A New Motion Editing Algorithm Based on Motion Capture Data

Zhihua Chen[1] Lizhuang Ma[1] Xiaomao Wu[1] Yan Gao[1]

[1]Department of Computer Science & Engineering,
Shanghai Jiao Tong University, 200030, Shanghai, China

Abstract

This paper presents a new algorithm for motion editing in a way that preserves basic physical features of the captured motion. The main difficulty that physically-based approaches encounter is high complexity of the physical character model. To address this problem, we utilize a motion analysis technique to automatically determine an appropriate simplification model, and then implement motion mapping by using this simplified model as an intermediate agent. Our motion editing algorithm includes three sub problems of motion editing which are all based on the constrained optimization technique. Experimental results show that physically realistic animation can be produced with our algorithm.

Keywords: Computer animation, motion capture, motion editing, motion mapping

1. Introduction

We propose a new motion editing method for adapting motion capture data to different character. Our algorithm is based on spacetime optimization method [1] because it is easy to adjust and modify overall physical properties of the resulting motions. But it is not practical for creating and editing physically realistic motions of the complex human model. The high nonlinearity of dynamics constraints and high degrees of freedom (DOFs) make it difficult for the optimizer to converge to the optimal solution.

Some researchers edit complex human motion by simplification. Popović and Witkin [2] present the motion transformation method based on model simplication. The main shortcoming of this method is that user manually reduces the numbers of DOFs of the original character. Pollard [3] constructs the task model for character run sequence by using Mass-Spring System and scales motion data. Although his method runs in real time, it still needs improvements for the better quality of the retarget motion. Sofonova and his colleagues [4] utilize the technique of principal component analysis (PCA) to reduce the motion capture data and solve a simplified optimization problem for synthesizing a physically realistic motion.

Similar to the above algorithms, our method also tries to lower the complexity of the motion editing problem. We employ simplified model, however, our simplification method relies on neither the animator's interaction nor the motion database. We reduce the DOFs automatically by using a motion analysis technique.

Our framework is a motion capture-driven system. The captured data here represent motion of human skeleton model. The system includes two main parts: automatic model simplification and motion mapping. The first part aims at automatically choosing a simplified model suitable for the captured motion. It breaks into three steps: constraints detection, motion identification and automatic model determination. By using this model as an intermediate agent, we carry out motion mapping based on the spacetime optimization in the second part. This part also consists of three steps: motion fitting, motion scaling and motion reconstruction.

2. Automatic simplification

Pullen and Bregler show that the movements of human joints are highly correlated [5]. That is to say, there are many 'redundant' DOFs in human joints. Once these DOFs are removed, the non-linearity and complexity of the motion editing is greatly reduced.

We develop a motion analysis technique to automatically find the simplified model suitable for the captured motion. The core of the technique contains constraints detection principles (CDP) and motion identification principles (MIP).

CDP focuses on detecting foot-ground contact constraints (Cc). It includes as follows:

Distance criteria: We check whether the vertical distance between the foot and the ground is below a threshold.

Velocity criteria: We check if the vertical component of the foot's velocity is below a threshold.

Applying CDP to the captured motion, we obtain the foot-plant constraint frame sets S_l and S_r. S_l is the set of frames when the left foot of the character touches the ground, while S_r is the set of frames when the right foot touches the ground.

MIP is designed to classify some typical motions including walk, run, jump and skate, in terms of foot plant constraints. Let $q_c(t)$ be a vector-valued function which represents the configuration of the character at time t. Then $q_c(t)$ denotes the captured motion. Let T be the set of all motion frames and can be represented as;

$$T = \{t_i \mid 1 \le t_i \le n\} \quad (1)$$

where n is the total number of motion frames. Then

MIP can be formulated as:

$$\begin{cases} type = WALK, if S_l \neq S_r, and S_l \bigcap S_r \neq \Phi \\ type = RUN, if S_l \neq S_r, and S_l \bigcap S_r \neq \Phi \\ type = JUMP, if S_l = S_r, and S_l \neq \mathrm{T} \\ type = SKATE, if S_l = S_r, and S_l = \mathrm{T} \end{cases} \quad (2)$$

The steps of the model simplification are as follows:

1. Use the approach of kinematic character simplification [2] and set up simplified models in advance for typical motions;
2. Detect the position constraints of the captured motion $q_c(t)$ in terms of CDP and then get S_l and S_r;
3. Identify the type of $q_c(t)$ in terms of MIP;
4. Determine the simplified model suitable for $q_c(t)$.

3. Motion mapping

After the simplified model is determined, the captured motion is then edited and retargeted in the motion mapping process. The process includes three mapping steps: motion fitting, motion scaling and motion reconstruction. All the three steps are based on constrained optimization approaches.

In the motion fitting step, the captured motion data is adapted to the simplified character model. This step aims at reducing the dimensionality of the motion data. The simplified motion data is then scaled and modified in order to fit new character in the motion scaling step. And in the motion reconstruction, the edited motion data from the previous step is then retargeted from the simplified character model back to the full one.

