Received: 2 May 2021 | Accepted: 6 May 2021

DOI: 10.1002/cav.2015

SPECIAL ISSUE PAPER

WILEY

Interactive multi-character motion retargeting

Jongmin Kim[1] | **Yeongho Seol**[2] | **Taesoo Kwon**[3]

[1]Computer Science and Engineering, Kangwon National University, Chuncheon, South Korea

[2]NVIDIA, Seoul, South Korea

[3]Department of Computer and Software, Hanyang University, Seoul, South Korea

Correspondence
Taesoo Kwon, Department of Computer and Software, Hanyang University, Seoul, South Korea.
Email: taesoobear@gmail.com

Abstract

A motion retargeting process is necessary as the body size and proportion of the actors are generally different from those of the target characters. However, the original spatial relationship between the multiple characters and the environment is easily broken when using previous motion retargeting methods, which are generally performed for each character independently. Therefore, time-consuming manual adjustments by animators are usually required to obtain satisfactory results. To address these issues, we present a novel multicharacter motion retargeting method that preserves various types of spatial relationships between characters and environments. We establish a unified deformation-based framework for the motion retargeting of multiple characters (more than two) or nonhuman characters with complex interactions. Also, an interactive motion editing interface with immediate feedback to the user is provided. We experimentally show that our method achieves a speedup when compared with previous motion retargeting methods.

KEYWORDS

human motion, mesh deformation, motion retargeting

1 | INTRODUCTION

Recently, motion capture systems are widely used by many visual effect and game companies to obtain realistic motion; however, the body size and proportion of the actors are generally different from those of the target characters. Therefore, a motion retargeting process is necessary. Capturing multiple characters with complex interaction has also been frequently attempted in many feature films and video games, specifically in relation to battle scenes, team sports, and choreography. However, motion retargeting is normally performed for each character independently. As a result, the original spatial relationship between the characters and the environment is easily broken during this process. Time-consuming manual adjustments by animators are necessary to obtain satisfactory results.

In this paper, we present a novel multi-character motion retargeting method that preserves various types of spatial relationships between characters and environments. Unlike conventional approaches that employ skeleton models and nonlinear optimization to compute the optimal joint angles, we embed a deformable graph that consists of a sparse set of vertices into each character and undertake motion retargeting in a linear system framework. Our two-step optimization process preserves the original shape of the graph subject to a set of constraints arising from various types of interactions. Our motion retargeting framework uses a combination of ideas from the mesh deformation transfer,[1,2] the modified as-rigid-as-possible (ARAP) mesh deformation process,[3] and cage-based editing.[4] With our unique graph-based representation, various constraints are effectively represented as linear constraints, and the resulting sparse linear system is efficiently solved. Also, one of the key features of our approach is its interactive space-time motion editing scheme. This scheme achieves good computation efficiency for the further manipulation of multiple characters after these characters are retargeted. Our motion editing tool is useful for producing collision-free results because it allows the user to fix glitches conveniently while the main parts of multicharacter animation remain intact.

The proposed motion retargeting method can be applied to various types of motions. Specifically, the method retargets non-human characters with complex joint configurations (e.g., spider, wolf, among others) while preserving the semantics of the original motion and constraints. A substantial performance gain is also obtained during the retargeting process, as our system solves sparse pose graphs simultaneously. The results show that the proposed method can handle complex motions such as a group motion consisting of 19 characters in less than a second, whereas previous methods find it difficult to handle such problems.

The contributions of the paper can be summarized as follows:

- A fast deformation-based framework for the motion retargeting of multiple characters (more than two) or nonhuman characters with interactions
- Interactive motion editing interfaces that provides immediate feedback to the user

2 | RELATED WORK

In this section, we review the previous work on motion retargeting and editing for single and multicharacter animation.

Retargeting the motion of a single human character has frequently been studied in the computer graphics and animation community. Gleicher et al.[5] presented a technique for adapting the motion of one character to another character with different body proportions. This problem was formulated as a constrained optimization problem with space-time constraints. Lee and Shin[6] suggested a hierarchical framework that adaptively adds motion details to satisfy certain constraints by adopting a multilevel B-spline representation scheme. This scheme provided a considerable speedup because the time-consuming space-time optimization process was avoided. Hecker et al.[7] introduced a system for animating a wide range of characters. Our goal is similar to the objectives in these earlier works; however, our focus is on establishing a motion retargeting system that is computationally efficient with regard to interactivity and that works for multiple characters. Our deformation-based approach formulates the motion retargeting problem as a sequence of constrained linear least-squares problems and is thus more efficient than the previous approaches[5,8,9] that relied on nonlinear optimization.

The inverse kinematics (IK) problem is also closely related to our work.[6,10-13] Many numerical IK solvers based on Jacobian matrices have been proposed for finding optimal joint angle displacements for characters to move the end-effectors slightly toward the goal in an iterative manner. However, this is a computationally inefficient process for the character with lots of constraints and complex joint structure. Moreover, how to handle the displacement of each end-effector at every iteration is nontrivial, especially when there are multiple conflicting goals or when multiple characters are involved.

Editing a group of characters closely related to each other is a challenging task in terms of retaining the original relationship, as this involves heavy computations and the need for an intuitive user interface. Several research works have presented editing solutions that formulate the problem in a mesh deformation framework to maintain the semantics observed in interaction scenes. Kwon et al.[14] proposed a system for editing group motions involving multiple characters using a Laplacian-based mesh editing scheme. Kim et al.[15] proposed a framework that uses both the Laplacian motion editing method and discrete motion rearrangement. Using spatial and temporal features, more precise multicharacter interactions, such as when characters carry boxes, were demonstrated. Kim et al.[4] dramatically improved the computational efficiency of these methods using a cage-based deformation technique. While they could edit a large number of characters interactively, the editing was limited to the motion paths, and it was impossible to edit the motion of each body part. This paper extends these works by enabling the intuitive editing of joint movements as well as motion paths.

