Rig-Space Motion Retargeting

Lin-Chuan Hu, Ming-Hsu Chang and Yung-Yu Chuang[†]

National Taiwan University

Abstract

This paper presents a framework for transferring rig parameters from a source animation to a target model, allowing artists to further refine and adjust the animation. Most previous methods only transfer animations to meshes or joint parameters. However, in industry, character animations are usually manipulated by rigs. Thus, it is difficult for artists to work further on the retargeted animations. Our method first applies motion transfer to deform the target model to mimic the source motion. Next, we estimate the rig parameters which satisfy the following properties: (1) the resultant animation resembles the retargeted animation and (2) the rig parameters match the artist's editing conventions. Artists could refine the produced rig parameters and the edits are propagated throughout the whole animation.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Character animations play an important role in the making of animation films. Their production is however very tedious and time consuming as it requires artists to tweak many parameters to generate natural-looking motions. Motion retargeting, transferring an animation of the source model to the target model, leverages existing animations for easing production of similar ones. It is especially useful for minor characters as their motions are less noticeable and minor artifacts can be tolerated. Most motion retargeting methods operate on meshes and joints of skeletons (the only exception is Hahn *et al.*'s method [HMT*12]). Thus, even though they could produce similar animations for different characters, the results are difficult for artists to edit.

In industry, characters are often animated by rigs. A rig often consists of a large number of controllers. Figure 1 gives an example of a character, Nezha, with 85 controllers and 603 rig parameters. Each controller can manipulate several joints and vertices by a complex many-to-many mapping function. The mapping function is operated by multiple parameters defined with different scales, ranges, meanings and constraints for artists to manipulate with. The many-to-many characteristic (a joint or a vertex is influenced by multiple controllers) and the intricate hierarchical relationship between controllers make manipulation in the rig space very challenging. This paper proposes a motion retargeting system which generates editable rig parameters so that the retargeted animation can be further edited.

[†] This project was partly funded by MOST104-2628-E-002-003-MY3, MOST104-2218-E-002-015 and TWR Entertainment, Inc.

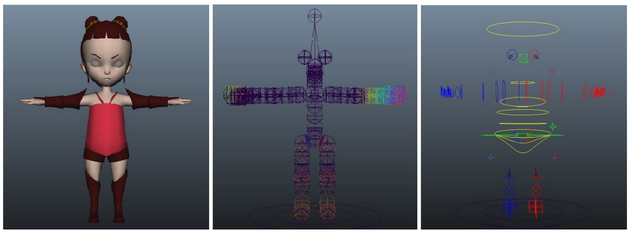

Figure 1: *A rig example. From the left to the right are the model, the skeleton and its 85 controllers.*

2. Rig-space motion retargeting

The input to our system contains the source model S, the target model T and the source animation we want to transfer. Note that the model here refers to the complete model including the mesh and rigs (including controllers and the skeleton) of the model. The input source animation is represented by the rig parameters $\Gamma^S = \{\gamma_1^S, \gamma_2^S, \ldots, \gamma_F^S\}$ of the source model, where γ_f^S is the rig parameter for the f-th frame and F is the number of frames in the animation. In addition to those models and parameters, we also require users to specify the rig parameters γ_1^T of the target model for the first frame.

With these inputs, our system first performs a motion transfer step to deform the target model so that it perform similar motions specified by the source animation. Next, we attempt to recover the rig parameters from the deformed target meshes with two properties: motion resemblance and parameter smoothness. Figure 2 illustrates the process. If the artist would like to refine/tweak param-

