Skeleton-Aware Skin Weight Transfer for Helper Joint Rigs

Z. Cao[†][1] and T. Mukai[1]

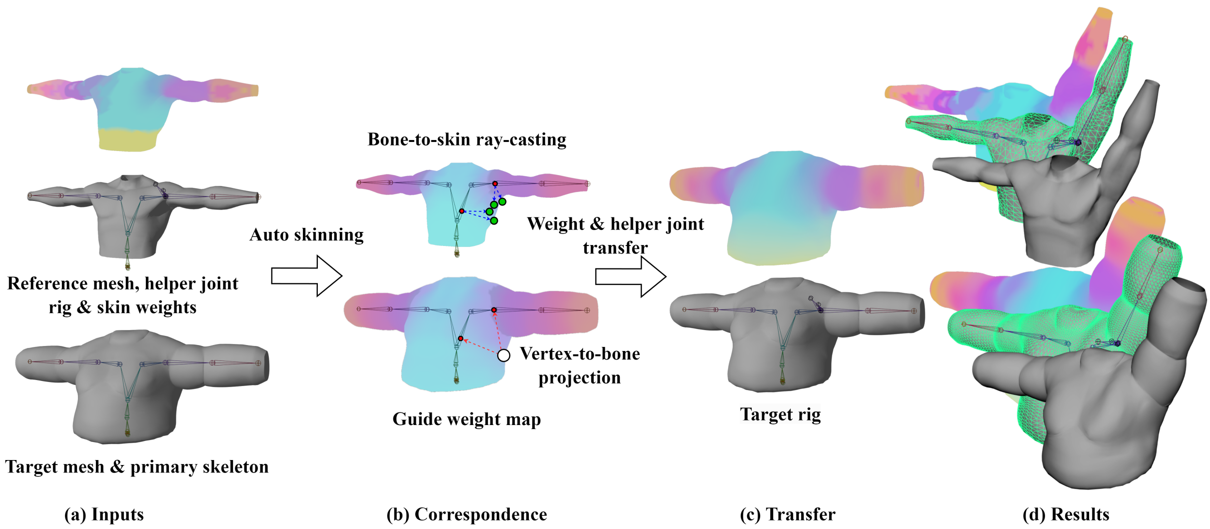

Figure 1: *Skin weights and helper joint rig are transferred from a human-like reference character to other targets. (a) The system inputs are the fully-rigged reference model, target surface mesh, and target primary skeleton. (b) A guide weight map for each model is generated using a standard skin binding method. Geometrical correspondence between the target vertex and reference skin region is found using distributed ray-casting. (c) The skin weights are transferred at each vertex by analyzing the guide weight similarity. (d) The resulting target model shows skin deformation similar to the reference model.*

Abstract

We propose a method to transfer skin weights and helper joints from a reference model to other targets. Our approach uses two types of spatial proximity to find the correspondence between the target vertex and reference mesh regions. The proposed method first generates a guide weight map to establish a relationship between the skin vertices and skeletal joints using a standard skinning technique. The correspondence between the reference and target skins is established using vertex-to-bone projection and bone-to-skin ray-casting using the guide weights. This method enables fully automated and smooth transfer of skin weight between human-like characters bound to helper joint rigs.

CCS Concepts

• ***Computing methodologies*** → ***Animation;***

1. Introduction

A real-time 3D animation system requires an efficient method to deform character shapes. Despite the recent development of machine learning-based methods, linear blend skinning (LBS) remains the de facto standard because of its stable and efficient computation. The LBS model computes skin deformation using a linear combination of local pose transformations of skeleton joints with skin weights defined on each vertex. The LBS model is often extended to provide a more complex deformation by adding *helper joints*, which are procedurally driven according to the primary skeleton movements [Par05]. A practical issue is the labor-intensive work by skillful riggers to construct the helper joint rigs. To reduce the rigging cost, automated techniques are used to setup an LBS model

† Tokyo Metropolitan University

by analyzing the geometrical relationship between the skin mesh and skeletal structure [DdL13] or by learning example skin deformations [Muk15]. These methods provide a good baseline for an LBS setup. However, they frequently require manual modification to achieve production-ready deformation quality. The other approach involves optimizing the skeleton structure and skin weights by referring to another fully-rigged model [AGR*16]. This method enables the rig structure transfer from a carefully crafted rig to various character models. However, the existing methods do not consider the helper joint rigs and often require a manual trial-and-error process to find the surface correspondence.