There have been a few results that shared the similar goal with the proposed work. Liu et al.[9] proposed a physics-based approach for synthesizing multicharacter animation. Represented as a space-time optimization problem with a novel time warp parameterization, a small set of input motions was sufficient to express a variety of multicharacter motions. Ho et al.[16] introduced a mesh for retargeting motions with spatial interactions of a coupe of characters. This was achieved by creating an interaction mesh that connects all of the body joints and that utilizes Laplacian mesh deformation. Although their work demonstrated feasible adaptation of the original motion, the method is difficult to extend to dozens of characters due to their mesh structure and heavy computational cost. Al-Asqhar et al.[17] presented an interactive motion adaptation scheme for characters moving through restricted environments and manipulating objects. Based spatial relationship descriptors, their method is good at adapting postures to nearby geometries but it is likely to get bogged down when having little room for further motion modification during the iterations. Our interfaces conveniently allows the user to edit the retargeted motion in space and time while other part remains intact.

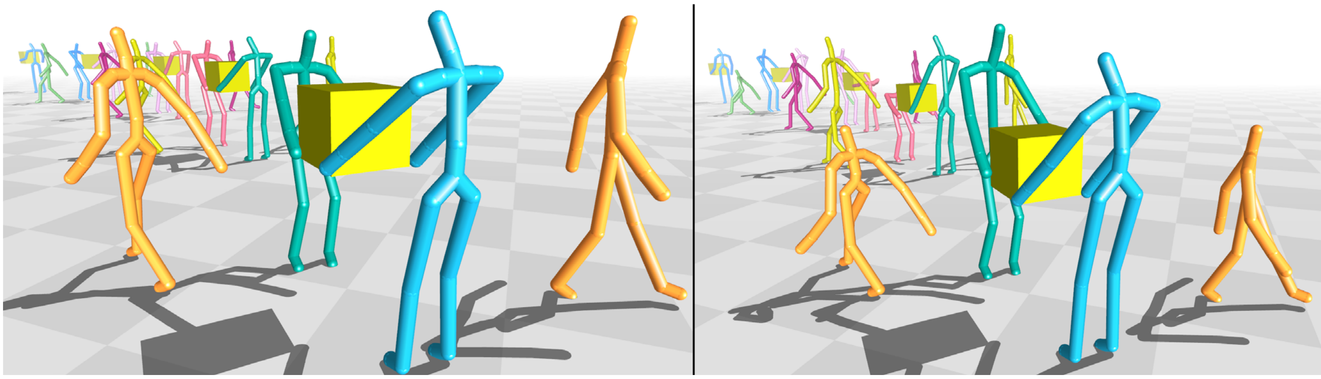

FIGURE 1 In the input motion, multiple characters deliver yellow boxes in a relay manner (left). The proposed method interactively retargets and edits multicharacter animation while maintaining a set of constraints (right)

Recently, deep learning-based motion retargeting methods have been reported and utilize large-scale captured motion data to train the neural networks. Kim et al.[18] introduced a motion retargeting system based on a variational autoencoder with intuitive user interfaces. Villegas et al.[19] proposed an unsupervised motion retargeting system established by neural kinematic networks. Aberman et al.[20] introduced convolutional neural networks-based motion retargeting system with skeleton convolution and pooling operation. These approaches produce a natural retargeted character animation. However, achieving a nonhuman motion retargeting is difficult as it requires large amount training data obtained by motion capture or manual character modeling/editing process. Also ours provides both multicharacter motion retargeting and editing tool.

In this work, we provide a unified framework for the interactive retageting of multiple characters. This is achieved based on a novel graph structure with fewer connections and a linear-constraint formulation. Others have studied how to rearrange precaptured motion segments interactively while preserving the interactions among characters.[21-26] Although a similar group motion editing technique is employed, our focus is on interactive motion retargeting rather than online motion generation. Our work is based on a combination of ideas from the mesh deformation and its transfer.[1,3,4] Our method can efficiently retarget and edit source motions onto any new motion, including nonhuman characters while maintaining various constraints without any visible artifact.

3 | OVERVIEW

Retargeted multicharacter animation is not a final solution as several unintended collisions between characters could inevitably take place. After retargeting motions, a user is capable of easily correcting the glitches by using the proposed motion editing system while the substantial portion of the retargeted result remain intact. Figure 1 shows the overall process of our multicharacter motion retargeting and editing scheme. We generate a pose graph for an individual character in a scene, which then allows novel mesh-based motion retargeting and editing (Section 4). Motion retargeting is performed based on a novel two-step optimization scheme. In the first optimization step, we perform motion retargeting based on the idea of mesh deformation transfer for individual characters. The resulting animation maintains the original motion of the character. In the second optimization step, we apply various constraints identified from the source motion (Section 5) to the intermediate result obtained from the first optimization. The space-time motion editing system is built with the cage-based interface that allows the user to manipulate multiple character and makes the editing process more efficient and intuitive (Section 6).

4 | MULTICHARACTER MOTION REPRESENTATION

In this section, we describe how multi-character animation is represented for our motion retargeting framework. First, we introduce the geometry of the pose graph and how it is generated from a character skeleton structure. Next, we describe a set of constraints to be used for retargeting the multicharacter animation.

