Nat. Volatiles & Essent. Oils, 2021; 8(3): 182-191
https://doi.org/10.52783/nveo.5496

Analysis Of Motion Retargeting Techniques For Character Animation Transfer

Swati Atreya

Asst. Professor, Department of Animation and Gaming, Graphic Era Hill University, Dehradun Uttarakhand India

Abstract

Transferring motion from a source character to a target character is known as "motion retargeting," and it is a crucial operation in character animation. It is essential to many applications, including computer-generated movies, virtual reality, and video games. This study examines the benefits, drawbacks, and prospective uses of motion retargeting algorithms for character animation transfer. The examination covers a variety of motion retargeting strategies, including deep learning-based techniques, motion graphs, forward kinematics, and inverse kinematics. Each method is rated according to how well it maintains the desirable aspects of the source motion while modifying it to fit the skeletal structure and dimensions of the target character. The examination also looks at the various strategies computing effectiveness and usability while taking into account the demands of real-time applications and user interaction. It also looks at the difficulties and unanswered concerns surrounding motion retargeting, including how to handle complicated character deformations, manage touch and interaction, and enhance generalisation among characters with various anatomical variations. For academics and industry professionals involved in character animation, the analyses conclusions offer useful insights. Developers can select the most effective solution for their unique requirements by considering the advantages and disadvantages of different motion retargeting strategies. The paper present prospective research avenues that may be pursued in order to overcome current obstacles and develop the state-of-the-art in motion retargeting for character animation transfer. This paper present methodology offers a versatile and effective motion retargeting strategy, which advances character animation. It helps animators and developers save time and effort by enabling them to quickly prototype animations for characters with different morphologies. Our tool's automatic features guarantee a streamlined workflow, allowing users to concentrate on the creative aspects rather than the motion retargeting's technical complexities.

I. Introduction

In the field of character animation, the automatic synthesis of motion for articulated characters continues to be a difficult subject. Although motion capture technology has somewhat made it easier to manually create expressive and organic human motion, there aren't many good authoring tools available for modifying already existing motion to fit new circumstances or various characters. Utilizing previously collected motion capture data requires the use of motion retargeting. Even when characters have varied body types or bone lengths, it addresses the issue of motion transference [1]. Despite having a comparable skeleton structure, directly mapping motion onto a different virtual character can result in unnatural motion without the necessary retargeting techniques. Existing motion retargeting techniques mainly concentrate on applying stylistic motion to comparable characters or demand additional authoring tools

from animators in order to apply motion to a wider variety of characters. These approaches might not be appropriate in situations that call for a great degree of adaptability and versatility. It is essential to create effective and automatic motion retargeting algorithms to address these issues. Such methods would allow animators to make use of the motion capture data already available without being constrained to particular character types or depending on human tweaks. The animation community will gain from more adaptable and user-friendly motion retargeting technologies by having access to a greater variety of character animations for use in a variety of applications[2].

Virtual reality, computer-generated movies, video games, and other applications all depend on character animation. It takes a lot of time and effort to give characters realistic, expressive motion, and animators frequently put a lot of work into it. Motion retargeting algorithms have become a valuable tool for transferring motion between characters with various morphologies, which lessens this strain. Motion retargeting is the process of transferring motion from one character to another while maintaining the key elements of the original motion and taking into account variations in skeletal structure, dimensions, and other physical features. By using pre-animated motions or motion capture data that already exists and applying it to various characters, animators can streamline the animation pipeline and save time. Motion retargeting's major objective is to provide outcomes that are natural and aesthetically acceptable, making sure that the retargeted motion is in line with the target character's body and preserves the intended behaviour. To[8] address this issue, a number of strategies have been developed, ranging from conventional inverse kinematics and forward kinematics methods to more sophisticated strategies utilising motion graphs and deep learning-based approaches. By efficiently mapping the motion from one skeleton to another, inverse kinematics techniques strive to change the joint angles of the target character to match those of the source character. On the other hand, forward kinematics propagates the transformations from the source character's root joint to the target character, resulting in a retargeted motion. Motion graphs interpolate or blend recorded motions from a database. The goal of this study is to provide a thorough overview of the benefits, drawbacks, and potential applications of various motion retargeting techniques for character animation transfer. This analysis seeks to help researchers and practitioners choose the best appropriate methodology for their unique needs by analysing the advantages and disadvantages of each technique. The investigation also clarifies ongoing issues and unanswered problems, paving the way for future developments in the field of character animation transfer via motion retargeting[10].