DOI: 10.2312/egp.20161039

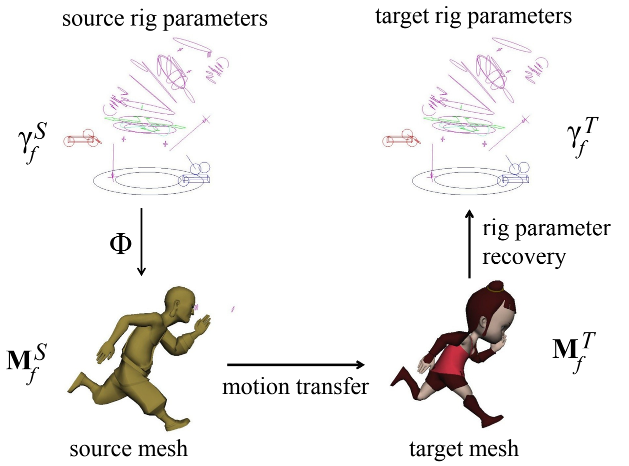

Figure 2: *The core idea of the proposed approach. To transfer the rig parameters of the source model to the target model, we first transfer the motion of the source animation to the target model so that the target model performs the similar motion as specified in the source animation by the* motion transfer *step. Next, given the desired mesh of the target model at each frame, we figure out the proper rig parameters which would result in a mesh similar to the desired mesh using the* rig parameter recovery *step.*

eters, he/she can edit rig parameters at selected keyframes and the edits will be propagated to the whole animation through the motion refinement process. Note that he/she can refine the animation by conventional rigging systems because the resultant rig parameters are intuitive to edit and match his/her own editing conventions.

Motion transfer. In the first step, we transfer the motion of the source animation to the target model so that the target model performs the similar motion as specified in the source animation. We adopt the framework of mesh-based deformation transfer to obtain better stability and performance [SP04]. Because of the intricate hierarchical relationship of the model skeletons used in industry, we adapt the deformation transfer method for multi-component objects proposed by Zhou *et al.* [ZXTD10]. Their method is for mesh deformation and we extend it with a temporal smoothness term since our problem deals with animations.

Rig parameter recovery. Given the mesh animation $\widetilde{\mathbf{M}}_f^T$ from the motion transfer step, this step figures out the proper rig parameters which would produce a mesh similar to the desired mesh. That is, it finds out a set of rig parameters $\Gamma^T = \{\gamma_1^T, \gamma_2^T, \ldots, \gamma_F^T\}$ for the target model so that the resultant meshes resemble $\widetilde{\mathbf{M}}_f^T$.

To handle the complex rig space and various rigging systems, similar to Hahn *et al.* [HMT*12], we treat the entire rig mapping process Φ as a black box. For the target model T, given the rig parameter γ_f^T at the frame f, the black box returns the corresponding mesh

$$\mathbf{M}_f^T = \Phi(T, \gamma_f^T). \tag{1}$$

By hiding dozens of complex non-linear rig mapping functions behind the black box, our approach is versatile and can be used with

Figure 3: *A motion retargeting example with a walking animation. (top: source. bottom: target.)*

various rigging systems. In this paper, we use Maya as the black box to obtain the corresponding meshes.

We estimate the rig parameters γ_f for the target model at the frame f by minimizing the following energy function:

$$E_P(\gamma_f) = E_m(\gamma_f) + w_u E_u(\gamma_f) + w_s E_s(\gamma_f), \tag{2}$$

where E_m, E_u, E_s are respectively the energy terms for maintaining mesh similarity to the desired mesh, undeformability of model structure and temporal coherence of the mapped target meshes.

3. Results and conclusions

Figure 3 shows a motion retargeting example with a walking animation. The source model, Arhat, has 77 controllers and 610 rig parameters. The proposed system successfully transfers its walking animation to the Nezha model. The artists can tweak parameters to enrich their styles or to fix errors introduced during the optimization processes. Our system allows users to further refine the animation by editing rig parameters at selected frames using controllers in their favorite rigging system. To reduce time and effort, the refinement will be automatically propagated to the whole animation through optimization.

This paper proposes a system that allows animators to obtain editable rig parameters for motion retargeting. Our method is compatible with production pipeline and can be used with standard animation packages.

References

[HMT*12] Hahn F., Martin S., Thomaszewski B., Sumner R. W., Coros S., Gross M. H.: Rig-space physics. *ACM Trans. Graph. 31*, 4 (2012), 72. 1, 2

[SP04] Sumner R. W., Popovic J.: Deformation transfer for triangle meshes. *ACM Trans. Graph. 23*, 3 (2004), 399–405. 2

[ZXTD10] Zhou K., Xu W., Tong Y., Desbrun M.: Deformation transfer to multi-component objects. *Comput. Graph. Forum 29*, 2 (2010), 319–325. 2