4.1 | Pose graph

Positional representation of a character is computationally efficient because the deformation of a 3D character can be formulated as a sparse linear system, whereas angular representations require dense Jacobian matrices. In this sense, we define a deformable pose graph from the joint positions for each character at each frame. Sparse interaction rather than dense one is common in multicharacter animation. Instead of defining a graph structure by connecting all multi-character, the proposed pose graph was created with each character. This approach is helpful for a user to selectively and conveniently establish specific spatial relationships between characters and add/delete the various constraints between non-adjacent joints without breaking the original graph structure. The vertices of the pose graph are extracted from all of the joint positions of the character. The ability to deform a pose graph in three dimension is necessary and it is not achievable without a volumetric structure. Therefore, we create a triangle with three vertices and three edges that interconnect the vertices for each joint, as shown in Figure 2. A triangle that approximately represents the volume of the joint. We perform the aforementioned operations for all joints of the character model, and conduct a quad-rangulation process in a way that three additional vertices connect a pair of triangles that correspond to a pair of adjacent joints.

The shape of the resulting graph resembles the original skeleton; this is different from the previous mesh structures proposed in earlier work,[16] which uses Delaunay tetrahedralization of the input joint positions. Our graph structure allows greater flexibility and a wider range of the character poses compared to the graph in the previous work[16] (see Figure 2). Note that our approach does not create temporal edges that connect the pose graphs between successive frames.

Existing numerical motion retargeting methods typically solve the entire range of an input motion as a single large-scale optimization problem. However, ours can achieve much more computational efficiency as it solves a sparse set of vertices and also retarget the motions to be smooth and continuous. In the proposed method, the problem of smooth space-time motion editing is reduced to the problem of specifying smooth constraints using the mesh deformation scheme.

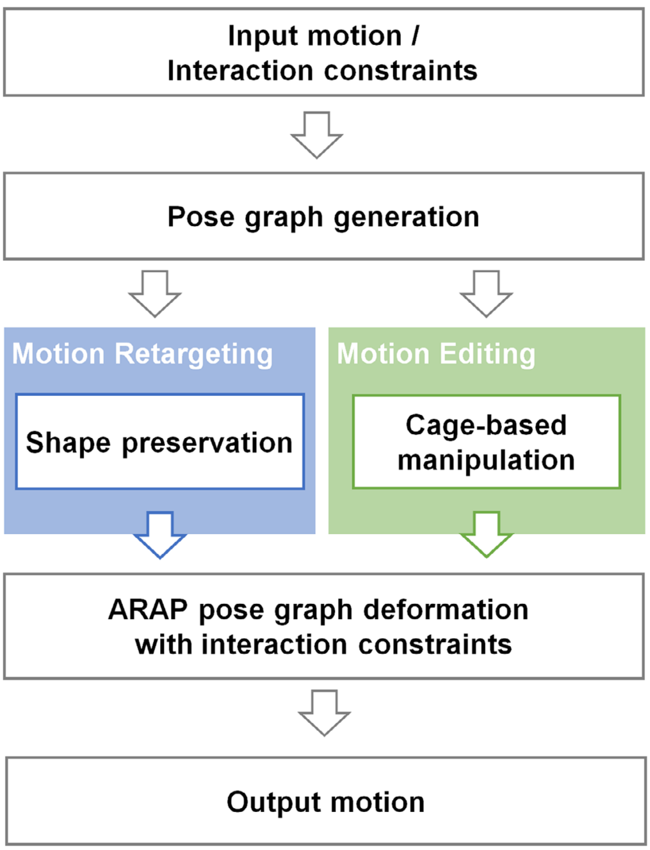

FIGURE 2 Overall workflow of our system

4.2 | Constraints

Multicharacter animation not only consists of individual character motions but also involves various types of relative interactions (see Figure 2). We specify a set of character–character and character–environment constraints that arise from contact and spatial relationship conditions. Each constraint is represented as a linear equation with respect to the pose graph vertices.

4.2.1 | Absolute constraints

The user can specify absolute positions of a joint at a certain frame. Also, the absolute constraints can be automatically specified at the beginning of each motion fragment to drag the joint position of the character. This allows smooth transitioning between motion fragments. The actions of pinning the position and direction on a motion path are formulated as linear equations.

4.2.2 | Contact constraints

We annotate the contact states from the input motion and enforce them during the editing process (e.g., a stance foot should not drift away from its initial position). We can constrain all the joint positions of a character pose. Specifically, the equality constraints are used to enforce feet to be in contact with the ground.

4.2.3 | Length constraints

The length of a link between two joints can be changed during the retargeting process. To address this issue, the length constraints are imposed on the second step optimization to retain the original length of a link with maintaining the direction of the link of the pose graph from the first optimization step:

$$\hat{\mathbf{v}}_i^\alpha - \hat{\mathbf{v}}_j^\alpha = \frac{\|\mathbf{v}_i^\alpha - \mathbf{v}_j^\alpha\|}{\|\overline{\mathbf{v}}_i^\alpha - \overline{\mathbf{v}}_j^\alpha\|}(\overline{\mathbf{v}}_i^\alpha - \overline{\mathbf{v}}_j^\alpha), \tag{1}$$

where $\mathbf{v}_i^\alpha$ and $\mathbf{v}_j^\alpha$ is the ith and jth vertex of the αth original pose graph, respectively. Here, $\overline{\mathbf{v}}_i^\alpha$ and $\overline{\mathbf{v}}_j^\alpha$ is the ith and jth vertex of the αth pose graph obtained from the first optimization, respectively. Then, the newly updated $\hat{\mathbf{v}}_i^\alpha$ and $\hat{\mathbf{v}}_j^\alpha$ of the ith and jth vertex of the αth pose graph are computed by Equation (1). This can be implemented as a soft constraint that minimizes the difference between the directional vector of the link and the desired direction of the link having the original length.

4.2.4 | Center of mass constraints

We use the Center of mass (COM) constraint to model the balancing behavior of a character. For example, one can constrain the COM to be on the barycenter of the support feet for static balancing. The COM constraint attempts to position the COM of the character on a desired location. We precompute the estimated mass of each vertex of the pose graph based on the mass of the nearby body parts. The mass of each body part is distributed to its neighboring vertices using weights inversely proportional to the distance from the center of mass of the body part. The weights are normalized to sum to one such that the total mass is preserved. Given a set of masses of all vertices, the COM position is linearly dependent on the vertex positions.

