Sampling-based Rig Conversion into Non-rigid Helper Bones

Tomohiko Mukai
Tokai University
tmki@acm.org

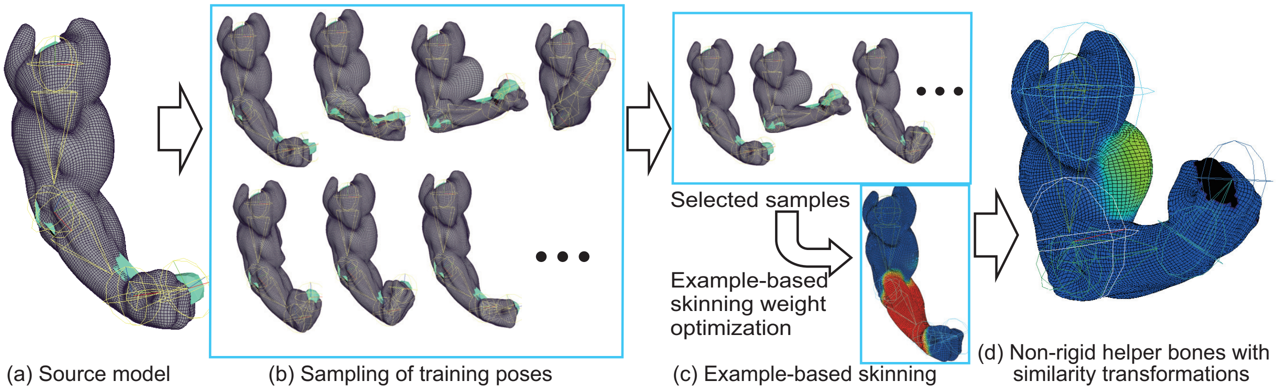

Selected samples
Example-based skinning weight optimization
(a) Source model
(b) Sampling of training poses
(c) Example-based skinning
(d) Non-rigid helper bones with similarity transformations

Figure 1: Our sampling-and-analysis strategy for converting an arbitrary skeleton-driven deformer into a linear blend skinning rig with helper bones. (a) Source model composed of virtual muscles. (b) Per-joint sampling for analyzing the influence of each joint movement on the skin deformation. (c) The target character rig is built using an example-based skinning technique with a minimum training dataset obtained by the importance sampling of joint rotations. (d) Non-rigid helper bones are added to compensate the conversion error from the source model using the skinning decomposition with similarity transformation algorithm.

ABSTRACT

While 3D animation packages provide a wide variety of animation rigs for creating expressive skin animation, most interactive systems employ linear blend skinning for hard realtime computation. We propose a method for converting an arbitrary skeleton-driven deformer into a linear blend skinning-based helper bone rig. Our system builds the target rig by applying an example-based skinning technique that uses a minimal training dataset obtained from the source model by two-pass sampling of the skin deformation. The first uniform sampling analyzes the relationship between the rotation of each joint and the deformation of skin vertices. The second sampling composes a minimum training dataset by selecting important pose samples using novel geometrical measures. We also propose a skinning decomposition with similarity transformation algorithm for accurately approximating the non-rigid skin deformation behavior by helper bone transformations. Our experimental results demonstrate the proposed automated rig conversion into non-rigid helper bones from several skeleton-driven deformers, including Delta Mush deformers, corrective blendshapes, and virtual-muscle systems.

CCS CONCEPTS

- **Computing methodologies → Animation**;

KEYWORDS

rig conversion, linear blend skinning, sampling, helper bone rig

ACM Reference Format:
Tomohiko Mukai. 2018. Sampling-based Rig Conversion into Non-rigid Helper Bones. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D'18)*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3203190

1 INTRODUCTION

Rigging is a core process in 3D animation production for making the skin deformation of characters believable. Artists use a wide variety of character rigs, such as linear blend skinning (LBS), cage-based deformers, blendshapes, and finite element simulations. These techniques are provided in digital content creation tools, which frequently require computational costs greater than that of the interactive rates for creating high quality skin deformations. In contrast, a hard realtime application always requires a lightweight rig because of the limited computational budget for the runtime animation synthesis. Hence, LBS is the de facto standard technique for interactive systems such as games and virtual reality systems because of its stable and efficient computation. An additional advantage of LBS-based methods is the high compatibility of LBS with standard graphics pipelines and game engines. This technique also has the capability to produce complex skin deformations by adding

I3D'18, May 2018, Montreal, Quebec, Canada

secondary rigs, such as helper bones and skeleton-driven corrective blendshapes.

Several techniques have been proposed for building an LBS-based rig that accurately approximates arbitrary types of skeleton-driven skin deformers. For instance, when a designer has set up a virtual muscle rig, the conversion mechanism generates a helper bone rig, requiring minimum manual editing. Most of these techniques take a sampling and analysis approach whereby a set of shapes is sampled from the source model for building an LBS-based rig using an example-based skinning method. This approach enables a deformer-independent rig conversion, since the sampling method assumes the source model as a blackbox system. However, existing methods frequently yield inaccurate results because of the naive sampling technique, such as uniform and random sampling.

In this paper, we propose a two-pass sampling approach for accurate rig conversion with a small number of important pose samples. Our method extends the previous example-based helper bone rigging [Mukai 2015] for dealing with a complex source model. Our primary contribution is the two-pass sampling strategy for generating a minimum training dataset for example-based skinning. Our method analyzes the effect of each joint transformation on the skin deformation using two types of geometrical measures for selecting important pose samples. The second contribution is an extension of the helper bone controller to approximate the nonlinear behavior of skin deformation. We propose a skinning decomposition with similarity transformation algorithm for estimating non-uniform scaling components of helper bones in addition to the rigid transformation. Our iterative algorithm guarantees stable and accurate approximation of non-rigid skin deformation such as muscle bulging.

Our summarized technical contributions are as follows.

- Two-pass sampling to generate a minimum training dataset for an efficient example-based rig conversion
- Geometrical criteria for quantifying the importance of training samples for reconstructing an LBS-based rig
- A skinning decomposition with similarity transformation algorithm for approximating stylized skin deformation by using non-rigid helper bones.

A major drawback of our approach is the strong assumptions made on the character model. Our method can take only skeleton-driven deformers as the source rig. The conversions between the different structures of skeleton hierarchy and between the different topologies of skin geometry are not incorporated into our methodology. Our method also does not produce dynamic skin deformation, such as jiggling of flesh. Despite these limitations, we believe that this technology provides a practical means for building a lightweight LBS-based rig for hard real-time applications that involves a minimum amount of manual labor.

2 RELATED WORK

Skeleton-driven deformers are frequently used tools for creating the skin animation of an articulated character. Pose space deformation is used for driving blendshapes with the skeleton pose [Lewis et al. 2000]. Delta Mush [Mancewicz et al. 2014] provides a detail-preserving skinning using multi-frequency representation of skin geometry. Regression techniques have been proposed to estimate the linear mapping from the skeletal pose to the additive vertex displacement [Kry et al. 2002] or to the deformation gradient of skin surface polygons [Wang et al. 2007]. Virtual muscle systems, such as Weta Digital's tissue system, provide high quality, physically-realistic skin deformation [Angelidis and Singh 2007; Saito et al. 2015].

Although these sophisticated methods provide high quality skin deformation, LBS remains the standard technique applied in many productions because of its efficient and stable runtime computation [Magnenat-Thalmann et al. 1988]. LBS is used to drive the skin deformation of a character according to its internal skeleton. Its variants, such as multi-weight enveloping [Wang and Phillips 2002], dual quaternion skinning [Kavan et al. 2007], and skinning with the optimal center of rotation estimation [Le and Hodgins 2016], also compute the skin deformation by blending the joint transformations with skinning weights. The setup of the LBS-based rig, however, requires a labor-intensive process for constructing the skeletal structure and skinning weight. In particular, LBS-based helper bone rigging [Mohr and Gleicher 2003; Parks 2005] frequently requires that skilled riggers use a trial and error process.