We propose a method to transfer skin weight for the helper joint rigs of human-like characters. The inputs to our system are a reference rigged character, rest skin shape of a target model, and the primary skeleton structure of the target, as shown in Figure 1(a). Our skeleton-aware method generates the helper joints of the target model and transfers the skin weights from the reference to inherit the reference skin deformation. Our key idea is to use two types of spatial relations between the skin vertices and primary skeleton. First, the proposed method generates *guide weight map* for each reference and target skin using a standard skinning technique to represent the skeleton-to-skin relationship, as shown in Figure 1(b). We assume that the skin vertices having similar guide weights should have similar skin weights. It is also assumed that the skin vertices in similar locations relative to the skeleton bones should have similar skin weights. Under this second assumption, the proposed algorithm limits the similarity search using vertex-to-bone projection and bone-to-mesh ray-casting. This approach enables an automated skin weight transfer based on these two correspondence analyses.

2. Related Work

Numerous methods have been proposed to transfer skin weights from the skinned character to unskinned targets. The standard off-the-shelf tool, such as the Autodesk Maya's copy skin weight tool, finds the closest mesh triangle of the reference model for each target vertex and copies the skin weight from the triangle using barycentric coordinate interpolation. A two-stage weight transfer method [ARMH23] was proposed to efficiently transfer skin weights between soft bodies such as clothes. Other methods enable weight transfer from a human-like model to their clothes. A ray-casting-based method determines the mesh correspondence by manually specifying the starting location of the ray projection [YKKL16]. The geodesic voxel binding method [DdL13] was extended to transfer skin weight while assuming that the reference and target skins share the same skeleton and that the target cloth closely covers the character skin [Got19].

The proposed method is designed to transfer skin weights between human-like characters of different proportions, and is closely related to the animation setup transfer method [AGR*16]. The conventional method transfers both the skeleton structure and skin weights by finding a dense correspondence between the reference and target meshes using a set of user-specified landmarks. By contrast, our method automatically finds the skin surface correspondence using the spatial relationship between the skin mesh and internal skeleton for an automatic weight transfer.

3. Proposed Method

The reference and target skins are assumed to have different mesh topologies. Let V^{r} and V^{t} be the number of mesh vertices of the reference and target skins, respectively. The 3D positions of the mesh vertices are denoted by $\{\mathbf{v}^{\mathrm{r}}_a | a \in \{1, \cdots, V^{\mathrm{r}}\}\}$ and $\{\mathbf{v}^{\mathrm{t}}_b | b \in \{1, \cdots, V^{\mathrm{t}}\}\}$. The reference skin is bound to J primal joints and H helper joints whose positions are represented by $\{\mathbf{p}^{\mathrm{r}}_j | j \in \{1, \cdots, J+H\}\}$. The skin weight at the a-th vertex is represented by $\mathbf{w}^{\mathrm{r}}_a \in \Re^{J+H}$. For the target skin, only the positions of primal joints $\{\mathbf{p}^{\mathrm{t}}_j | j \in \{1, \cdots, J\}\}$ are given. The goal of our method is to transfer H helper joints $\{\mathbf{p}^{\mathrm{t}}_j | j \in \{J+1, \cdots, J+H\}\}$ from the reference and optimize the skin weights $\{\mathbf{w}^{\mathrm{t}}_b \in \Re^{J+H}\}$ to mimic the reference skin deformation.