4.2.5 | Relative constraints

We specify the relative constraints between the characters and formulate them as linear constraints. Given two arbitrary positions in the αth and βth pose graph, the relative constraint equation is represented as $\hat{\mathbf{v}}_i^\alpha - \hat{\mathbf{v}}_j^\beta = \frac{\|\mathbf{v}_i^\alpha - \mathbf{v}_j^\beta\|}{\|\overline{\mathbf{v}}_i^\alpha - \overline{\mathbf{v}}_j^\beta\|}(\overline{\mathbf{v}}_i^\alpha - \overline{\mathbf{v}}_j^\beta)$, which is similar to the length constraint (Equation (1)). This constraint can be set as a soft constraint. Our system also supports a rotation-invariant relative constraint by adding an edge between an arbitrary pair of vertices.

5 | RETARGETING MULTIPLE CHARACTERS

In this section, we describe the multicharacter motion retargeting method, which interactively transfers the source characters onto the target characters with different body scales and proportions. Given a multicharacter motion, we initially produce the new graph geometry of each character individually. Each character motion retargeting is formulated as a quadratic optimization problem performed at each frame independently for better computational efficiency. Specifically, we sequentially solve the pose graph transfer and as-rigid-as possible pose graph deformation. From the first optimization, the solution represents the intermediate motion that preserve the details of original motion. The solution of second optimization is obtained with preserving various constraints as much as possible. After motion retageting, collisions between characters can occur and those are rectified using our multicharacter motion editing framework, which will be explained in the next section.

5.1 | Pose graph transfer

The purpose of the quadratic optimization problem is to preserve the details and local features of the source graph geometry during the retargeting process. Motivated by the work by Sumner and Popović,[1] our retargeting process can be defined as the minimization of the equation below:

$$E_t(\overline{\mathbf{v}}) = \sum_{j=1}^{l} \|\mathbf{S}_j - \mathbf{T}_j\|_F^2, \tag{2}$$

where l is the number of joints of the character, and the source and target transformations of a joint, $\{\mathbf{S}_j\}$ and $\{\mathbf{T}_j\}$, approximately represent the current orientations of the jth joint (see Figure 3). We can formulate Equation (2) as $\mathbf{A}_p\overline{\mathbf{v}} = \mathbf{b}_p$ for each character pose. Concatenating the linear system of a set of hard constraints including foot constraints $\mathbf{H}_p\overline{\mathbf{v}} = \mathbf{h}_p$, the energy function to be minimized can be written as below:

$$\arg\min_{\overline{\mathbf{v}},\lambda_p} \frac{1}{2}\overline{\mathbf{v}}^\top \mathbf{A}_p^\top \mathbf{A}_p \overline{\mathbf{v}} - \overline{\mathbf{v}}^\top \mathbf{A}_p^\top \mathbf{b}_p + \lambda_p^\top(\mathbf{H}_p\overline{\mathbf{v}} - \mathbf{h}_p). \tag{3}$$

Here, λ_p is a Lagrangian multiplier. Note that we solve Equation (3) while dispensing with a set of user-defined constraints to prevent undesirable scaling artifacts in the result. Differentiating Equation (3) with respect to $\overline{\mathbf{v}}$ and λ_p then leads to an augmented system, as follows:[27]

$$\begin{pmatrix} \mathbf{A}_p^\top\mathbf{A}_p & \mathbf{H}_p^\top \\ \mathbf{H}_p & \mathbf{0} \end{pmatrix} \begin{pmatrix} \overline{\mathbf{v}} \\ \lambda_p \end{pmatrix} = \begin{pmatrix} \mathbf{A}_p^\top\mathbf{b}_p \\ \mathbf{h}_p \end{pmatrix}. \tag{4}$$

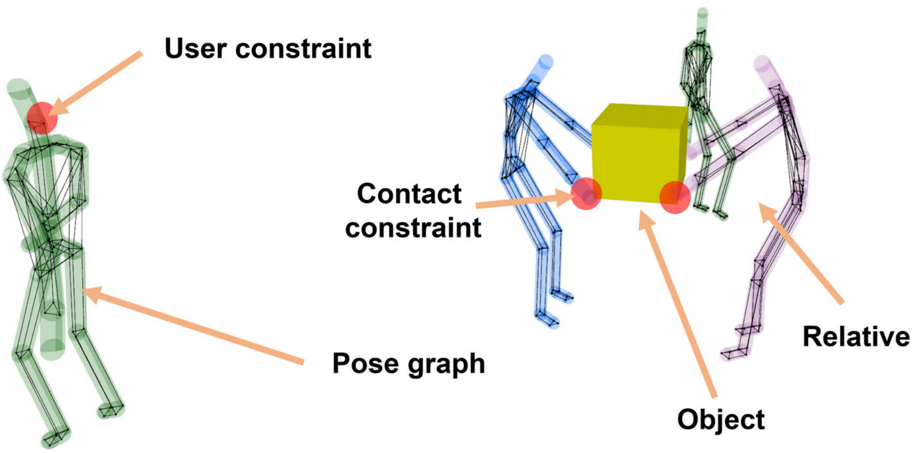

FIGURE 3 Pose graphs (black lines) and constraints are displayed on top of a multicharacter motion. According to the user constraint, the multicharacter motion is edited while maintaining the set of predefined constraints