Automated rigging methods, such as Pinocchio [Baran and Popović 2007], an elasticity-inspired deformer [Kavan and Sorkine 2012], and skinning transformation techniques [Jacobson et al. 2012], can optimize the skinning weight by minimizing certain skin deformation energy without requiring example data. Several example-based methods have been proposed for automatically building an LBS-based skeleton rig. Skinning decomposition methods build a skeleton rig that approximates the shape examples using a non-negative least-square technique [James and Twigg 2005], subspace optimization [Kavan et al. 2010], and rigid transform approximation and quadratic programming [Le and Deng 2012, 2014], which is later applied for example-based rigging with rigid helper bones [Mukai 2015; Mukai and Kuriyama 2016]. The SMPL model [Loper et al. 2015] learns pose-dependent corrective blendshapes from the 4D-captured shape deformation sequence of the human body. An additional method constructs a physics-based muscle rig from the captured human skin deformation [Kadlecek et al. 2016]. Our method converts any type of skeleton-driven deformers into a helper bone rig using an example-based rigging technique via sampling a minimal training dataset built from the source model.

The rig transfer method is closely related to our method. The purpose of this technique is to transfer an animation rig into a different character having a different size. Rig transfer methods leverage a shape correspondence method to detect the location at which the skeleton joints should be embedded. For example, an LBS-based rig is transferred by determining the shape correspondence between character models having different shapes [Avril et al. 2016]. The rig transfer method for a muscle-based model [Ali-Hamadi et al. 2013; Seo et al. 2010] optimizes the embedding of virtual muscles using geometric correspondence information. The Frankenrigs method [Miller et al. 2011] constructs a whole-body rig by transferring partial rigs from multiple sources. In contrast, our method replaces skeleton-driven deformers with an LBS-based rig without changing the topology and size of either the skeleton hierarchy or the skin mesh geometry, except for adding extra bones. This basic idea was proposed in a previous article [Mukai 2015] and commercial tools, such as the Maya bake deformer tool, have been developed.

However, they use a naive sampling method that frequently causes inaccurate conversion when the source model exhibits non-linear deformation behavior such as muscle bulging.

3 PROBLEM FORMULATION

In this section, we formulate our rig conversion problem. We assume that the source and target characters have the same skin mesh surface geometry and the same hierarchical skeleton structure, which is composed of rigid bones and spherical joints. We also assume that the vertex normals are skinned using the same skinning weights and rotational component of the bone transformations. Given a set of indices of skeleton joints $\mathcal{J}$, a local rigid transformation of joint $j \in \mathcal{J}$ is represented by the combination of a logarithmic map of the joint rotation quaternion $\mathbf{r}_j$ and a constant translation $\bar{\mathbf{t}}_j$ from the parent joint. Let $\mathbf{v}_{i \in \mathcal{V}}$ be the position of the i-th skin vertex in the homogeneous coordinates, where $\mathcal{V}$ is a set of indices of the skin vertices. A character rig is represented as a set of per-vertex rig functions $\Upsilon_i(\mathbf{p}; \boldsymbol{\theta})$ that maps a skeleton pose $\mathbf{p} := (\mathbf{r}_j)_{j \in \mathcal{J}}$ to the i-th vertex position $\mathbf{v}_i$ with rig parameter $\boldsymbol{\theta}$ as $\Upsilon_i : \Re^{3 \cdot n(\mathcal{J})} \to \Re^3$, where $n(\cdot)$ denotes the number of elements of a set and the rig parameter $\boldsymbol{\theta}$ varies depending on the type of skin deformer.

Here, we define the function of helper bone rig Υ_i^{HB} as

$$\mathbf{v}_i = \Upsilon_i^{HB}\left(\mathbf{p}; \boldsymbol{\theta}^{HB}\right) := \sum_{j \in \mathcal{J}} w_{i,j} \mathbf{G}_{\psi(j)} \Lambda(\bar{\mathbf{t}}_j, \mathbf{r}_j) \bar{\mathbf{G}}_j^{-1} \bar{\mathbf{v}}_i + \sum_{h \in \mathcal{H}} w_{i,h} \mathbf{G}_{\psi(h)} \mathbf{M}_h(\mathbf{p}) \bar{\mathbf{G}}_h^{-1} \bar{\mathbf{v}}_i \ , \quad (1)$$

where $\mathbf{G}_j = \mathbf{G}_{\psi(j)} \Lambda(\bar{\mathbf{t}}_j, \mathbf{r}_j)$ represents the global transformation of the j-th joint, $\bar{\mathbf{G}}_j$ and $\bar{\mathbf{v}}_i$ are the global transformation and the vertex position, respectively, at the rest state, Λ is a function that composes a homogeneous transformation matrix, $\mathcal{H}$ denotes a set of indices of helper bones, and $\psi(j) \in \mathcal{J}$ and $\psi(h) \in \mathcal{J}$ denote the parent of the j-th primary joint and h-th helper bone, respectively. A local similarity transformation of the helper bone is procedurally controlled according to the primary skeleton pose as $\mathbf{M}_h(\mathbf{p}) : \Re^{3 \cdot n(\mathcal{J})} \to \Re^{4 \times 4}$. The skinning weights satisfy the non-negativity constraints $\forall i \in \mathcal{V}, \forall j \in \mathcal{J}, w_{i,j} \geq 0$ and $\forall i \in \mathcal{V}, \forall h \in \mathcal{H}, w_{i,h} \geq 0$, the affinity constraint $\forall i \in \mathcal{V}, \sum_{j \in \mathcal{J}} w_{i,j} + \sum_{h \in \mathcal{H}} w_{i,h} = 1$, and the sparsity constraint $\forall i \in \mathcal{V}, \sum_{j \in \mathcal{J}} |w_{i,j}|_0 + \sum_{h \in \mathcal{H}} |w_{i,h}|_0 \leq \chi$, where $| \cdot |_\alpha$ represents the L_α norm and χ denotes the maximum number of influencing joints for each vertex.

Our goal is to achieve automated construction of an LBS-based helper bone rig Υ^{HB} with optimal parameter $\boldsymbol{\theta}^{HB*} := \left(w_{i,j}, w_{i,h}, \mathbf{M}_h, \bar{\mathbf{G}}_h, \psi(h)\right)_{i \in \mathcal{V}, j \in \mathcal{J}, h \in \mathcal{H}}$ that approximates the skin deformation of the source rig Υ_i. We employ the sampling-based approach for analyzing the skin deformation behavior of the source model. Given a set of indices of skeleton pose samples Q, our sampling-based rig conversion is formulated as a least-square problem to minimize the shape differences:

$$\boldsymbol{\theta}^{HB*} = \underset{\boldsymbol{\theta}^{HB}}{\operatorname{argmin}} \sum_{q \in Q} \sum_{i \in \mathcal{V}} \left| \Upsilon_i\left(\mathbf{p}_q; \boldsymbol{\theta}\right) - \Upsilon_i^{HB}\left(\mathbf{p}_q; \boldsymbol{\theta}^{HB}\right) \right|_2^2 \ . \quad (2)$$

This least-square problem cannot be directly solved because of its high non-linearity. We therefore decompose the problem into two subproblems [Mukai 2015; Mukai and Kuriyama 2016]. The first subproblem is the optimization of the skinning weight of only primary joints while helper bones are neglected (§4). The second subproblem optimizes the skinning weights and transformation controllers of the helper bones to compensate the approximation error derived from the source rig (§5). The conversion process is summarized in Figure 1.

4 SAMPLING-BASED RIG CONVERSION

We first review the example-based skinning optimization technique and present our two-pass sampling method for generating a minimum training dataset from the source model.