II. Related Work

The field of computer graphics and animation has intensively researched motion retargeting methods for character animation transfer. To overcome the difficulties of transmitting motion between characters with various morphologies, researchers have put forth a variety of strategies. An overview of related work in the field of motion retargeting is given in this section.

Inverse kinematics (IK), which focuses on matching the joint angles of the source character to those of the target character, is one class of approaches that is frequently utilised. IK problem for motion retargeting

has been solved using algorithms like Jacobian transposition, Jacobian pseudo-inverse, and CCD (Cyclic Coordinate Descent). Although these techniques offer precise joint angle mapping, they may have trouble dealing with complex character deformations and contact conditions. An alternative method uses forward kinematics (FK), which propagates the transformations from the root joint of the source character to the target character. FK-based techniques are straightforward and computationally efficient, but they may have trouble retaining the original motion's aesthetic and nuanced details during the retargeting process[3].

Motion retargeting has also been extensively studied using motion graphs. Motion graphs allow interpolation, blending, and sampling to create new motions for the target character by constructing a graph structure that represents a collection of previously recorded motions. These techniques excel in maintaining motion style and offering a wide range of motion possibilities, but for best results, they may need a sizable motion database. Motion retargeting has lately become more popular because to deep learning-based approaches. The mapping between source and target motions can be learned using neural networks, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs). These methods show that it is possible to record intricate motion correlations and produce effective retargeting outcomes. They might, however, need a lot of training data and computer power. The retargeting problem has also been studied using optimization-based techniques, which aim to discover the greatest fit between source and target motions while taking into account a variety of restrictions, such as joint limits and physical plausibility[5].

a) Skeleton Extraction for Animation:
   In animation, the technique of autonomously creating a skeletal structure from a given character or object is known as "skeleton extraction." To identify the underlying joint hierarchy and bone connections, it includes analysing the geometry and motion of the model. For a variety of animation tasks, such as rigging, motion retargeting, and character control, this extracted skeleton serves as the basis. The animation workflow has been substantially improved by improvements in skeleton extraction techniques, which offer an automated and dependable way to create and manipulate articulated characters.
b) Volumetric Method for Extraction:
   The underlying skeletal structure of a character or object is estimated using volumetric representations, such as voxel grids or signed distance fields, in volumetric approaches for skeleton extraction in animation. The following steps are often included in these techniques:
   i. Voxelization: The process of discretizing a 3D space into a grid of voxels to depict a character or item as a volumetric representation. Whether or not the voxels intersect the surface geometry determines their values.
   ii. Skeletonization: The process of extracting the skeleton structure from the volumetric representation. This is accomplished by locating the voxel grid's medial axes or centerlines, which roughly correlate to the locations of the skeletal branches.
   iii. Skeleton Refinement: To increase accuracy and anatomical realism, the extracted skeleton is further polished. This could entail post-processing techniques like branch pruning, skeleton structure smoothing, or applying geometric constraints.

c) Geometric methods for skeleton extraction:

Utilising the surface geometry to infer the underlying skeletal structure, geometric approaches for skeleton extraction provide a simple and straightforward way. They work well for characters who have distinct and articulated surface features. However, they could have trouble capturing complex or ambiguous geometries effectively, and they might need extra steps to deal with situations like self-intersections or imperfect or noisy surface data [4].

i. Mesh Segmentation: Based on geometric or topological properties, the character's surface mesh is divided into areas or components. This makes it easier to distinguish between diverse body parts or areas that correspond to various skeletal sections.

ii. Geometric parameters like curvature, local extreme, or crucial points are used to identify junctions or significant spots within each divided region. These joints act as the skeletal structure's anchor points.

iii. The bones or limb segments connecting the joints are then assigned based on proximity and connection after the joints have been identified. The character's basic skeleton is established at this step.

iv. Skeleton Refinement: To increase accuracy and anatomical realism, the original skeleton structure is polished. This could entail post-processing techniques like smoothing the connections between the bones.

