Space-Time Curve Analogies for Motion Editing

Yuan Wu, Hongxin Zhang, Chao Song, and Hujun Bao

State Key Laboratory of CAD&CG, Zhejiang University, P.R. China
{wuyuan,zhx,songchao,bao}@cad.zju.edu.cn

Abstract. This paper presents a method for analogizing high-dimensional space-time curves, and shows how it can be used to transform a motion sequence into new content and styles according to two accompanied reference sequences. By providing examples, our system can estimate the parameters of a statistical model which incorporates spatial and temporal factors for motion curves. A user can then easily modify existing motion sequences, within these parameters, to have new semantics as examples. This method can also be used to enrich motion capture databases or edit recorded animations in batch mode.

Keywords: motion editing, space-time curve, motion capture.

1 Introduction

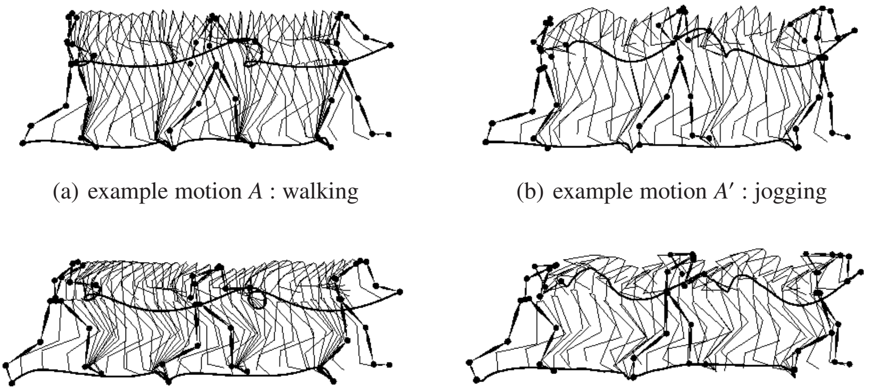
With the development of motion capture systems and the growing needs from entertainment industry, complicated motion databases are leveraged in many graphics applications. These databases always contain various characters, contents, styles and their tremendous combinations, and constructing such a database is laborious and quite time-consuming. Moreover, the animators suffer from creating new characters, which means enormous animation clips should be generated in order to match those new characters. It is difficult to rapidly attach new contents and styles to a motion sequence while preserving its original characteristics. To remedy this shortcoming, we propose a method that can fast calculate the difference between two motions and apply this difference to other motions easily.

In this paper, we present a method to learn motion transformation from examples, which alleviates these issues by providing animators a tool for fast motion analogies. With this approach, the animators may choose two reference sequences (say A and A'), our system will capture the difference between them, which includes content, styles, rhythms and pathes. Then, the animator can apply this difference on the third sequence B to obtain a desired motion sequence B'. Mathematically, the above procedure can be formulated as following paradigm:

$$A : A' :: B : B' \tag{1}$$

For example, one may set a normal walking sequence (A) and a jogging sequence (A') as reference sequences, the difference between the two examples will be extracted immediately, and apply this difference on a normal crouching sequence (B) to acquire a jog-like crouching sequence (B'), see Figure 1. Essentially, we treat motion sequences

F. Chen and B. Jüttler (Eds.): GMP 2008, LNCS 4975, pp. 437–449, 2008.



(a) example motion A : walking

(b) example motion A' : jogging

(c) target motion B : crouching

(d) result motion B' : jog-like crouching

Fig. 1. Style analogies: crouching (B) is transformed to jog-like crouching (B'), according to the manner that walking (A) transforms to jogging (A')

as space-time curves and convert our problem of motion editing to the problem of high-dimensional curve processing. So we use the terminology from Hertzmann et al. [10] for reference, and call the approach *motion analogies*.

Our work has three main contributions. First, we show how learning of motion transformation can be formulated as a high dimensional space-time curve analogies problem. Second, we show the way to establish a statistical model, estimate its parameters, and then transfer the target motion sequence with these parameters. Finally, we model and analogize the spatial and temporal transformation in an uniform framework.

The remainder of this paper is organized as follows. Section 2 provides an overview of related work. Section 3 describes our motion analogies algorithm in detail. We demonstrate several results of our analogies experiments in section 4. Lastly, section 5 concludes the paper and gives some promising directions for future work.

2 Related Work

There are several different ways to synthesize motion contents and styles. Retargeting approaches [6,21,27] transfer entire motion from one character to another which has different skeleton size. Graph based approaches [19,16,2,8] search short clips that satisfy constraints specified by user from motion databases and assemble them to synthesize long motion sequences. Blending approaches [24,14,23] interpolate and extrapolate from examples with linear combinations to synthesize motion sequences that have multiple styles. Cut-Paste methods [13,9] cut parts of skeleton from different characters and combine these separate parts into a whole skeleton. Parametric approaches [15,22] can extend motion example space and allow users to adjust motion in parametric space. However, applying these approaches to our problem is difficult, as the sequences generated by these method are unlikely outside the example space and new contents as well as styles cannot be acquired.