4.1 Example-based Skinning Weight Optimization

The skinning weight optimization is formulated as a per-vertex constrained least-square problem to minimize the sum of the squared shape differences between the source and the target model over the entire training dataset as

$$(w_{i,j}^*)_{j \in \mathcal{J}} = \underset{(w_{i,j})_{j \in \mathcal{J}}}{\operatorname{argmin}} \sum_{q \in Q} \left| \Upsilon_i(\mathbf{p}_q) - \sum_{j \in \mathcal{J}} w_{i,j} \mathbf{G}_{\psi(j)} \Lambda(\bar{\mathbf{t}}_{j,q}, \mathbf{r}_{j,q}) \bar{\mathbf{G}}_j^{-1} \bar{\mathbf{v}}_i \right|_2^2 \quad (3)$$

$$\text{subject to} \ \ \forall i \in \mathcal{V}, \ \forall j \in \mathcal{J}, \ w_{i,j} \geq 0, \quad (4)$$

$$\forall i \in \mathcal{V}, \ \sum_{j \in \mathcal{J}} w_{i,j} = 1, \quad (5)$$

$$\forall i \in \mathcal{V}, \ \sum_{j \in \mathcal{J}} |w_{i,j}|_0 \leq \chi \ . \quad (6)$$

We use an approximation solution proposed in [Le and Deng 2012] to solve this problem. We first exclude the L_0-norm constraint (Equation 6) and solve the resulting quadratic programming problem. When the solution does not satisfy the L_0-norm constraint, the highest-effort χ joints are selected and the weights for other joints are set to zero. The final solution is obtained by solving the quadratic programming problem again with respect to the selected χ bones.

The remaining problem is the automated composition of the training dataset. Existing methods [Mukai 2015; Mukai and Kuriyama 2016] use a straightforward approach, such as uniform and random sampling. However, such a naive method decreases the conversion accuracy and is likely to lose the important characteristics, such as stylized deformation. Minimization of the datasize is also important for the interactive operation, since both the example-based skinning weight optimization and the helper bone rigging require a computation time that increases with the number of samples. Our observation is that the importance of a pose sample can be evaluated by analyzing the linear relationship between the joint rotation and vertex displacement, but it cannot be estimated prior to the sampling because of the blackbox function of the source rig. Therefore, the first pass generates many pose samples using uniform sampling to cover the entire range of joint motion, and in the second pass, a smaller number of important samples are selected based on the deformation influence analysis.

4.2 Per-joint Uniform Sampling

The first pass generates training samples by uniformly sampling the rotation of one joint while the remaining joints are maintained in the rest state. We use a uniform sampling technique for 3D spatial rotations [Yershova et al. 2010] in order to generate the rotation samples $\mathbf{r}_{j,s\in\mathcal{S}}$ of each joint, where $\mathcal{S}$ is the set of indices of the rotation samples. The valid samples are then selected within the range of joint rotation, which then compose a set of sample indices $\mathcal{R}_j$. Consequently, the first sampling generates $\sum_{j\in\mathcal{J}} n(\mathcal{R}_j)$ pose samples in total.

Originally, different combinations of rotations among multiple joints had to be sampled to approximate the high-dimensional, non-linear function of the source rig. However, this leads to the so-called curse of dimensionality problem, which causes an exponential increase in the number of pose samples according to the number of joints. Our basic insight is that there is a sparse relation between the skin vertex and joints, as each skin vertex is driven by only a smaller number of adjacent joints. Moreover, the basic assumption of LBS is that each vertex displacement is determined by a linear combination of independent joint transformations. In other words, we can neglect the complexed effect of the multiple joints on the skin deformation when we assume a linear relationship between the joint rotation and vertex displacement. Consequently, we take the per-joint sampling approach for reducing the number of training samples by necessarily tolerating the minimal degradation of conversion accuracy.

4.3 Selection Criteria

The second sampling pass selects the most important samples to reduce the number of training samples. We define two types of geometrical measures, called the displacement measure and dependency measure, as the selection criteria. The displacement measure $\mathrm{disp}_i(\mathbf{r}_j)$ quantifies the size of the displacement that the j-th joint's rotation $\mathbf{r}_j$ yields on the i-th vertex, and the dependency measure $\mathrm{dep}_i(\mathbf{r}_j)$ quantifies the influence of the vertex displacements that are obtained between the source rig and the LBS result, as illustrated in Figure 2(a). Let $\Upsilon_i(\mathbf{r}_j)$ be a source rig function that takes only $\mathbf{r}_j$ and maintains the other joints in the rest state. We define the displacement measure $\mathrm{disp}_i(\mathbf{r}_j)$ using the Euclidean distance between the deformed position $\Upsilon_i(\mathbf{r}_j)$ and the rest position $\bar{\mathbf{v}}_i$ as

$$\mathrm{disp}_i(\mathbf{r}_j) := \left|\Upsilon_i(\mathbf{r}_j) - \bar{\mathbf{v}}_i\right|_2 \ . \tag{7}$$

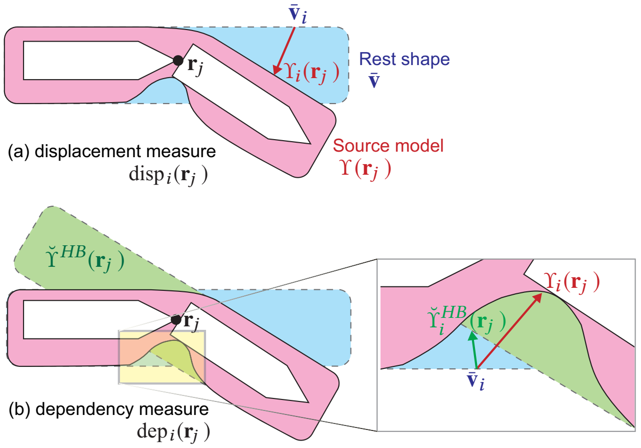
This measure takes a non-zero value according to the joint rotation $\mathbf{r}_j$, if the joint has any influence on the vertex. In contrast, the joint that has no effect on the vertex holds $\forall \mathbf{r}_j \in SO(3), \mathrm{disp}_i(\mathbf{r}_j) \leq \epsilon$, where the positive cutoff threshold ϵ is used to take the numerical error into consideration.

For the rotation samples satisfying $\mathrm{disp}_i(\mathbf{r}_j) > \epsilon$, the dependency measure $\mathrm{dep}_i(\mathbf{r}_j)$ is defined using an inner product of the two displacement vectors as

$$\mathrm{dep}_i(\mathbf{r}_j) := \frac{\left\{\Upsilon_i(\mathbf{r}_j) - \bar{\mathbf{v}}_i\right\} \cdot \left\{\check{\Upsilon}_i^{HB}(\mathbf{r}_j) - \bar{\mathbf{v}}_i\right\}}{|\check{\Upsilon}_i^{HB}(\mathbf{r}_j) - \bar{\mathbf{v}}_i|_2^2} \ , \tag{8}$$

$$\check{\Upsilon}_i^{HB}(\mathbf{r}_j) := \mathbf{G}_{\psi(j)}\Lambda(\bar{\mathbf{t}}_j, \mathbf{r}_j)\bar{\mathbf{G}}_j^{-1}\bar{\mathbf{v}}_i \ , \tag{9}$$

where $\check{\Upsilon}_i^{HB}(\mathbf{r}_j)$, derived from the formula of LBS (Equation 1) with maximum skinning weight $w_{i,j} = 1$, corresponds to the LBS result of the i-th vertex using only the j-th joint, as illustrated in Figure 2(b). This measure has an approximately linear relationship with the skinning weight. For instance, $\mathrm{dep}_i(\mathbf{r}_j) = 1$ if the behavior of the source rig $\Upsilon_i(\mathbf{r}_j)$ is exactly described by $\check{\Upsilon}_i^{HB}(\mathbf{r}_j)$ and $|\mathrm{dep}_i(\mathbf{r}_j) - 1|_1 > 0$ if the joint rotation $\mathbf{r}_j$ has less deformation influence on the i-th vertex. The importance of the rotation sample is therefore evaluated using the normalized measure $\rho/\{|\mathrm{dep}_i(\mathbf{r}_j) - 1|_1 + \rho\}$, where ρ is a positive constant used to avoid division by zero.