III. Algorithm Overview

Character animation research has focused extensively on the development or adaption of a deformation skeleton from a mesh. Numerous techniques have been created, some of which rely on the game Spore's implicit definition of the skeleton. However, the methods that are dependent on input from a general static mesh will be the focus of this review. These approaches—which are frequently utilised in animation are more suitable for achieving the purpose of the technique that is being described[14].

Similar issues like joint location must be resolved when creating or retargeting a skeleton. As a result, the methods used for both tasks are frequently similar.

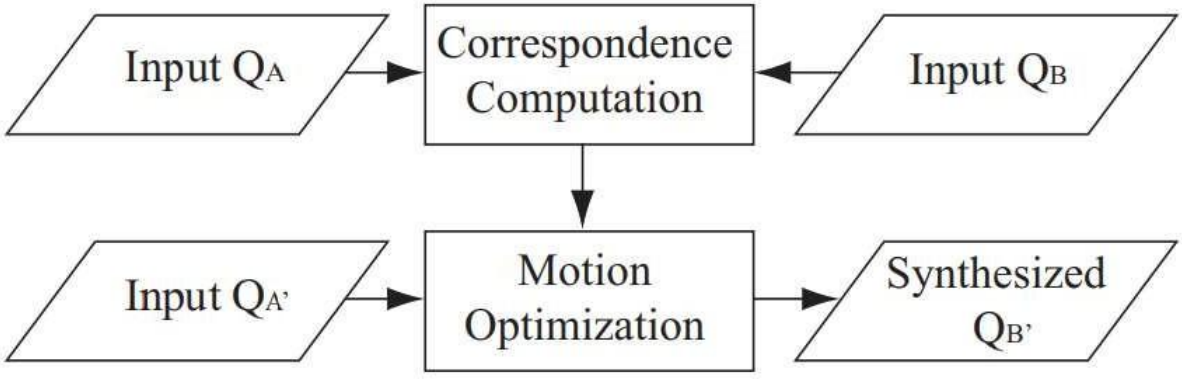


Figure 1: Step wise block diagram of algorithm

The motion transfer algorithm for creating new motion for character CB utilising input motions is shown in Figure 1. There are two main steps in the algorithm:

- Correspondence Computation: Establishing correspondence between the two characters based on the motion of the point cloud on the characters in the example motions QA and QB is the task of the correspondence computation stage. The programme looks at how the characters move and pinpoints any connecting spots or places. For mapping the motion from the source character to the destination character, this correspondence information is essential.
- Motion Transfer Optimisation: After the characters' correspondence is determined, the algorithm moves on to motion transfer optimisation. The method seeks to identify the suitable coordinations or transformations for character CB that mimic the behaviour of QA' given a new motion, QA'. It provides a solution for the key variables that must be changed for character CB to behave in the same way as QA.

An overview of a motion retargeting algorithm for character animation transfer is given below:

Step 1: Input: The source motion data for the source character and the target character, including joint angles, positions, or keyframe animation, are provided as input to the algorithm.

Step 2: Skeleton Mapping: Using the joints of the source and target characters as a starting point, the algorithm creates a mapping. This mapping specifies how the joints of the two characters relate to one another, enabling motion transfer.

Step 3: Pose Alignment: The algorithm matches the root locations and orientations of the starting poses of the source and target characters. This process reduces any early inconsistencies between the characters.

Step 4: Joint Transformation: To retarget the motion from the source to the target character, the algorithm uses joint transformations. On the basis of the mapping, this often entails computing the transformation matrices or quaternions for each joint else repeat step 2.