The skin weight transfer has been resolved by finding the mesh correspondence between the reference and target meshes [AGR*16; YKKL16; Got19]. These existing methods semi-automatically optimize the correspondence by comparing the mesh geometries using manually specified landmarks. By contrast, the proposed method uses the structural similarity of the primary skeletons to find the correspondence between reference and target meshes. First, it independently generates the guide weight map for each reference and the target to represent the correspondence between the primary skeleton and skin mesh. The guide weight of the reference $\omega^{\mathrm{r}}_a \in \Re^J$ and target $\omega^{\mathrm{t}}_b \in \Re^J$ are generated using an automated skinning method such as the geodesic voxel binding method [DdL13]. Note that the helper joints cannot be considered in the binding process because they are frequently arranged in the same position of another primary joint.

The vertices having similar guide weights are assumed to geometrically correspond. Given the guide weight at a-th vertex of reference ω^{r} and b-th vertex of target ω^{t}, their similarity $C \in [0,1]$ is evaluated by

$$C(\omega^{\mathrm{r}}_a, \omega^{\mathrm{t}}_b) = 1 - \frac{1}{\mathrm{card}(\mathcal{J}_{a,b})} \sum_{j \in \mathcal{J}_{a,b}} \frac{|\omega^{\mathrm{r}}_{a,j} - \omega^{\mathrm{t}}_{b,j}|_1}{\omega^{\mathrm{r}}_{a,j} + \omega^{\mathrm{t}}_{b,j}}, \quad (1)$$

where $|\cdot|_\beta$ denotes L_β-norm and $\mathcal{J}_{a,b} = \{j | j \in \{1, \cdots, J\}, \omega^{\mathrm{r}}_{a,j} + \omega^{\mathrm{t}}_{b,j} \neq 0\}$ represents the set of influencing joints.

Because similar guide weights appear in many distant areas, the proposed method limits the similarity search area by considering the spatial relationship between the skeleton bones and skin vertices. First, each vertex of the target skin $\mathbf{v}^{\mathrm{t}}_b$ is projected onto each bone connecting an influencing joint $\mathbf{p}^{\mathrm{t}}_{i \in \mathcal{I}_b}$ and its child joint $\mathbf{p}^{\mathrm{t}}_{i' \in \mathrm{child}(i)}$, where $\mathcal{I}_b = \{i | i \in \{1, \cdots, J\}, w^{\mathrm{t}}_{b,i} \neq 0|\}$ denotes a set of joint indices influencing on b-th vertex and child(i) represents a set of child joints of i-th joints, respectively. The projected position is denoted by $\mathrm{proj}(\mathbf{v}^{\mathrm{t}}_{b,i})$. Next, the corresponding location on the reference bone $\mathrm{proj}(\mathbf{v}^{\mathrm{r}}_{b,i})$ is determined using the same interpolation ratio on the target bone. S rays are then casted from $\mathrm{proj}(\mathbf{v}^{\mathrm{r}}_{b,i})$ to find the reference mesh region corresponding to $\mathbf{v}^{\mathrm{t}}_b$. Distributed ray-casting is performed in a cone shape towards the axis $\mathbf{r}$ with a cone angle θ, where the cone angle θ and the number of rays S are manually specified. The cone axis $\mathbf{r}$ is determined by rotating the projection direction $\mathbf{v}^{\mathrm{t}}_b - \mathrm{proj}(\mathbf{v}^{\mathrm{t}}_b)$ to maintain the orientation relative to the bone. When s-th ray intersects a triangle of the reference

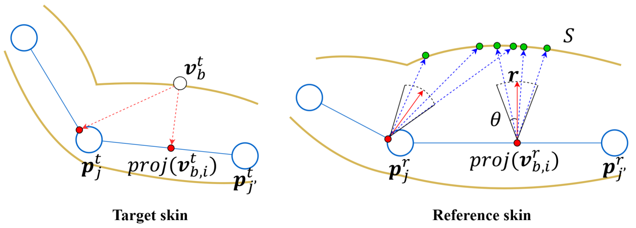

Figure 2: *The correspondence between the target vertex and reference skin region is searched with vertex-to-bone projection in the target model and bone-to-mesh ray-casting in the reference model.*

mesh, the skin weight $\mathbf{w}^{\mathrm{r}}_{b,i,s}$ and guide weight $\omega^{\mathrm{r}}_{b,i,s}$ are computed using barycentric interpolation of the weight of the triangle vertices.