$\bar{\mathbf{v}}_i$
$\mathbf{r}_j$
$\Upsilon_i(\mathbf{r}_j)$
Rest shape
$\bar{\mathbf{v}}$
(a) displacement measure
$\mathrm{disp}_i(\mathbf{r}_j)$
Source model
$\Upsilon(\mathbf{r}_j)$
$\check{\Upsilon}^{HB}(\mathbf{r}_j)$
$\mathbf{r}_j$
$\Upsilon_i(\mathbf{r}_j)$
$\check{\Upsilon}_i^{HB}(\mathbf{r}_j)$
$\bar{\mathbf{v}}_i$
(b) dependency measure
$\mathrm{dep}_i(\mathbf{r}_j)$

Figure 2: Displacement measure and dependency measure. (a) The displacement measure is computed as the Euclidean distance between the deformed vertex position and the rest position. (b) The dependency measure quantifies the similarity between the deformation of the source rig and the global rotation around the joint.

4.4 Selection of Important Samples

The number of rotation samples is reduced based on the displacement measure $\mathrm{disp}_i(\mathbf{r}_j)$ and the dependency measure $\mathrm{dep}_i(\mathbf{r}_j)$. First, the rotation samples that give the strongest dependency and the largest displacement are selected for all combinations of vertices and joints as $\forall(i,j) \in \mathcal{V}\times\mathcal{J}, \mathbf{r}_{i,j}^{\mathrm{dep}*} = \mathrm{argmax}_{\mathbf{r}_{i,j}\in\mathcal{R}_j}\, \rho/\{|\mathrm{dep}_i(\mathbf{r}_j) - 1|_1 + \rho\}$, $\mathbf{r}_{i,j}^{\mathrm{disp}*} = \mathrm{argmax}_{\mathbf{r}_{i,j}\in\mathcal{R}_j}\, \mathrm{disp}_i(\mathbf{r}_{i,j})$. Next, the selected samples of the same joint index are classified into sets of $\mathcal{R}_j^{\mathrm{dep}} = \{\mathbf{r}_{i,k}^{\mathrm{dep}*} | i \in \mathcal{V}, k \in \mathcal{J}, k = j\}$ and $\mathcal{R}_j^{\mathrm{disp}} = \{\mathbf{r}_{i,k}^{\mathrm{disp}*} | i \in \mathcal{V}, k \in \mathcal{J}, k = j\}$ for each joint. Finally, given a user-specified number of active samples M, a set of active samples $\mathcal{R}_j^*$ is composed by selecting the $M/2$ samples with the highest dependency from $\mathcal{R}_j^{\mathrm{dep}}$ and the $M/2$ samples with the largest displacement from $\mathcal{R}_j^{\mathrm{disp}}$. As a result, $n(\mathcal{Q}) = M \cdot n(\mathcal{J})$ pose samples are used for the skinning weights optimization (§4.1) and the helper bone rigging described in the next section. Note that $n(\mathcal{R}_p)$ is in general more than 200 and M is experimentally set to $M \leq 20$.

5 RIGGING OF NON-RIGID HELPER BONES

We extend the helper bone rigging procedure [Mukai 2015] to deal with non-rigid transformation, including non-uniform scaling components in addition to the rotation and translation. The rigging starts with a converted LBS-based skeleton rig without any helper bones. First, helper bones are incrementally inserted into the LBS rig (§5.2). Next, its optimal transformation $(\mathbf{A}_{h,q})_{h \in \mathcal{H}, q \in Q}$ for each pose sample and skinning weights $(w_{i,j}, w_{i,h})$ is updated by solving the least-square problem:

$$\min \sum_{q \in Q} \Big| \Upsilon_i(\mathbf{p}_q) - \sum_{h \in \mathcal{H}} w_{i,h} \mathbf{A}_{h,q} \bar{\mathbf{v}}_i - \sum_{j \in \mathcal{J}} w_{i,j} \mathbf{G}_{\psi(j)} \Lambda(\bar{\mathbf{t}}_{j,q}, \mathbf{r}_{j,q}) \bar{\mathbf{G}}_j^{-1} \bar{\mathbf{v}}_i \Big|_2^2 , \qquad (10)$$

where we omit the constraint on the helper bone transformation and constraints related to skinning weights (Equations 4, 5, and 6). This is the skinning decomposition with similarity transformation problem and is solved by the alternative least-square technique (§5.3). Finally, the helper bone controller $\mathbf{M}_h$ is obtained as a linear regression model by learning the relation between the skeleton pose $\mathbf{p}_q$ and the optimized helper bone transformation $\mathbf{A}_{h,q}$ (§5.4). Before giving the details of this procedure, let us describe our point cloud registration algorithm with similarity transformation in the next section.

5.1 Point Cloud Registration with Similarity Transformation

Given two point sets $(\mathbf{v}_i)_{i \in \mathcal{V}}$ and $(\mathbf{u}_i)_{i \in \mathcal{V}}$ with point correspondence information, the best similarity transformation to minimize the difference between them is defined as

$$(\mathbf{t}^*, \mathbf{R}^*, \mathbf{s}^*) = \underset{\mathbf{t}, \mathbf{R}, \mathbf{s}}{\operatorname{argmin}} \sum_{i \in \mathcal{V}} |\mathbf{R}(\mathbf{s} \otimes \mathbf{v}_i) + \mathbf{t} - \mathbf{u}_i|_2^2 ,$$
$$\text{subject to } \ \mathbf{R} \in SO(3), \ \ \mathbf{s} \geq \mathbf{0} , \qquad (11)$$

where $\mathbf{t}$, $\mathbf{R}$, and $\mathbf{s}$ are a translation vector, a rotation matrix, and a scale vector, respectively, and $\otimes$ represents element-wise multiplication. This is a non-linear convex optimization problem that has no closed-form solution to the best of our knowledge. We use a coordinate descent algorithm that alternately optimizes the rigid transformation $(\mathbf{R}, \mathbf{t})$ and the scale $\mathbf{s}$. The iterations start with $\mathbf{s} = \mathbf{1}$, $\mathbf{R} = \mathbf{I}$, and $\mathbf{t} = \mathbf{0}$. The rotation $\mathbf{R}$ and the translation $\mathbf{t}$ are updated by solving an absolute orientation problem between two point sets $(\mathbf{s} \otimes \mathbf{v}_i)_{i \in \mathcal{V}}$ and $(\mathbf{u}_i)_{i \in \mathcal{V}}$ [Horn 1987] while the scale $\mathbf{s}$ is fixed. Next, the scale component is updated using a least-square technique as $\mathbf{s} = [\sum_{i \in \mathcal{V}} (\mathbf{R}^{-1}\mathbf{u}_i - \mathbf{t}) \otimes \mathbf{v}_i] \oslash [\sum_{i \in \mathcal{V}} \mathbf{v}_i \otimes \mathbf{v}_i]$, where $\oslash$ denotes element-wise division. We use the soft thresholding technique $\mathbf{s} = \max(\mathbf{s}, \mathbf{0})$ to maintain the non-negative value. This iterative process is repeated several times and quickly converges to the global optimal. This approach guarantees a stable conversion to the global optimal by virtue of a closed-form algorithm at each optimization step.

5.2 Helper Bone Insertion

Our method incrementally inserts a new helper bone into the region where the largest approximation error is yielded. Our system first searches for a vertex with the largest squared error summed over all training samples. Next, a similarity transformation that best approximates the displacement of the found vertex and its one-ring neighbors is computed using the algorithm described in the previous section. The new helper bone is then inserted using the estimated transformation. Finally, the skinning weights $(w_{i,j}, w_{i,h})$ and the transformation matrix of all helper bones $\mathbf{A}_{h,q}$ are updated by solving the skinning decomposition algorithm described in the next section. This process is repeated until the specified number of helper bones is reached.