Step 5: Motion Adaptation: The algorithm modifies the target character's unique traits and constraints in order to fit the source character's motion. To preserve realism and physical plausibility, this may entail changing joint angles, velocities, or accelerations.

Step 6: Blending and Smoothing: To remove jitter or abrupt changes, the system applies smoothing techniques to the retargeted motion. In order to interpolate between keyframes or transition between motions smoothly, blending techniques may also be used.

Step 7: Output: The algorithm generates the target character's retargeted motion data, which may include joint angles, positions, or keyframe animation.

Nat. Volatiles & Essent. Oils, 2021; 8(3): 182-191
https://doi.org/10.52783/nveo.5496

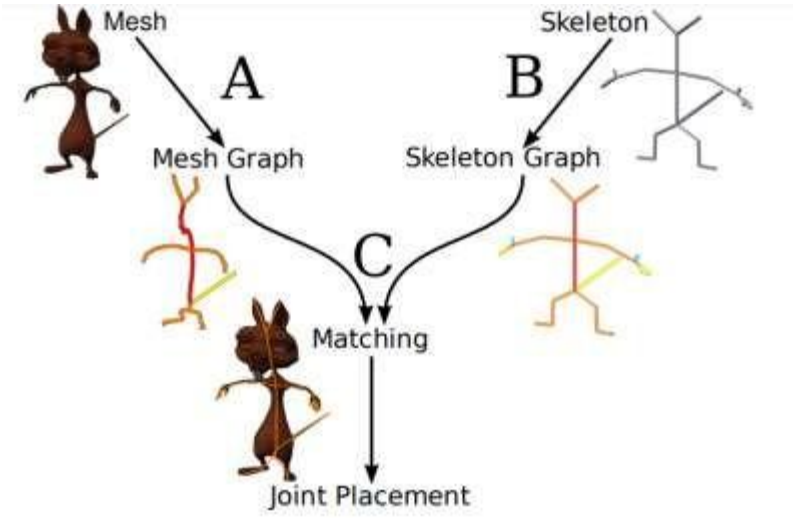


Figure 2: An overview of working Algorithm for motion retargeting

The motion optimisation procedure makes use of the correspondence information gathered in the first stage. The algorithm does, however, allow for manual determination of the correlation in cases where motion QB is not available or the correspondence cannot be computed precisely. Due to this flexibility, animators can, if necessary, manually establish the mapping between the source and target characters[17].

The algorithm allows for the synthesis of new motion QB' for character CB that mimics a specific motion QA' for character CA by carrying out these two phases, even if the characters have distinct morphologies or looks. With the use of this method, current motion may be automatically and generically retargeted to a variety of character motions, allowing for quick prototyping and the transfer of motion from existing characters to new ones.

IV. Implementation

The open-source programme Blender was used to process the mesh and skeleton for the motion retargeting technique. All of the experiments used Blender's automatic skinning function, which uses bone heat equilibrium to attach a skeleton to a mesh.

A sparse matrix direct solver was used to solve the harmonic equation for the weights of the nodes in the Reeb graph. The OpenNL library specifically made use of the SuperLU solver. The Reeb graph nodes' weights are effectively computed by this solution, allowing for the extraction of useful skeletal data. It should be noted that embedding problems, such as arcs that occasionally pierce the mesh surface, may emerge from the algorithm's present Reeb graph creation mechanism. However, solutions to these concerns have been created, and using one of the established solutions [1, 13] can successfully fix these embedding problems.

It's vital to understand that during the optimisation process, we do not transfer the root translation. As a result, we translate QB' using the translation from the motion QA'. However, depending on the dimensions of the characters involved in the motion transfer, this root translation might be scaled.