The skin weight $\mathbf{w}^{\mathrm{t}}_{b}$ is then obtained by interpolating the skin weight at all intersection points as follows.

$$\mathbf{w}^{\mathrm{t}}_{b} = \sum_{i \in \mathcal{I}_b} \sum_{j=1}^{J} \omega^{\mathrm{t}}_{b,j} \sum_{s \in \mathcal{S}} C^{\alpha}(\omega^{\mathrm{t}}_{b,i}, \omega^{\mathrm{r}}_{b,i,s}) \mathbf{w}^{\mathrm{r}}_{b,i,s} , \tag{2}$$

where $\mathcal{S}$ represents a set of intersected rays and α is a power coefficient to control the fall-off rate: the larger power coefficient α produces a larger blending ratio for $\mathbf{w}^{\mathrm{r}}_{b,i,s}$ having similar guide weight $\omega^{\mathrm{r}}_{b,i,s}$. The estimated weight $\mathbf{w}^{\mathrm{t}}_{b}$ is finally modified to satisfy the affinity constraint $\sum \mathbf{w}^{\mathrm{t}}_{b,j} = 1$ and the sparsity constraint $\sum |\mathbf{w}^{\mathrm{t}}_{b,j}|_0 \leq K$ where K is the maximum number of influencing joints per vertex. Note that we experimentally confirmed that this interpolation-based approach provides more accurate and smoother weight distribution on the transfer result than the single selection of the best candidate, which is expressed as $\mathbf{w}^{\mathrm{t}}_{b} = \mathbf{w}^{\mathrm{r}}_{b,\hat{i},\hat{s}}$ where $\hat{i}, \hat{s} = \operatorname{argmax}_{\, i \in \mathcal{I}_b, s \in \mathcal{S}} C^{\alpha}(\omega^{\mathrm{t}}_{b,i}, \omega^{\mathrm{r}}_{b,i,s})$.

The final process is the helper joint transfer. The current implementation clones the helper joints and their procedural controllers from the reference without any modifications. Automated modification of helper joint controllers should be explored in future research for more semantically accurate rig transfer.

4. Experiments

The proposed algorithm was implemented as a C++ plugin of Autodesk Maya 2020. The geodesic voxel binding method built in Maya 2020 was used to generate the guide weights. In all the experiments, the fall-off parameter α , number of rays S, and cone angle θ was set to 3.0, 128, and 30 degrees, respectively. The maximum number of influencing joints K was set to four. Because the surface size affects the angle and quantity of ray-casting, the target character was scaled to be as close to the reference character as possible. A desktop PC with Intel(R) Core(TM) i9-10920X CPU @ 3.50GHz was used for the experiments. The experimental assets are shown in Figure 3. The statistics of the assets, including the computation time of the skin weight transfer, are summarized in Table 1. The measured time accounts for the combined duration of the mesh correspondence and weight interpolation processes.

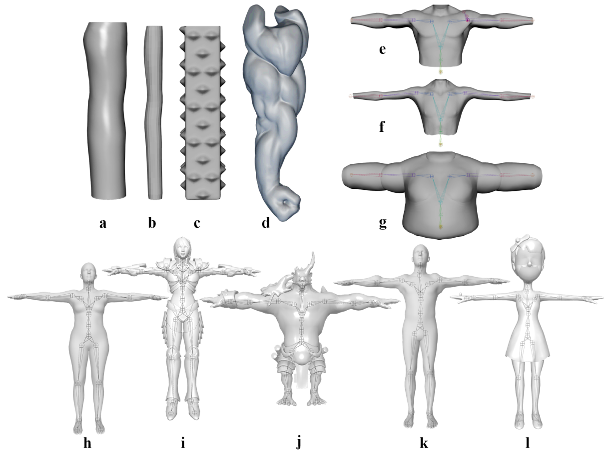

Figure 3: *Experimental assets. Skin weights and helper joints of reference models (a), (e), (h), and (k) were transferred to the target models (b)-(d), (f)-(g), (i)-(j) and (l), respectively.*