5.3 Skinning Decomposition with Similarity Transformations

We employ the block coordinate descent techniques presented in [Le and Deng 2012] for alternately solving the two subproblems of skinning weight optimization and helper bone transformation optimization. This approach updates the skinning weights by maintaining all joint transformations at each subiteration, which amounts to the same problem as Equation 3, except in the case of helper bones. The transformation of the helper bones is updated one-by-one while the other variables remain fixed. More specifically, the similarity transformations of the h-th helper bone for training samples are optimized while the skinning weights and the transformations of both the primary joints and the other helper bones remain fixed at each subiteration.

Let us rewrite the product of the similarity transformation matrix $\mathbf{A}_{h,q}$ and the rest vertex position $\bar{\mathbf{v}}_i$ as $\mathbf{A}_{h,q}\bar{\mathbf{v}}_i = [\mathbf{R}_{h,q}(\mathbf{s}_{h,q} \otimes \bar{\mathbf{v}}_i) + \mathbf{t}_{h,q}]$. The transformation estimation is then formulated as

$$\underset{\mathbf{t}_{h,q}, \mathbf{R}_{h,q}, \mathbf{s}_{h,q}}{\operatorname{argmin}} \sum_{i \in \mathcal{V}} \Big| \mathbf{u}_{i,q} - w_{i,h} \left[\mathbf{R}_{h,q}(\mathbf{s}_{h,q} \otimes \bar{\mathbf{v}}_i) + \mathbf{t}_{h,q} \right] \Big|_2^2 \qquad (12)$$
$$\text{subject to } \ \mathbf{R}_{h,q} \in SO(3), \ \ \mathbf{s}_{h,q} \geq \mathbf{0} , \qquad (13)$$

where

$$\mathbf{u}_{i,q} = \Upsilon_i(\mathbf{p}_q) - \sum_{j \in \mathcal{J}} w_{i,j} \mathbf{G}_{\psi(j)} \Lambda(\bar{\mathbf{t}}_j, \mathbf{r}_j) \bar{\mathbf{G}}_j^{-1} \bar{\mathbf{v}}_i - \sum_{\hat{h} \in \mathcal{H}, \hat{h} \neq h} w_{i,\hat{h}} \left[\mathbf{R}_{\hat{h},q}(\mathbf{s}_{\hat{h},q} \otimes \bar{\mathbf{v}}_i) + \mathbf{t}_{\hat{h},q} \right] . \qquad (14)$$

The optimal skinning weights and helper bone transformation for each training pose are estimated using an extended version of the block coordinate descent algorithm [Le and Deng 2012] with similarity transformation estimation. The iterations start with $\mathbf{s}_{h,q} = \mathbf{1}$, $\mathbf{R}_{h,q} = \mathbf{I}$, and $\mathbf{t}_{h,q} = \mathbf{0}$. All vertices are first translated to bring the center of rotation to the origin as

$$\tilde{\mathbf{v}}_i = \mathbf{v}_i - \mathbf{v}^* , \qquad \tilde{\mathbf{u}}_{i,q} = \mathbf{u}_{i,q} - w_{i,h}\mathbf{u}_q^* , \qquad (15)$$

$$\text{where } \ \mathbf{v}^* = \frac{\sum_{i \in \mathcal{V}} w_{i,h}^2 \mathbf{u}_{i,q}}{\sum_{i \in \mathcal{V}} w_{i,h}^2} , \qquad \mathbf{u}_q^* = \frac{\sum_{i \in \mathcal{V}} w_{i,h} \mathbf{u}_{i,q}}{\sum_{i \in \mathcal{V}} w_{i,h}^2} . \qquad (16)$$

The optimal similarity transformation between $(\tilde{\mathbf{v}}_i)_{i \in \mathcal{V}}$ and $(\tilde{\mathbf{u}}_{i,q})_{i \in \mathcal{V}}$ is then estimated as the h-th helper bone transformation using the algorithm described in §5.3. After the transformations of all helper bones are updated, the skinning weights are updated by solving Equation 3. This iterative process is repeated several times, where we used 20 iterations for all experiments.

5.4 Constructing Helper Bone Controllers

The helper bone controller $\mathbf{M}_h$ computes the optimal local similarity transformation according to the pose of the primary skeleton. Each controller is modeled as a linear regression function and built by learning the relation between the skeleton pose sample $\mathbf{p}_q$ and the optimized similarity transformation $\mathbf{t}_{h,q}$, $\mathbf{R}_{h,q}$, and $\mathbf{s}_{h,q}$. We first extract the local transformation components relative to the parent joint from the global transformation. Assuming $\mathbf{M}_{h,q} = \mathbf{I}$ holds at the rest pose, $\bar{\mathbf{G}}_h = \bar{\mathbf{G}}_{\psi(h)}$ is satisfied by the definition of forward kinematics and we can uniquely extract the local transformation relative to the parent joint by

$$\mathbf{M}_{h,q} = \mathbf{G}_{\psi(h)} \Lambda(\mathbf{t}_{h,q}, \mathbf{R}_{h,q}, \mathbf{s}_{h,q}) \bar{\mathbf{G}}_{\psi(h)}^{-1} , \tag{17}$$

and the homogeneous matrix $\mathbf{M}_{h,q}$ is then decomposed into the local transformation components $\mathbf{r}_{h,q}^l$, $\mathbf{t}_{h,q}^l$, and $\mathbf{s}_{h,q}^l$. Note that we omit the algorithm used to select the best $\psi(h)$ in this paper. Please refer to [Mukai 2015] for the details.

We use a B-th order polynomial function as a regression model. The transformation of each helper bone is computed by

$$\begin{bmatrix} \mathbf{t}_h^{l\,T} & \mathbf{r}_h^{l\,T} & \mathbf{s}_h^{l\,T} \end{bmatrix}^T = \mathbf{F}_h \begin{bmatrix} 1 & \mathbf{x}_1^T & \cdots & \mathbf{x}_{n(\mathcal{J})}^T \end{bmatrix}^T , \tag{18}$$

where $\mathbf{F}_h$ is a regression coefficient matrix, $\mathbf{x}_j \in \Re^{{}_4\mathrm{H}_B - 1}$ is an independent variable vector that is composed of all variables of the P-th order polynomial of the joint rotation $\mathbf{r}_j$, and ${}_a\mathrm{H}_b$ denotes the number of combinations with repetitions. For example, if we take $B = 2$, the independent variable vector from $\mathbf{r} = [r_1, r_2, r_3]$ is $\mathbf{x} = [r_1, r_2, r_3, r_1^2, r_2^2, r_3^2, r_1r_2, r_1r_3, r_2r_3]$.

Our method builds an efficient model by estimating the sparse matrix $\mathbf{F}_h \in \Re^{9\times(1+n(\mathcal{J})\cdot\dim(\mathbf{x}_j))}$ by solving a lasso problem [Tibshirani 2011] as

$$\min_{\mathbf{F}_h} |\mathbf{Y}_h - \mathbf{F}_h\mathbf{X}|_2^2 + \beta\,|\mathbf{F}_h|_1 , \tag{19}$$

where

$$\mathbf{Y}_h = \begin{bmatrix} \mathbf{t}_{h,1}^l & \cdots & \mathbf{t}_{h,n(Q)}^l \\ \mathbf{r}_{h,1}^l & \cdots & \mathbf{r}_{h,n(Q)}^l \\ \mathbf{s}_{h,1}^l & \cdots & \mathbf{s}_{h,n(Q)}^l \end{bmatrix} \in \Re^{9\times n(Q)},$$

$$\mathbf{X} = \begin{bmatrix} 1 & \cdots & 1 \\ \mathbf{x}_{1,1} & \cdots & \mathbf{x}_{1,n(Q)} \\ \vdots & \ddots & \vdots \\ \mathbf{x}_{n(\mathcal{J}),1} & \cdots & \mathbf{x}_{n(\mathcal{J}),n(Q)} \end{bmatrix} \in \Re^{(1+\sum_{j\in\mathcal{J}}\dim(\mathbf{x}_j))\times n(Q)},$$

and β is the shrinkage parameter that controls the trade-off between model accuracy and the number of non-zero coefficients.