Table 1: Comparison with Existing methods

Method	Complexity	Skeletons	Control Bones	Joint Placement	Filtering	Topology Independence	Orientation Independence
Volumetric [12]	Simple	Skeletons may be simple	No	Fixed weights	N/A	Fully independent	No
Geometric [13]	Skeletons	Biped/Quadruped templates	No	Fixed template	Single threshold	No	Yes
Geometric [15]	Full-featured	Professional skeletons	Yes	Controlable and balanced weights	Multiresolution	No	Yes

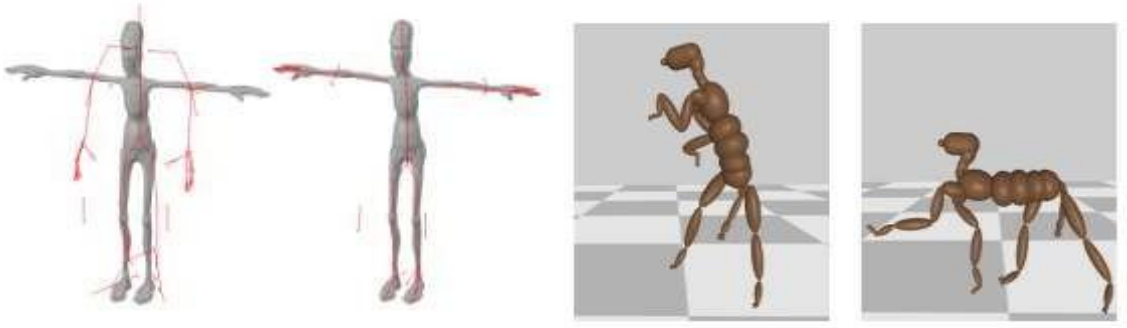
The geometric approach, on the other hand, offers more flexibility and control. To create more intricate and realistic skeletons, skeletonization is performed using biped or quadruped templates. Control bones allow for the creation of fully functional professional skeletons with excellent animation control. The chosen template determines how the joints are placed, however the weights can be changed to provide control and balance. Multiresolution methods are one type of filtering technique that can be used. Orientation independence is promoted, however topological independence is not entirely accomplished.

These qualities show the complexity, skeleton types, control bones, joint location, filtering, and independence in topology and orientation differences between volumetric and geometric techniques.

V. Result and Discussion

In order to solve the harmonic equation for Reeb graph node weights, the implementation made use of Blender's mesh and skeleton processing tools, including automated skinning, and the SuperLU solver via the OpenNL library. The programme recognises potential embedding issues that can arise during the creation of Reeb graphs and emphasises the existence of well-established fixes for these problems. We show the usefulness of our automatic motion transfer procedures in the outcomes of our experiments. We used SNOPT [GSM96] to tackle the space-time optimisation challenge presented in motion retargeting. All of the samples in this section's synthesis took less than a minute. The "march" motion in

the animation was transferred from a child model to a dog figure using the motion retargeting method. The motion transfer's outcome showed that it was possible to create fresh motion for the dog character that mimicked the walking gait of the kid as shown figure 3. It was successful in transferring the child march motion, which is characterised by the child's leg movements and general pace, to the dog character. Although there were modifications made to account for the anatomical variations between the child and the dog, the retargeted motion displayed similar walking behaviour to that of the youngster.



(A) (B)

Figure 3: (A) Child March Pose (B) Dog March Pose

The resulting dog march animation showed how the dog's legs moved in a way that seemed like it was marching, matching the child's pace. The translated motion was modified to fit the dog's body shape and skeletal structure while keeping the essential elements of the child's march, resulting in a natural and convincing animation. Without explicit user assistance or the establishment of correspondences between the child and the dog, the system was able to automatically synthesise motion for the dog character using the motion retargeting method. The algorithm's general nature makes it possible to transfer motion between characters with various morphologies, allowing for quick prototyping and animation adaptation for various character kinds.