Table 1: *Statistics of the experiments. The reference models are indicated in bold font.*

	#verts	#faces	#rays	time [s]
a. **Arm model**	622	1240	–	–
b. Slender arm	1010	2016	169088	2.676
c. Throne covered	7839	15612	1810816	22.492
d. Muscular arm	15768	31531	3309312	43.241
e. **Half body**	1811	3534	–	–
f. Slender body	56634	56400	1384064	53.763
g. Fat body	7216	14360	5094528	196.793
h. **SMPL model 1**	6890	13776	–	–
i. Maria	8094	14566	7755520	1204.650
j. Warrok	6557	12626	6328704	982.429
k. **SMPL model 2**	6890	13776	–	–
l. Claire	4224	8181	4069888	671.576

First, we validated the skinning quality of our method by comparing with the Maya copy weight tool and the geodesic voxel skinning method using simple two-link arm models, as shown in Figure 3(a)-(d). The reference arm (a) depicted muscle expansions and wrinkles around the elbow by six helper joints. The skin weights and helper joints were optimized to three different models (b)-(d) using the three methods. As shown in Figure 4, the geodesic voxel binding method failed to generate skin weights of the helper joints, which resulted in a lack of wrinkles at the joints. The Maya copy weight tool caused unnatural compression and breakage in all results. By contrast, the proposed method successfully produced a muscle contraction effect in slender arms. It retained skin protrusions while preserving the muscle contractions and wrinkles of the reference model.

The proposed method was applied for an upper body model as shown in Figure 3(e)-(g), which included muscle stretching in the arms, waist, chest, and shoulders by swinging the arms back and forth. The proposed method successfully reproduced the muscle be-