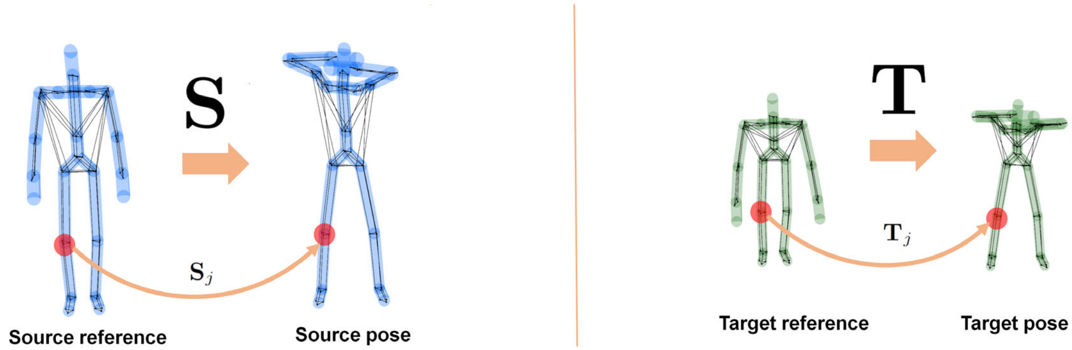

FIGURE 4 We precompute the source transformations $\{\mathbf{S}_j\}$ for all joints in a single character. Our system then automatically generates the target pose with the optimal target transformations $\{\mathbf{T}_j\}$

We can obtain the intermediate result $\overline{\mathbf{v}}$ by means of quadratic optimization, which is sparse linear system, completing this efficiently using a sparse LU solver (see Figure 4(b)).

5.2 | Pose graph deformation

To retain the spatial relationship between characters, the editing of multiple characters is performed in an iterative manner. Each iteration adjusts the intermediate result $\overline{\mathbf{v}}$. We define the pose graph deformation energy term, which is later combined with other energy terms enforcing the contact with the environment and original edge lengths. The deformed pose graphs are obtained by minimizing the following equation:

$$E_d(\hat{\mathbf{v}}) = \sum_i \sum_{j \in \mathcal{N}(i)} w_{ij} \| (\hat{\mathbf{v}}_i - \hat{\mathbf{v}}_j) - \mathbf{R}_i(\overline{\mathbf{v}}_i - \overline{\mathbf{v}}_j) \|_2^2, \tag{5}$$

where $\hat{\mathbf{v}}_i$ is the deformed ith vertex, w_{ij} is the normalized weight value for the edge of the pose graph, and $\mathbf{R}_i$ is the ith rotation matrix. A well-known approach for minimizing this energy is to use alternating least-squares optimization steps first using the constant $\mathbf{R}_i$ and then to update the $\mathbf{R}_i$ as proposed in the work.[3]

We can formulate Equation (5) combined with other energy terms for a set of soft constraints including edge length, relatives, and absolute position constraints to keep the character motion quality as a linear system, $\mathbf{A}_s\hat{\mathbf{v}} = \mathbf{b}_s$ for all the character poses. Concatenating the linear system with a set of user-defined hard constraints including character-environment contact constraints and foot constraints $\mathbf{H}_s\hat{\mathbf{v}} = \mathbf{h}_s$ forms a constrained least squares optimization for the multi-character editing as below:

$$\arg\min_{\hat{\mathbf{v}}, \lambda_s} \frac{1}{2}\hat{\mathbf{v}}^\top \mathbf{A}_s^\top \mathbf{A}_s \hat{\mathbf{v}} - \hat{\mathbf{v}}^\top \mathbf{A}_s^\top \mathbf{b}_s + \lambda_s^\top (\mathbf{H}_s \hat{\mathbf{v}} - \mathbf{h}_s). \tag{6}$$

Here, λ_s is the vector containing the Lagrange multipliers. Differentiating Equation (6) with respect to $\hat{\mathbf{v}}$ and λ_s leads to the augmented system as follows:[27]

$$\begin{pmatrix} \mathbf{A}_s^\top \mathbf{A}_s & \mathbf{H}_s^\top \\ \mathbf{H}_s & \mathbf{0} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{v}} \\ \lambda_s \end{pmatrix} = \begin{pmatrix} \mathbf{A}_s^\top \mathbf{b}_s \\ \mathbf{h}_s \end{pmatrix}.$$

This optimization is performed for the selected characters having the spatial relationships between characters and character-environment contacts which should be kept preserved (see Figure 4(c)).

6 | SPACE-TIME EDITING OF MULTIPLE CHARACTERS

In this section, we describe the interactive space-time motion editing algorithm that allows the preservation of user-defined constraints. Our approach differs from those in previous works[4,14,15] that provide user interfaces for full-body

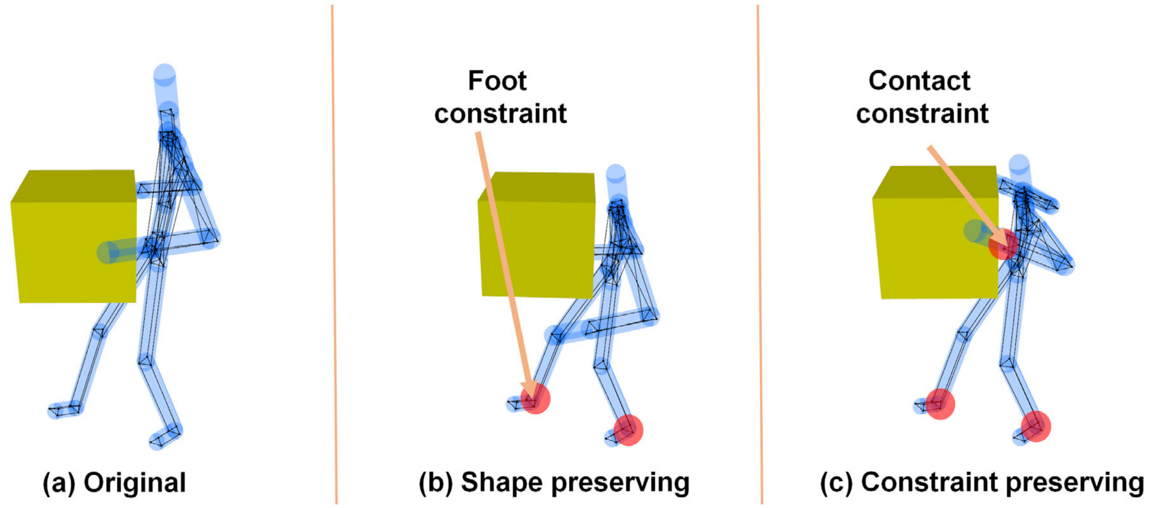

FIGURE 5 (a) original pose (b) retargeted pose after solving Equation (4) only. (c) retargeted pose by sequentially solving Equations (4) and (7)

