X., DAI Q., THEOBALT C.: Performance capture of interacting characters with handheld kinects. In *ECCV*. Springer, 2012, pp. 828–841. 3