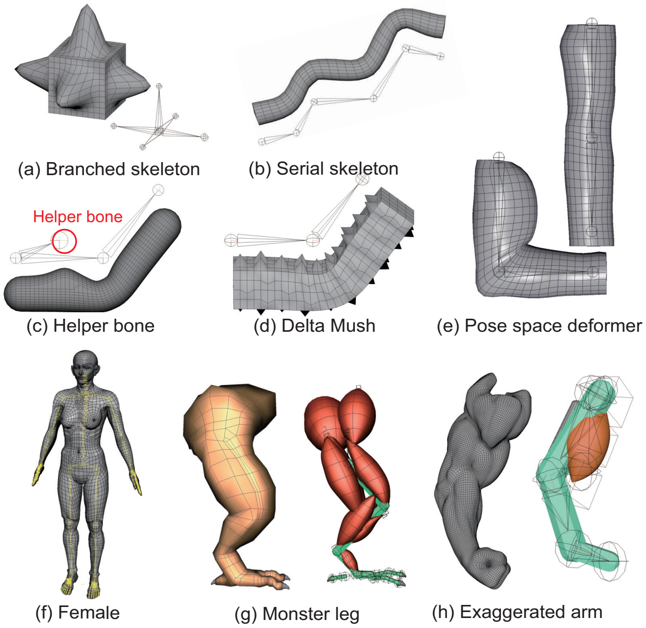


Figure 3: Test assets used for the experiments.

Table 1: Statistics of rig conversion performance

Model	$n(\mathcal{V})$	$n(\mathcal{J})$	$n(\mathcal{H})$	α [%]	Time [s]
Branch (LBS)	602	6	0	0.00	3.5
Serial (LBS)	622	6	0	0.00	3.4
Helper bone	1962	3	1	1.81	49.9
Delta Mush	514	3	0	0.86	6.4
Arm (PSD)	622	3	4	2.97	12.8
Female (CBS)	11356	142	0	1.57	1054
Leg (Muscle)	522	20	2	7.85	88.0
Arm (Muscle)	15768	10	2	3.26	329.4

6 EXPERIMENTAL RESULTS

We implemented our prototype system as a plug-in for Autodesk Maya 2017 with Python. The computation time was measured on a workstation with dual Intel Xeon E5-2687W CPUs 3.1 GHz and 192 GB RAM. In all the experiments, the maximum number of influencing joints was set to $\chi = 4$, the sampling range to $\forall j \in \mathcal{J}, \sigma_j = 90^\circ$, the number of active samples to $M = 20$, and the displacement threshold ϵ to $\epsilon = 1.0 \times 10^{-2}$. For sampling the joint rotations, $n(\mathcal{S}) = 576$ rotation samples were uniformly generated, which approximately corresponds to sampling every 7.5°. We used random sampling with $M = 20$ samples as the baseline method.

The conversion capability and computational performance were evaluated using several types of source model, as shown in Figure 3 and Table 1. The accuracy of the conversion was quantified using a distortion measure α [Karni and Gotsman 2004] for the scale-invariant evaluation:

$$\alpha := 100\sqrt{\frac{\sum_{i\in\mathcal{V}}\sum_{q\in Q_{rand}} |\Upsilon_i(\mathbf{p}_q) - \Upsilon_i^{HB}(\mathbf{p}_q)|_2^2}{\sum_{i\in\mathcal{V}}\sum_{q\in Q_{rand}} |\Upsilon_i(\mathbf{p}_q) - \bar{\mathbf{v}}_i|_2^2}} , \tag{20}$$

where Q_{rand} is a set of indices of pose samples generated by randomly selecting the joint rotations $\mathbf{r}_{j\in\mathcal{J}}$ from the valid samples $\mathcal{R}_{j\in\mathcal{J}}$. We used $n(Q_{rand}) = 10^4$ samples for all the experiments.

6.1 Linear Blend Skinning Rig

We evaluated the conversion performance using an LBS model with known skinning weights. The test model consisted of $n(\mathcal{V}) = 602$ vertices and was bound to $n(\mathcal{J}) = 6$ joints, as shown in Figure

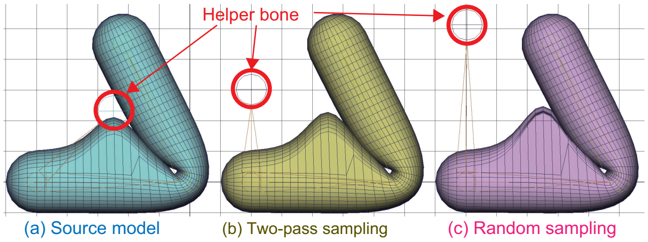


Figure 4: Conversion from helper bone rig. The similar skin deformation was reproduced by the different behavior of the helper bone.

3(a). The skinning weights were automatically initialized using a skinning function of Maya 2017 with the sparsity constraint $\chi = 4$.

The conversion results showed a perfect reconstruction, as the root mean squared error of skinning weights was 4.0×10^{-9} and there was no distortion in the skin deformation as $\alpha = 0.00\%$. The random sampling method also resulted in the perfect conversion as $\alpha = 0.00\%$. The rig conversion took 3.5 s for the per-joint sampling, deformation influence analysis, and skinning weight optimization. An additional LBS model (Figure 3(b)) was also successfully reconstructed with $\alpha = 0.00\%$ in 3.4 s using our method, and with $\alpha = 0.00\%$ in 2 s using the random sampling method.

6.2 Helper Bone Rig

We used a simple helper bone model with known skinning weights and manually designed the helper bone controller, as shown in Figure 3(c). This model had $n(\mathcal{J}) = 3$ joints and $n(\mathcal{H}) = 1$ rigid helper bone, which was procedurally controlled according to the middle joint's rotation for expanding the left half of the skin shape. We reconstructed this helper bone rig by sampling the three primary joints' rotations, and the conversion took approximately 50 s. Figure 4 compares the source model, the converted helper bone rigs ($\alpha = 1.81\%$), and the result using random sampling ($\alpha = 1.88\%$). Whereas the random sampling caused visible distortion around the expanding area, our method successfully approximated the original deformation. The estimated helper bone was located higher than in the source model, and the skinning weights of the helper bone correspondingly became smaller. This difference was caused by the high redundancy in the rigging. Our method found one solution to approximate the source deformation from many feasible solutions.

The computation time of the runtime helper bone control was about 30 μs per frame with Python implementation, which increases proportionally to the number of primary joints and the number of helper bones. We could further improve the performance by parallelizing the execution of helper bone controllers using a more efficient implementation.

6.3 Delta Mush Deformer

Delta Mush [Mancewicz et al. 2014] provides detail-preserving skinning by decomposing the skin mesh model into a low-frequency shape and high-frequency details. It then performs LBS on the low-frequency shape and applies the high frequency component back to the deformed shape. We used a square pole model with many thorns on each surface (Figure 3(d)) for the conversion experiment. The conversion took 3 s and resulted in a distortion of $\alpha = 0.86\%$. Visible artifacts occurred around the rotating joint, as shown in Figure 5. The LBS model with the original skinning weights caused a large distortion of $\alpha = 5.18\%$, as the thorns around the rotating joints are folded because of the so-called elbow collapse artifact of LBS. In contrast, our method provides a better deformation quality by refining the skinning weight using an example-based optimization.

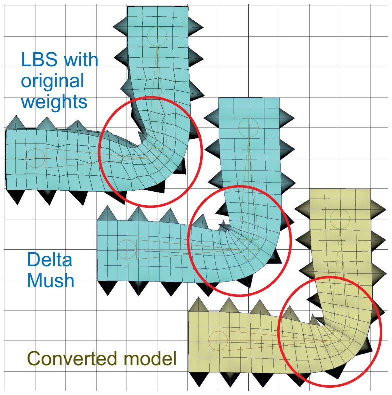


Figure 5: Conversion from Delta Mush deformer. Our method preserved the thorn shapes using the sampling-based skinning technique.