space-time motion editing. In contrast to earlier methods that interactively edit only two-dimensional multicharacter motion paths (trajectories), we provide the user with a three-dimensional deformable cage mesh, which encloses a range of frames of the joints of character motions. We then can manipulate subsets of joint points in the cage mesh coherently so that a set of frames of character motion can be locally deformed while retaining the user-defined constraints. Our space-time motion editing method is technically based on hybrid deformation, which combines space deformation[28] for the smooth manipulation of the joint point of the character mesh, and surface deformation for editing the cage mesh in an ARAP manner.[3]

6.1 | Spatiotemporal cage

For interactive and smooth space-time motion editing, we use a three-dimensional deformable cage which encloses a trajectory of joint constraints. We can manipulate subsets of a joint trajectory in a cage coherently so that a set of frames of character motions can be locally deformed while maintaining the user-defined constraints. The cage mesh is a polygonal mesh in three-dimensional space. Similar to the method for creating the pose graph, we construct the cage mesh from the triangulation connecting a sparsely sampled set of constraint. (see Figure 5). The key-frames are evenly distributed in time, and the number of key-frames is set proportional to the arc length of the trajectory of the constraint. The user can also drag and pin down any vertex and/or arbitrary interior point to manipulate them interactively.

6.2 | Cage deformation

The key idea of cage deformation is to combine the mean value coordinates (MVC)[28] for the parametrization of joint points inside the cage mesh and ARAP surface deformation[3] to find the optimal rotation matrices of all triangles of the deformed mesh so as to maintain local details and the global shape of the original mesh to the greatest extent possible. The interior of the cage is parametrized with MVC. The newly updated joint point can be computed as a weighted sum of the cage vertices. When the cage vertex is updated, the displaced joint point $\hat{\mathbf{p}}_k$ is computed as $\hat{\mathbf{p}}_k = \sum_{i=1}^{m} \lambda_i^k \hat{\mathbf{v}}_i^c$, where $\hat{\mathbf{v}}^c = \{\hat{\mathbf{v}}_1^c, \hat{\mathbf{v}}_2^c, \dots, \hat{\mathbf{v}}_m^c\}$ are the updated cage vertices. In addition, we make the joint trajectories smooth across the cage boundary. To do so, we blend the updated and original joint trajectories using the smooth ease-in/ease-out function to avoid discontinuities (see more details in Reference 4). Cage deformation starts with the fixing of $\{\mathbf{R}_i\}$ as identity matrices. Similar to the iterative solving of Equation (5), we can simultaneously obtain the deformed cage vertices and desired joint points.

7 | RESULTS

Figures 6 and 7 show screenshots from the results of our method (see the accompanying Video S1). These experiments were conducted to demonstrate the usefulness of the relative constraints. In the first example, each character is performing a single squat exercise while linking both arms together. Using a relative constraint at each linkage, the squat motion is seamlessly retargeted to characters with largely different body proportions. The second experiment shows that the editing algorithm can be used for removing collisions which occur during the original motion of relaying boxes. After

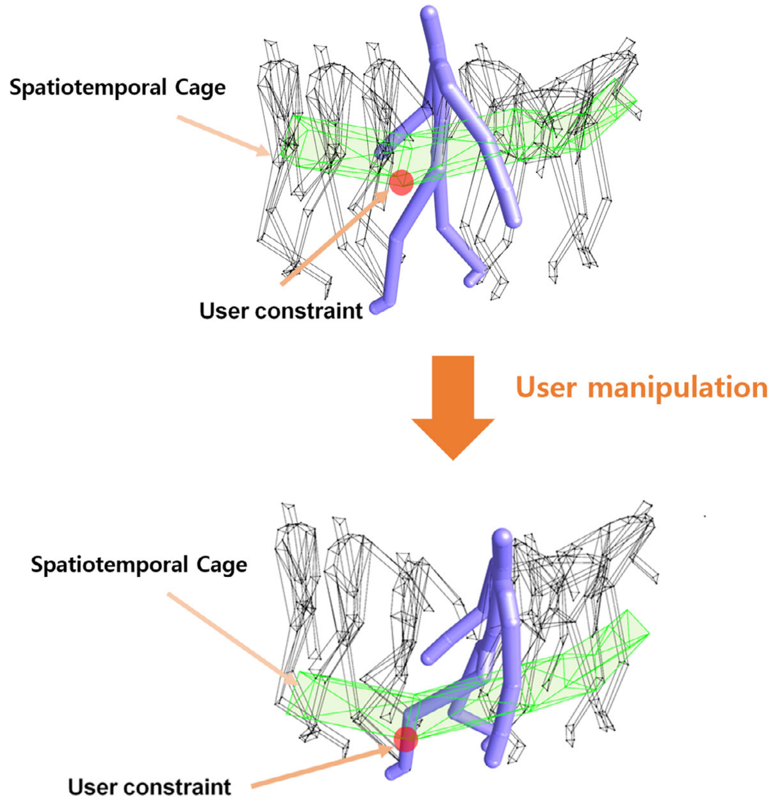

FIGURE 6 A spatiotemporal cage colored green is constructed on demand when the user clicks on a vertex to move (upper); our system edits the pose graphs coherently when the user releases the mouse button at the desired location of the cage vertex (lower)