3.1. Motion fitting

In this step, we use spacetime constraints optimization method to retarget the captured data from the origin character to the simplified character that has been chosen automatically. In order to ensure that the motions of the two character models are dynamically similar to each other, we define a set of physical measurements M. The definition of M varies with the types of motions and aims to capture the basic properties of motions. For example, we define M for both of human run and walk as:

$$M = \{M_{com}, M_{angle}, M_{feet}\} \quad (3)$$

where the first component of M includes the centers of mass(COM) of the upper body and COM of the lower body, the second component contains the angle measurements of the hip joint , and the third one represents the trajectories of the feet .

Let $q(t)$ be a vector-valued function which represents the configuration of the character at time t. The measurement of the motion is a function of $q(t)$:

$$M = M(q(t)) \quad (4)$$

Then the objective function is defined as:

$$E(q_2(t)) = w_1 * \int \left| (M(q_1(t)) - M(q_2(t)) \right| dt + w_2 * \int \left| \ddot{q}_2(t) \right| dt \quad (5)$$

where $q_1(t)$ represents the captured motion of the full character, $q_2(t)$ represents the motion of the simplified character, $\ddot{q}_2(t)$ is the second derivative of $q_2(t)$, and w_1 and w_2 are weights.

The constraints (C) in our method break down into three types of constraints, namely, kinematics constraint (C_k), comfort constraint (C_c) and balance constraint (C_b). C_k defines the end-effectors of the character to be located at the desired position, for example, the feet of the character is required to touch the ground at a specific instant. C_k is defined as follows:

$$f(q_2(t)) = c_1 \quad (6)$$

where $f(q_2(t))$ is kinematics function, and c_1 is constant vector which represents the locations of character limbs.

C_c ensures that the torques of the character joints do not exceed the specific limit. This constraint is as follows:

$$\tau(q_2(t)) \le c_2 \quad (7)$$

where $\tau(q_2(t))$ represents the torque function that calculates the torques of human joints, and c_2 is the torque data from [6].

C_b is used to achieve dynamic balance of motions. We calculate C_b according to Tak's method [7] that adjusts the ZMP trajectory of the character so that it falls inside the supporting area S.

To summarize, the forward mapping algorithm is reduced to an optimization problem:

$$\min E(q_2(t)) \text{ subject to } C(q_2(t)) = 0 \quad (8)$$

3.2. Motion scaling

Since the target character has different parameters, such as length and mass of the limbs, from the original character, we scale the simplified motion to the simplified model of the target character according to relative length measurements and relative mass measurements. Then we get the scaled motion $q_3(t)$ by calculating a constrained optimization problem like (8).

3.3. Motion reconstruction

This step is an inverse process of the forward mapping step. In the step, the scaled motion is retargeted to the full model of the target character which has same DOFs as the original character. We define the objective function as:

$$E(q_4(t)) = w_1 * \int \left| q_4(t) - \hat{q}_1(t) \right| dt + w_2 * \int \left| \ddot{q}_4(t) \right| dt \quad (9)$$

where $q_4(t)$ represents the retargeted motion of the full character, $\hat{q}_1(t)$ represents the motion which is obtained by simply scaling the captured motion $q_1(t)$ according to relative length measurements.

Then we define a constraint on measurements as:

$$C_m(q_4(t)) = M(q_1(t)) + M(q_3(t)) - M(q_2(t)) \quad (10)$$

So this step is formulated as an optimization problem:

$$\min E(q_4(t)) \text{ subject to } \begin{cases} C(q_4(t)) = 0 \\ C_m(q_4(t)) = 0 \end{cases} \quad (11)$$

where C contains C_k, C_c and C_b.

4. Results

We use our editing algorithm to edit several common motions , involving human run, walk and jump, etc. to different characters (Figure1). We obtain the body dimensions and mass distributions from the biomechanics literature [6,8]. Sequential Quadratic Programming (SQP) method is used to solve the constrained optimization problem. Experiments show our algorithm can easily converge to the optimal value due to model simplification. Reduction of motion DOFs, for example, from 54DOFs to 20DOFs for human walk (Table1), reduces greatly not only the dimension of the optimization solution, but also the non-linearities of the physical constraints.

Times that the motion fitting step takes range from 5 to 15 minutes, depending on the duration of the captured motion sequences. This steps cost long time because it runs on the original model, however, it does not matter. Given a captured motion, the motion fitting step and automatic simplification are only calculated one time. Then by repeatedly running the motion scaling step and motion reconstruction step while modifying character parameters, we can obtain resulting motions for different characters. These steps require times less than 1 minute in our experiments.

5. Conclusions

In this paper, we propose a new motion editing algorithm based on a physical spacetime optimization method. In contrast to the previous simplification methods which were dependent on either human intervention or motion database, our algorithm uses a motion analysis technique to automatically determine the simplified model. That is to say, our algorithm is easier to use and requires no database. The traditional method of spacetime optimization can only deal with geometrical constraints, however, our optimization algorithm can also solve physical constraints.