6.4 Pose Space Deformation Rig

We compared the conversion capability of our method with a rig conversion function, named the Maya bake deformer tool, using a pose space deformer (PSD) model shown in Figure 3(e). This PSD rig generates wrinkles around the middle joint by interpolating five blendshapes according to the rotation of the middle joint. The bake deformer tool caused a large elbow-collapse artifact in the converted model, probably because of its naive sampling method, as shown in Figure 6(c). In contrast, our method reproduced the near-rigid deformation behavior of the source model, except for the wrinkles and the bulging of the biceps, when we added no helper bone (Figure 6(d)). The detailed deformation was successfully approximated using four helper bones (Figure 6(b)). These results demonstrate the accuracy of our two-pass sampling method and the non-rigid deformation approximation using similarity transformation. The computation time of each helper bone controller was about 40 μs per frame.

6.5 Corrective Blendshapes

We used a female character model consisting of $n(\mathcal{V}) = 11356$ vertices and 112 corrective blendshapes (CBSs) bound to $n(\mathcal{J}) = 142$ joints, including the fingers and toes, as shown in Figure 3(f). The CBSs were used for compensating the deformation of several body parts (e.g., shape deformation around each shoulder was generated by the combination of LBS and four CBSs driven by the shoulder rotation). Our method took 1054 s for the conversion, and resulted in a visible distortion of $\alpha = 1.57\%$. We found that the reconstruction error occurred mainly around the shoulders and torso, which exhibit complicated deformation behavior when using CBSs. We

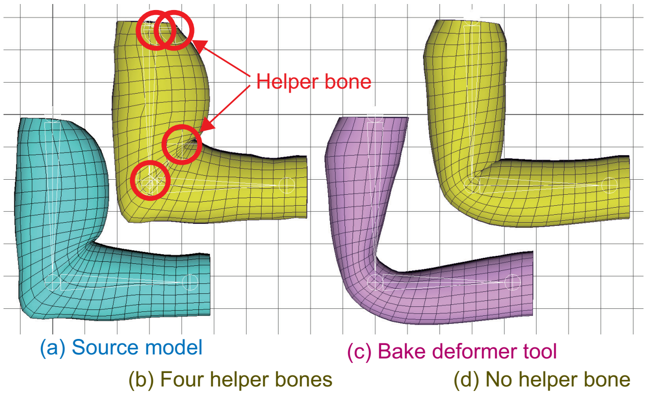


Figure 6: Pose space deformation rig. Our method reproduced the wrinkles around the joint using four helper bones.

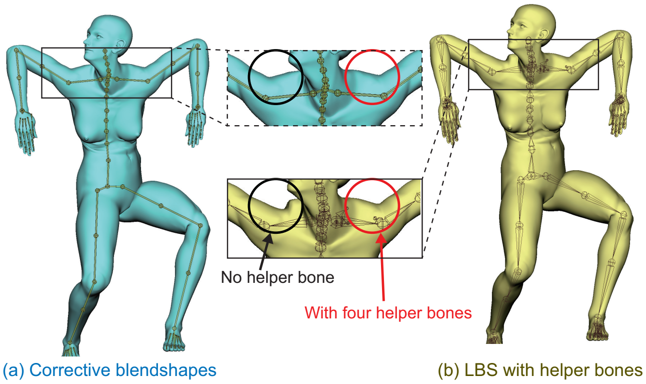


Figure 7: Conversion from corrective blendshapes. The collapse around the left shoulder was successfully compensated using the procedural control of four helper bones.

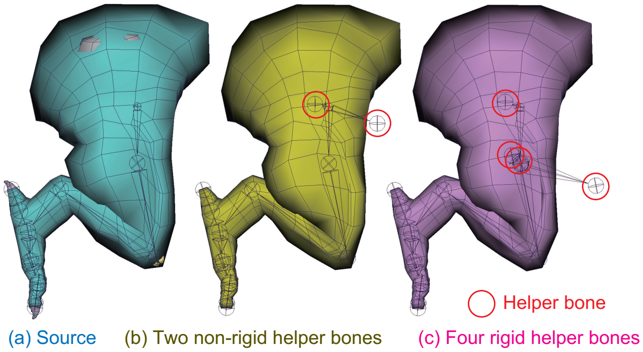


Figure 8: Conversion of monster's leg. (b) The expansion of the thigh was approximated using two non-rigid helper bones. (c) Similar results could be achieved using at least four rigid helper bones without the scale component.

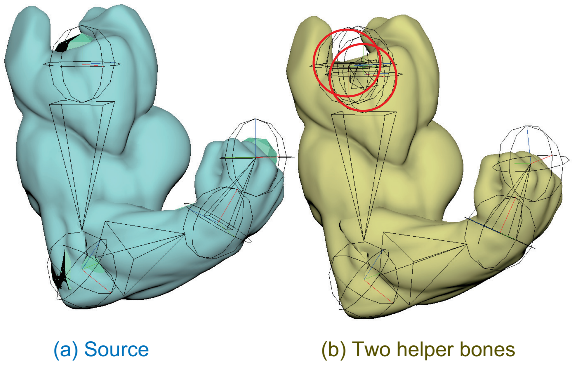


Figure 9: Conversion of exaggerated arm. The expansion of one biceps was approximated using two non-rigid helper bones.

added four helper bones to compensate the error around the left shoulder, as shown in Figure 7. The computation time for adding the helper bones was approximately 30 s. The LBS artifact was successfully compensated by the procedural control of the helper bone transformation.

6.6 Virtual Muscle System

We used a monster's leg asset of a Maya tutorial, supplied by Autodesk Inc., which was built using the muscle function of Maya 2017 (Figure 3(g)). The skeleton has $n(\mathcal{J}) = 20$ joints including toes, and the 11 muscles expand and contract to drive the deformation of $n(\mathcal{V}) = 552$ vertices according to the movement of the skeleton. The conversion with two helper bones took 88 s, the runtime computation took approximately 64 μs for each helper bone, and the distortion measured 7.85%. We also evaluated the conversion with four helper bones that had only the rigid components and disabled the scaling. The computation time for the conversion was 93.7 s and resulted in a distortion measure of 7.69%. These results demonstrate that non-linear skin deformation, such as expansion of the thigh, is well emulated by the similarity transformations of the small number of helper bones, as shown in Figure 8.

An additional experimental asset consisted of one exaggerated virtual muscle emulating the bulging of biceps (Figure 3(h)). The expansion of the one muscle was well emulated by the non-uniform scaling of two helper bones, as shown in Figure 9. In contrast, the same model required four rigid helper bones spreading in every direction when we disabled the optimal scale estimation.

7 DISCUSSION

We proposed a sampling-based method for automatically converting from arbitrary skeleton-driven deformers to an LBS-based helper bone rig with similarity transformation. Our method generates important training samples from the source model based on the deformation influence analysis via per-joint, two-pass sampling. This approach successfully reduces the size of the training dataset to provide efficient and accurate example-based rigging. We demonstrated that our automated system converts several types of skeleton deformers, with tens to hundreds of joints, to a small number of non-rigid helper bones.

The basic assumption of our method is that the source rig always has a skeleton hierarchy. While this assumption is practical, because skeleton-driven deformers are widely used in many productions, we should further extend our sampling-based method to support non-skeletal rigs such as cage-based deformers and blendshapes. The skeletal rigging technique [Le and Deng 2014] is a promising approach, which constructs an LBS-based skeleton rig from an example animation generated by sampling the skin deformations of the source model. The additional assumption is that both the vertex position and the vertex normal are computed using the same skinning weights and bone transformations. Our method thus evaluates the conversion accuracy based only on the position. However, the normal vectors also play an important role in the real-time rendering, and sometimes different skinning weights are used for the skinning of vertex normals. We should investigate a skinning decomposition algorithm for approximating the vertex normals.