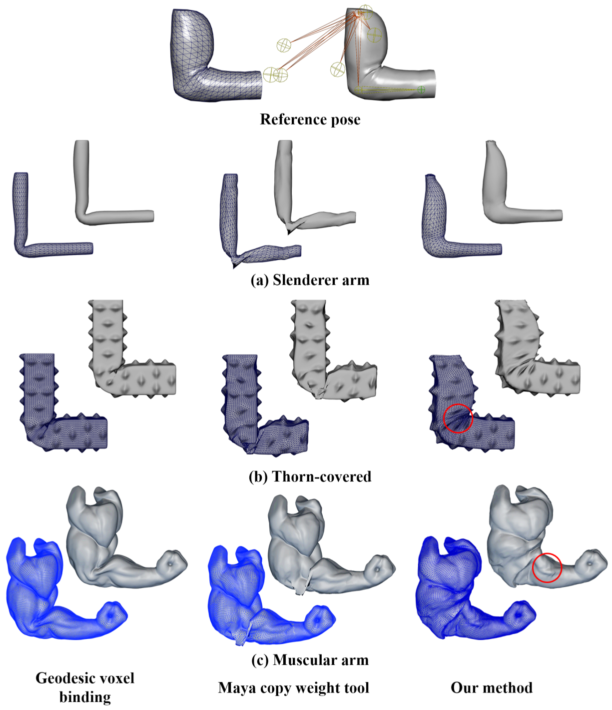

Figure 4: *Experimental results of the arm models.*

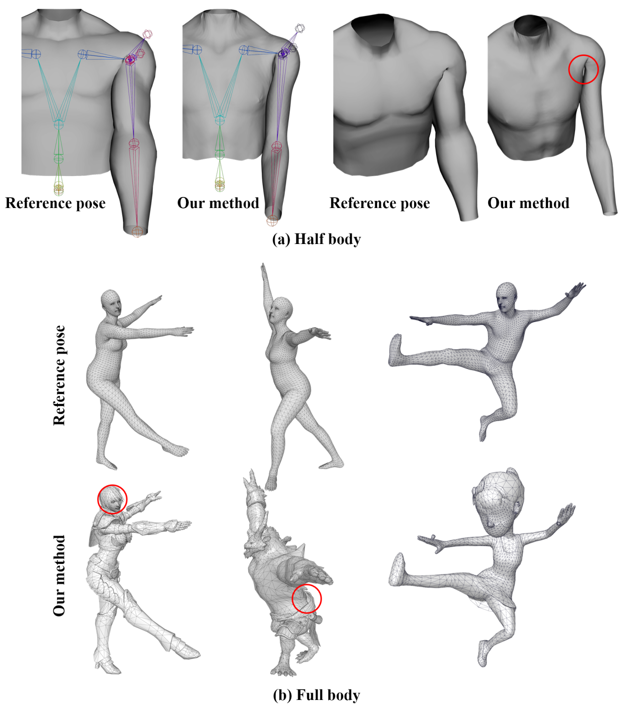

Figure 5: *Experimental results of the character models.*

havior of the reference, especially for the arm and armpit muscles when the arms are naturally lowered, as shown in Figure 5(a)

We also performed the skin weight transfer for the SMPL model [LMR*15] as shown in Figure 3(h) and (k). Using a dance animation composed of continuous complex movements, the SMPL model (h) was first approximated using six helper joints through the example-based rigging method [Muk15]. The resulting helper joints and skin weights were transferred to two different characters, as shown in Figure 3(i) and (j), respectively. On the other hand, no helper joint was used for the SMPL model (k), and only the skin weights were transferred to character (l). Our method successfully provided smooth skin deformation on the armor and body and exhibited the posture of the SMPL character more naturally.

5. Conclusions

We have proposed a geometric approach for transferring the skin weight of helper bone rigs between human-like characters. The skeleton-aware method combines the similarity evaluation of the guide weights and distributed ray-casting to stably establish skin mesh correspondence. This method offers an automated rig transfer for expressive skin deformations without any manually specifying additional constraints for surface correspondence. In future work, we will improve the transfer quality in densely branched and complex joint areas where the current algorithm tends to fail. Moreover, the surface correspondence based on the distributed ray-casting cannot deal with highly detailed skin geometries, and higher computational cost is required as the number of skin vertices increases. We should investigate more efficient algorithms and implementation for practical use.

Acknowledgements This work was supported by PlatinumGames Inc. and JSPS KAKENHI Grant Numbers 21K12193.

References

[AGR*16] Avril, Quentin, Ghafourzadeh, Donya, Ramachandran, Srinivasan, et al. “Animation Setup Transfer for 3D Characters”. *Computer Graphics Forum* 35.2 (2016), 115–126 2.

[ARMH23] Abdrashitov, Rinat, Raichstat, Kim, Monsen, Jared, and Hill, David. “Robust Skin Weights Transfer via Weight Inpainting”. *SIGGRAPH Asia 2023 Technical Communications*. 2023 2.

[DdL13] Dionne, Olivier and de Lasa, Martin. “Geodesic Voxel Binding for Production Character Meshes”. *SCA ’13: Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2013, 173–180 2.

[Got19] Got, Emeline. “Developement of a New Automatic Skinning Ssystem”. MA thesis. Universitat Politècnica de Catalunya, 2019 2.

[LMR*15] Loper, Matthew, Mahmood, Naureen, Romero, Javier, et al. “SMPL: A Skinned Multi-Person Linear Model”. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34.6 (Oct. 2015), 248:1–248:16 4.

[Muk15] Mukai, Tomohiko. “Building Helper Bone Rigs from Examples”. *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games*. 2015, 77–84. isbn: 9781450333924 2, 4.

[Par05] Parks, Jason. “Helper Joints: Advanced Deformations on Runtime Characters”. *Game Developers Conference*. 2005 1.

[YKKL16] Yoon, Seung-Hyun, Kim, Taejoon, Kim, Ho-Won, and Lee, Jieun. “Transferring Skin Weights to 3D Scanned Clothes”. *ETRI Journal* 38.6 (2016), 1095–1103 2.