6. References

[1] M. Gleicher. “Motion editing with spacetime constraints”. Proceedings 1997 Symposium on Interactive 3D Graphics, pp. 139–148, Apr. 1997.

[2] Z. Popović and A. Witkin.”Physically based motion transformation”. Proceedings of Siggraph 1999.

[3] N. Pollard. “Simple machines for scaling human motion”. Eurographics workshop on Animation and Simulation,1999.

[4] A. Safonova, J. K. Hodgins, N. S. Pollard. “Synthesizing physically realistic human motion in low-dimensional, behavior specific spaces”. Proceedings of Siggraph 2004.

[5] K.Pullen and C.Bregler. “Motion capture assisted animation: Texturing and synthesis”. In Proceedings of SIGGRAPH 2002.

[6] M.Grosso, R.Quach, E.Otani, J.Zhao, S.Wei, P.Ho, J.Lu and N. Badler. Anthropometry for computer graphics human figures. Technical Report MS-CIS-87-71, University of Pennsylvania, Dept. of Computer and Information Science, Philadelphia, PA, 1987.

[7] S.Tak, O-y.Song, and H-s Ko. Motion balance filtering . Proceedings of the 21th European Conference on Computer Graphics (Eurographics-2000), 2000, vol. 19, 3 of Computer Graphics Forum, 437–446.

[8] P. Leva. Adjustments to Zatsiorsky-Seluyanov’s segment inertia arameters. .Journal of Biomechanics, 29(9):1223–1230, 1996.

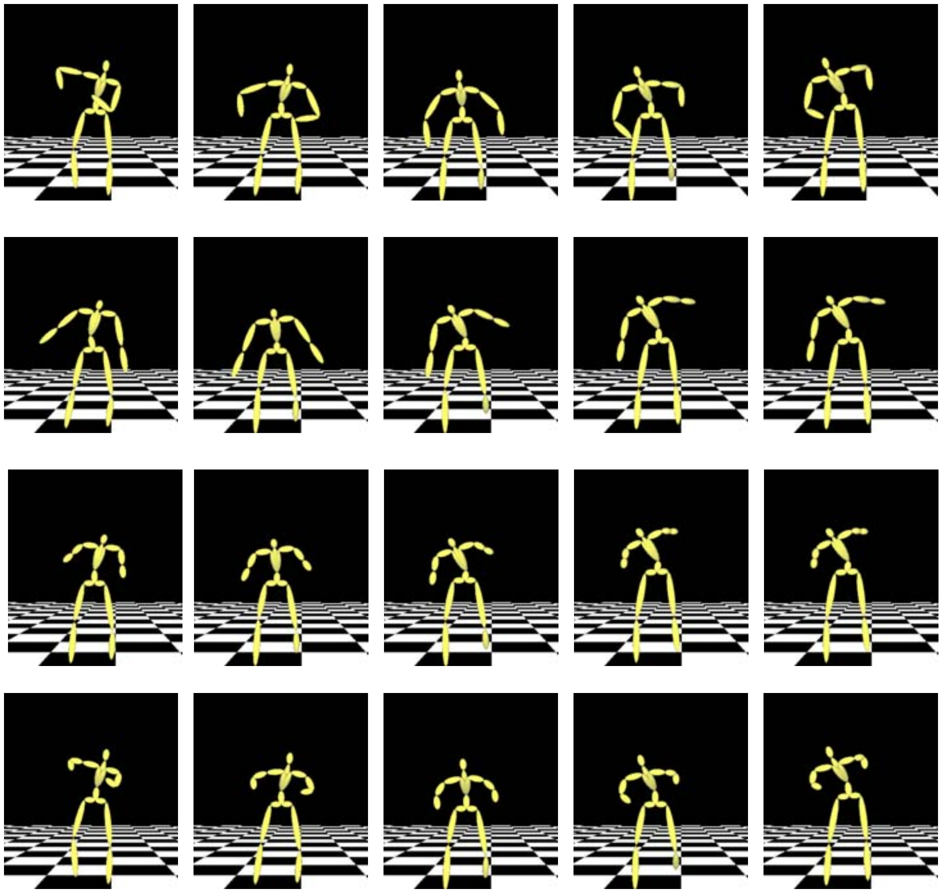

Fig.1: Motion mapping of human walk. (Top row) Motion capture data. (Second row) The motion is simplified during the motion editing step. (Third row) In the motion scaling step, the simplified motion is scaled to the new character which has short arms. (Fourth row) The scaled motion is retargeted to the complex character model in the motion reconstruction step.

Tab. 1: The DOFs of the simplified models.

Name of joints	Number of DOFs
Shoulders	6(3 for Euler angles of each shoulder)
Root (pelvis)	6 (3 for Euler angles and 3 for spatial position)
Hips	6 (3 for Euler angles of each hip)
Knees	2 (1 for z Euler angle of each knee)
Total	20