Manual modification of the converted rig should be considered for practical purposes. Our current system allows artists to edit the skinning weights using the common paint interface provided by animation packages. However, the behavior of the helper bone controller can be modified by editing the regression coefficient matrix, which requires in-depth knowledge of the regression technique. We should explore a more intuitive interface for editing the helper bone controller.

The current method applies the constant threshold ϵ over the entire skin surface. However, the optimal setting might vary with the size of the body parts, such as the thigh and little finger. Moreover, we have no theoretical grounding of the minimality of the training dataset. The number of rotation samples M and the number of uniform samples of the spatial rotations $\mathcal{S}$ were empirically determined. Our future work will include an investigation of the theoretical basis of the settings of the cutoff thresholds and the optimality of the training dataset generation.

One of the main applications of our method is in the conversion from an arbitrary, computationally heavy rig to a lightweight asset for hard real-time animation systems. This scenario also demands mesh reduction and skeleton retargeting, which should be integrated into a unified system for the asset conversion. One possible direction is to simultaneously simplify the skin mesh geometry and skeleton hierarchy during the rig conversion process. Our deformation influence analysis may be useful for eliminating insignificant polygonal faces and skeleton joints. An extension to non-linear blending [Kavan et al. 2007; Wang and Phillips 2002] and to helper bone-based dynamic skinning [Mukai and Kuriyama 2016] is interesting future work.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number 15K16110, 15H02704. We would like to thank PlatinumGames Inc. for providing useful suggestions and the arm assets. We also thank the anonymous reviewers for their helpful comments.

REFERENCES

Dicko Ali-Hamadi, Tiantian Liu, Benjamin Gilles, Ladislav Kavan, Francois Faure, Olivier Palombi, and Marie-Paule Cani. 2013. Anatomy Transfer. *ACM Transactions on Graphics* 32, 6 (2013), 188:1–188:8.

Alexis Angelidis and Karan Singh. 2007. Kinodynamic Skinning using Volume-preserving Deformations. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2007*. 129–140.

Quentin Avril, Donya Ghafourzadeh, Srinivasan Ramachandran, Sahel Fallahdoust, Sarah Ribet, Olivier Dionne, Martin de Lasa, and Eric Paquette. 2016. Animation Setup Transfer for 3D Characters. *Computer Graphics Forum* 35, 2 (2016), 115–126.

Ilya Baran and Jovan Popović. 2007. Automatic Rigging and Animation of 3D Characters. *ACM Transactions on Graphics* 26, 3 (2007), 72:1–72:8.

Berthold K. P. Horn. 1987. Closed-form Solution of Absolute Orientation Using Unit Quaternions. *Journal of Optical Society of America A* 4, 4 (1987), 629–642.

Alec Jacobson, Ladislav Kavan, Ilya Baran, Jovan Popović, and Olga Sorkine. 2012. Fast Automatic Skinning Transformations. *ACM Transactions on Graphics* 31, 4 (2012), 77:1–77:10.

Doug L. James and Christopher D. Twigg. 2005. Skinning Mesh Animations. *ACM Transactions on Graphics* 24, 3 (2005), 399–407.

Petr Kadlecek, Alexandru-Eugen Ichim, Tiantian Liu, Ladislav Kavan, and Jaroslav Krivanek. 2016. Reconstructing Personalized Anatomical Models for Physics-based Body Animation. *ACM Transactions on Graphics* 35, 6 (2016), 213:1–213:13.

Zachi Karni and Craig Gotsman. 2004. Compression of Soft-body Animation Sequences. *Computers & Graphics* 28, 1 (2004), 25–34.

Ladislav Kavan, Steven Collins, Jiri Zara, and Carol O'Sullivan. 2007. Skinning with Dual Quaternions. In *Proc. of ACM SIGGRAPH Symposium on Interactive 3D Graphics 2007*. 39–46.

Ladislav Kavan, Peter-Pike Sloan, and Carol O'Sullivan. 2010. Fast and Efficient Skinning of Animated Meshes. *Computer Graphics Forum* 29, 2 (2010), 327–336.

Ladislav Kavan and Olga Sorkine. 2012. Elasticity-Inspired Deformers for Character Articulation. *ACM Transactions on Graphics* 31, 6 (2012), 196.

Paul G. Kry, Doug L. James, and Dinesh K. Pai. 2002. EigenSkin: Real Time Large Deformation Character Skinning in Hardware. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2002*. 153–159.

Binh Huy Le and Zhigang Deng. 2012. Smooth Skinning Decomposition with Rigid Bones. *ACM Transactions on Graphics* 31, 6 (2012), 199:1–199:10.

Binh Huy Le and Zhigang Deng. 2014. Robust and Accurate Skeletal Rigging from Mesh Sequences. *ACM Transactions on Graphics* 33, 4 (2014), 84:1–84:10.

Binh Huy Le and Jessica K. Hodgins. 2016. Real-time Skeletal Skinning with Optimized Centers of Rotation. *ACM Transactions on Graphics* 35, 4 (2016), 37:1–37:10.

J. P. Lewis, Matt Cordner, and Nickson Fong. 2000. Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation. In *Proc. of SIGGRAPH 2000*. 165–172.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Transactions on Graphics* 34, 6 (2015), 248:1–248:16.

Nadia Magnenat-Thalmann, Richard Laperrire, and Daniel Thalmann. 1988. Joint-dependent Local Deformations for Hand Animation and Object Grasping. In *Proc. of Graphics Interface 88*. 26–33.

Joe Mancewicz, Matt L. Derksen, Hans Rijpkema, and Cyrus A. Wilson. 2014. Delta Mush: Smoothing Deformations While Preserving Detail. In *Proceedings of the Symposium on Digital Production*. 7–11.

Christian Miller, Okan Arikan, and Donald S. Fussell. 2011. Frankenrigs: Building Character Rigs From Multiple Sources. *IEEE Transactions on Visualization and Computer Graphics* 17, 8 (2011), 1060–1070.

Alex Mohr and Michael Gleicher. 2003. Building Efficient, Accurate Character Skins from Examples. *ACM Transactions on Graphics* 22, 3 (2003), 562–568.

Tomohiko Mukai. 2015. Building Helper Bone Rigs from Examples. In *Proc. of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2015*. 77–84.

Tomohiko Mukai and Shigeru Kuriyama. 2016. Efficient Dynamic Skinning with Low-Rank Helper Bone Controllers. *ACM Transactions on Graphics* 35, 4 (2016), 36:1–36:11.

Jason Parks. 2005. Helper Joints: Advanced Deformations on RunTime Characters. In *Game Developers Conference 2005*.

Shunsuke Saito, Zi-Ye Zhou, and Ladislav Kavan. 2015. Computational Bodybuilding: Anatomically-based Modeling of Human Bodies. *ACM Transactions on Graphics* 4, 34 (2015), 41:1–41:12.

Jaewoo Seo, Yeongho Seol, Daehyeon Wi, Younghui Kim, and Junyong Noh. 2010. Rigging Transfer. *Computer Animation and Virtual Worlds* 21, 3–4 (2010), 375–386.

Robert Tibshirani. 2011. Regression Shrinkage and Selection Via the Lasso: A Retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73, 3 (2011), 273–282.

Robert Y. Wang, Kari Pulli, and Jovan Popović. 2007. Real-Time Enveloping with Rotational Regression. *ACM Transactions on Graphics* 26, 3 (2007), 73:1–73:10.

Xiaohuan Corina Wang and Cary Phillips. 2002. Multi-weight Enveloping: Least-Squares Approximation Techniques for Skin Animation. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 129–138.

Anna Yershova, Swati Jain, Steven M. LaValle, and Julie C. Mitchell. 2010. Generating Uniform Incremental Grids on SO(3) Using the Hopf Fibration. *International Journal of Robotics Research* 29, 7 (2010), 801–812.