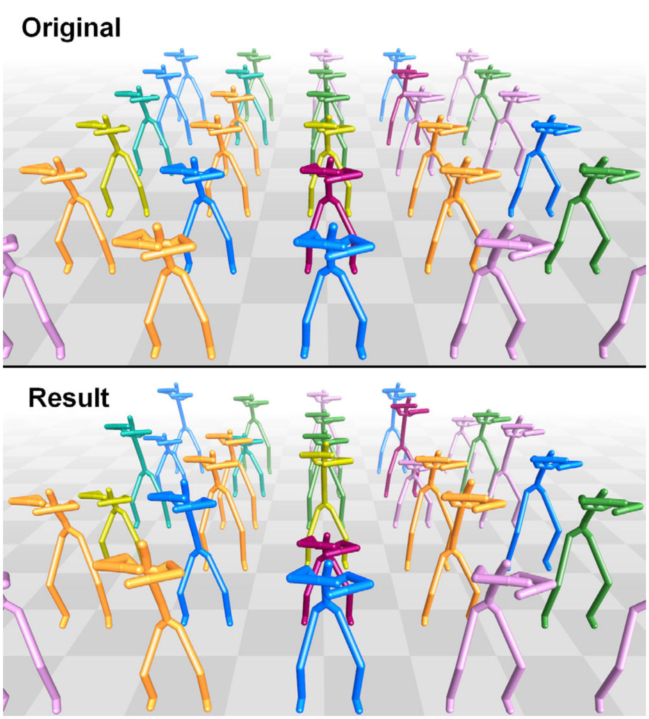

FIGURE 7 Our system interactively retargets dozens of squat motions while preserving the spatial relationship between the hands

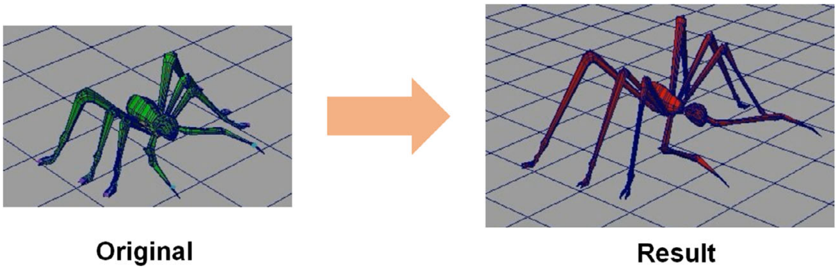

FIGURE 8 Our system can retarget spider motion with a complex joint structure

TABLE 1 Comparison with previous inverse kinematics (IK)-based motion retargeting approaches with L-BFGS.[29] First column represents the total number of frames. The second column is the number of joints of a given source character. The third column is the number of constraints imposed on the joints. The fourth and last column indicates the total solving time for IK-based motion retargeting and for the proposed method, respectively

	Number of frames	Number of joints	Number of constraints	IK-based retargeting	Ours
Human	120	32	4	157 ms	41 ms
Spider	120	48	12	3198 ms	72 ms
Wolf	120	108	6	1728 ms	179 ms

deforming the cage, the input motion is manually edited so that the resulting animation does not have any collisions between characters and boxes. Also, Figure 8 shows that the proposed method has the ability to retarget to complex-shaped characters efficiently.

Table 1 shows the timing results for each character with different skeletal configurations. Our system outperforms IK-based motion retargeting as ours solves a sparse linear system. Our method is approximately one order of magnitude faster than the previous approach.

In all experiments, the proposed method produced natural animation of multiple characters. We solve the alternating least squares problem to obtain the optimal degree of graph deformation. The process took less than a second for the most complex scenario (Figure 6). Furthermore, the user interface was smooth and continuous at a rate faster than 60 Hz, providing immediate feedback to the user as the user drags the constraint. This was possible because the space-time cage editing can be finished in a negligible amount of time.

8 | DISCUSSION

We have presented a novel multicharacter retargeting and editing method based on a linear mesh deformation framework. The resulting pose graphs from the editing are smooth and accurate, and we achieve good computational efficiency and robustness compared to conventional space-time optimization. In addition, we experimentally showed that our cage-based space-time editing method provides a speedup due to per-frame IK solutions given smooth trajectories of constraints. A further speedup is easily possible because multi-character spatial editing for multiple frames can be executed in a parallel manner using multiple threads.

An automatic collision avoidance algorithm is not incorporated into our motion retargeting framework. However, the existing algorithm shown in Kim et al.[4] can be easily adapted to our method with advanced features. For example, when crowds are passing through a narrow corridor, the existing approach can only modify the root trajectories while each character uses a conservative collision bound that contains all the body parts of the character. In our case, the similar method can be applied, then the remaining intersections could be automatically avoided using only joint movements.

For the future work, we plan to support the physical-based constraints. If the edited trajectory is largely different from the original trajectory, the edited motion is likely to no longer be physically plausible. The physical models such as the linear inverted pendulum model[30] are possible extension and can be easily adapted to our data structure.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their constructive comments. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2019R1F1A1063467, NRF-2020R1A2C1012847). We thank Dexter Studio for modeling the character and rendering the scenes.

DATA AVAILABILITY STATEMENT

Research data are not shared.