VI. Conclusion

This investigation concentrated on character animation transfer methods using motion retargeting. The objective was to provide a generic, automatic system for transferring motion across characters with varied morphologies without the use of physical labour or defined character correspondences. In this paper, we presented a method with two basic steps: motion optimisation and correspondence computation. In the computation of correspondence, characters were matched together based on how point clouds moved during model motions. In order to recreate the behaviour of the source motion on the target character, the motion optimisation step solved for the crucial coordinations. The outcomes of our tests showed how successful the motion transfer method is. We were successful in synthesising new motion for the target character that accurately replicated the motion of the source character using the methods provided here. The programme accomplished this transfer while taking into account the variations in character morphologies, producing animations that are both realistic and aesthetically beautiful. For mesh and skeleton processing, Blender was used as part of the algorithm's implementation. A sparse matrix direct

solver was used to address the optimisation issue. The whole synthesis time for the instances given was below one minute, demonstrating the algorithm's effectiveness. This study offers a generic, automatic motion retargeting method, which advances the field of character animation. The programme reduces the manual work necessary for motion development by enabling rapid prototyping and adapting current motion to new characters. With the help of this research, character animation could become more effective and imaginative in a variety of contexts, such as video games, movies, and virtual reality experiences.

References:

[1] YE Y., LIU C. K.: Animating responsive characters with dynamic constraints in near-unactuated coordinates. ACM Trans. Graph. (SIGGRAPH Asia) 27, 5 (2008), 1–5.

[2] G. Aujay, F. Hetroy, F. Lazarus, and C. Depraz. Harmonic skeleton ´ for realistic character animation. In SCA ’07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 151–160, 2007.

[3] I. Baran and J. Popovic. Automatic rigging and animation of 3D char- ´ acters. ACM Transactions on Graphics, 26(3):72, 2007.

[4] Blender-Foundation. Blender 3D, 2008. www.blender.org.

[5] D. Brunner and G. Brunnett. Fast force field approximation and its application to skeletonization of discrete 3D objects. Computer Graphics Forum, Vol. 27, No. 2 (2008), Eurographics 2008, pages 261–270, 2008.

[6] N. D. Cornea, D. Silver, and P. Min. Curve-skeleton properties, applications, and algorithms. IEEE Transactions on Visualization and Computer Graphics, 13(3):530–548, 2007. Member-Deborah Silver.

[7] F. Dellas, L. Moccozet, N. Magnenat-Thalmann, M. Mortara, G. Patane, M. Spagnuolo, and B. Falcidieno. Knowledge-based ex- ¨ traction of control skeletons for animation. In SMI ’07: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2007, pages 51–60, 2007.

[8] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu. A supernodal approach to sparse partial pivoting. SIAM Journal on Matrix Analysis and Applications, 20(3):720–755, 1999.

[9] C. Hecker, B. Raabe, R. W. Enslow, J. DeWeese, J. Maynard, and K. van Prooijen. Real-time motion retargeting to highly varied usercreated morphologies. In SIGGRAPH ’08: ACM SIGGRAPH 2008 Papers, pages 1–11, 2008.

[10] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In SIGGRAPH01, pages 203–212, 2001.

[11] S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. In SIGGRAPH ’03: ACM SIGGRAPH 2003 Papers, pages 954–961, 2003.

[12] B. Levy. OpenNL: Open numerical library. alice.loria.fr/index.php/software/4-library/23-opennl.html.

[13] V. Pascucci, G. Scorzelli, P.-T. Bremer, and A. Mascarenhas. Robust on-line computation of reeb graphs: simplicity and speed. ACM Transactions on Graphics, 26(3):58, 2007.

[14] M. Poirier and E. Paquette. Rig retargeting for 3d animation - video, 2009.

[15] L. Shapira, A. Shamir, and D. Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. The Visual Computer, 24(4):249–259, 2008.

[16] J. Tierny, J.-P. Vandeborre, and M. Daoudi. Enhancing 3D mesh topological skeletons with discrete contour constrictions. The Visual Computer, 24(3):155–172, 2008.

[17] L. Wade and R. E. Parent. Fast, fully-automated generation of control skeletons for use in animation. In CA ’00: Proceedings of the Computer Animation, page 164. IEEE Computer Society, 2000.