ORCID

Jongmin Kim https://orcid.org/0000-0001-6637-8215

REFERENCES

1. Sumner RW, Popović J. Deformation transfer for triangle meshes. ACM Trans Graph. 2004;23(3):399–405.
2. Botsch M, Sumner R, Pauly M, and Gross M. Deformation transfer for detail-preserving surface editing. In: Vision, modeling & visualization. Berlin, Germany: AKA Verlag; 2006. p. 357–364.
3. Sorkine O, Alexa M. As-rigid-as-possible surface modeling. Proceedings of the Symposium on Geometry Processing. Barcelona, Spain: 2007;4:109–16.
4. Kim J, Seol Y, Kwon T, Lee J. Interactive manipulation of large-scale crowd animation. ACM Trans Graph. 2014;33(4):83:1–83:10.
5. Gleicher M. Retargetting motion to new characters. Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM; 1998. p. 33–42.
6. Lee J, Shin SY. A hierarchical approach to interactive motion editing for human-like figures. Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. New York, NY: ACM; 1999. p. 39–48.
7. Hecker C, Raabe B, Enslow RW, DeWeese J, Maynard J, van Prooijen K. Real-time motion retargeting to highly varied user-created morphologies. ACM Trans Graph. 2008;27(3):27:1–27:11.
8. Witkin A, Kass M. Spacetime constraints. Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques. New York, NY: ACM; 1988. p. 159–168.
9. Liu CK, Hertzmann A, Popović Z. Composition of complex optimal multi-character motions. Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Vienna Austria: 2006. p. 215–22.
10. Zhao J, Badler NI. Inverse kinematics positioning using nonlinear programming for highly articulated figures. ACM Trans Graph. 1994;13(4):313–36.
11. Buss SR, Kim JS. Selectively damped least squares for inverse kinematics. J Graph Tools. 2004;10:37–49.
12. Grochow K, Martin SL, Hertzmann A, Popović Z. Style-based Inverse kinematics. ACM Trans Graph. 2004;23(3):522–31.
13. Kulpa R, Multon F. fast inverse kinematics and kinetics solver for human-like figures. Proceedings of IEEE Humanoids. Tsukuba, Japan; 2005. p. 38–43.
14. Kwon T, Lee KH, Lee J, Takahashi S. Group motion editing. ACM Trans Graph. 2008;27(3):80:1–8.
15. Kim M, Hyun K, Kim J, Lee J. Synchronized multi-character motion editing. ACM Trans Graph. 2009;28(3):79.
16. Ho ES, Komura T, Tai CL. Spatial relationship preserving character motion adaptation. ACM Trans Graph. 2010;29(4):33.
17. Al-Asqhar RA, Komura T, Choi MG. Relationship descriptors for interactive motion adaptation. Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation. New York, NY: ACM; 2013. p. 45–53.
18. Uk Kim S, Jang H, Kim J. A variational U-Net for motion retargeting. Comput Animat Vir Worlds. 2020;31(4-5):e1947.
19. Villegas R, Yang J, Ceylan D, Lee H. Neural kinematic networks for unsupervised motion retargetting. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT: 2018:8639–8648.
20. Aberman K, Li PU, Lischinski D, Sorkine-Hornung O, Cohen-Or D, Chen B. Skeleton-aware networks for deep motion retargeting. ACM Trans Graph. 2020;39(4):62–1.
21. Lee J, Lee KH. Precomputing avatar behavior from human motion data. Graph Models. 2006;68(2):158–74.
22. Shum HP, Komura T, Shiraishi M, Yamazaki S. Interaction patches for multi-character animation. ACM Trans Graph. 2008;27(5):114.
23. Kwon T, Cho YS, Park SI, Shin SY. Two-character motion analysis and synthesis. IEEE Trans Vis Comput Graph. 2008;14(3):707–20.
24. Wampler K, Andersen E, Herbst E, Lee Y, Popović Z. Character animation in two-player adversarial games. ACM Trans Graph. 2010;29(3):26.
25. Shum HPH, Komura T, Yamazaki S. Simulating multiple character interactions with collaborative and adversarial goals. IEEE Trans Vis Comput Graph. 2012;18(5):741–52.
26. Ho ES, Chan JC, Komura T, Leung H. Interactive partner control in close interactions for real-time applications. ACM Trans Multimed Comput Commun Appl. 2013;9(3):21.
27. Boyd S, Boyd SP, Vandenberghe L. Convex optimization. Cambridge, MA: Cambridge University Press; 2004.
28. Ju T, Schaefer S, Warren J. Mean value coordinates for closed triangular meshes. ACM Trans. Graph. 2005;24:561–6.
29. Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. Math Program. 1989;45(1-3):503–28.

30. Kajita S, Kanehiro F, Kaneko K, Yokoi K, Hirukawa H. The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Maui (HI), USA: 2001:1.

AUTHOR BIOGRAPHIES

Jongmin Kim is an Assistant Professor at the Department of Computer Science and Engineering, Kangwon National University (KNU), Republic of Korea. He was a research professor at the Institute for Embedded Software, Hanyang University in 2015 and a mocap software developer and researcher at Weta Digital in 2016, and also worked with Weta Digital as a mocap consultant in 2017. His current research interests include computer graphics and animation, deep learning, and numerical optimization.

Yeongho Seol is a senior engineer at NVIDIA. He's experienced in developing computer vision, character animation, and machine learning technologies and making them useful in real-world digital human/character production. The tools and techniques he developed have been actively used to create successful digital characters in many movies, including the εPlanet of the Apesε series, the εHobbitε series, and the εAvengersε series. His research has appeared in a number of forums including Siggraph and Siggraph Asia. Prior to NVIDIA, he worked at Weta Digital as a motion capture developer and asset researcher.

Taesoo Kwon is a professor of Computer Science at the Hanyang University since 2012. He also leads a computer animation research group. His research interests center on improving the design and performance of animation techniques, mainly through the application of physics-based models and machine learning techniques.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Kim J, Seol Y, Kwon T. Interactive multi-character motion retargeting. Comput Anim Virtual Worlds. 2021;32:e2015. https://doi.org/10.1002/cav.2015