Of closet relation to our method are those works that seek to modify the contents and styles by warping motion curves. Witkin et al. [30] adjust motion curves by doing displacement mapping at key-frames. Amaya et al. [1] extract “emotion” component by comparing neutral and ”emotional” motions and then apply this emotion component to another neutral motion to generate new sequence with the same “emotion” component. Lee and Shin [18] provide a framework which can hierarchically edit motion sequences by using multilevel B-spline approximation. Our method is similar to [1] on the concept of extracting difference of motions. But our system can transfer not only the “emotion” component but also the semantic and rhythm to the target motion sequences. Moreover, we define difference in a more flexible way than displacement mapping does.

Several researchers focus on extracting style factors from one or more examples. Bruderlin and Williams [4] treat motion data as signal and modifies motion styles by adjusting frequency band. Unuma et al. [28] represent motion data as coefficients of fourier transformation and controls styles on frequent domain. Brand and Hertzmann [3] analyze motion sequences with Hidden Markov Model to extract styles, and these styles can be interpolated, extrapolated and applied to new sequences. Unfortunately, these approaches only change motion sequences mainly on the aspect of styles rather than contents. Our method, on the contrary, has no such limitation, and we can transfer both contents and styles.

Most recently, Urtasun et al. [29] use Principle Component Analysis (PCA) to analyze motion data sets, and use PCA coefficients to generate motion sequences with various styles. However, synthesizing new motions depends on a specified training set such as walking, running, jumping, and these sets must be stored somewhere in advance. Our method differs in that we do not need classifying and training the motion data sets and we can transfer the differences by using only two examples. Shapiro et al. [26] apply Independent Component Analysis (ICA) on one example to extract style components and allow users intuitively choosing the components that they want to transfer. Our method differs in that the users specify the styles for transfer by selecting examples at the beginning, instead of choosing independent components after an analytic procedure. Moreover, we can transfer the differences on cadences which cannot be manipulated by [26].

A few authors have described ways to learn the mapping between input and output motions by examining their relationship. Dontcheva et al. [5] learn to map a low-dimensional acting to a high-dimensional motion by using Canonical Correlation Analysis, and drive character animations by acting widgets movements. Hsu et al. [12] treat the mapping mechanism as a linear time invariable system and transfer styles by estimating dynamic metrics. Their system requires that the input motions must be inside the example space. Our method does not need such requirement, because our intent is exactly to generate motion sequences for new characters which does not exist in the origin motion database.

The paradigms of analogies also exist in other research fields. Hertzmann et al. [10] presents a method for image analogies to filter an image by following the manner of example origin-filtered images pair. Later, Hertzmann et al. [11] analogize stylistic curves from example line drawings for non-photorealistic rendering applications by using texture synthesis technique. Our work is highly inspired by their creative ideas. Our goal is quite different from theirs, although our methods adopt the same paradigms that analogize curves using examples. In our approach, we aim at generating new styles of motion instead of simply modeling styled curves. On the other hand, we emphasis the term *space-time curves* as motion data are characterized not only by geometrical information but also dynamic motion rhythm as well as other time variance factors.

3 Algorithm

Given two example motions A and A', our aim is to learn the transformation from A to A', and apply it to the third motion B to generate a new motion B'. Firstly, the example motions are aligned to the reference motion through a procedure called *time warping*. Our method then use linear model to describe the spatial difference between the warped examples in *spatial estimate* stage. Once the spatial parameters are computed, the model attaches target motion B with new contents and styles to achieve an intermediate motion by *spatial transform* while retains its original characteristics. The procedures of handling temporal factors are quite similar and the details are expatiated in 3.6 . Finally, the intermediate motion is adjusted in temporal domain, through the *inverse time warping* operation, to generate our desirable result. The flow of our method is illustrated in Figure 2.

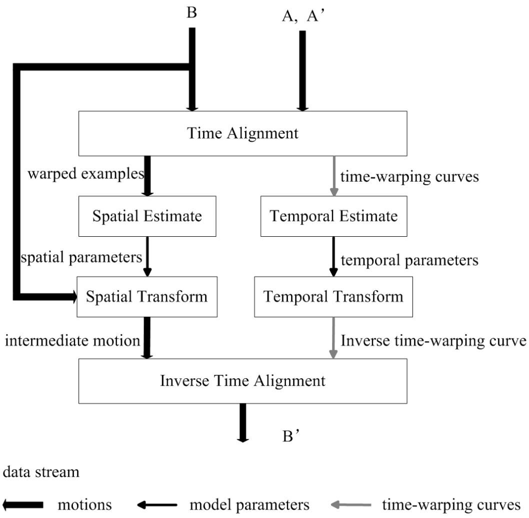


Fig. 2. Diagram of motion analogies

The result motion B' maybe contain artifacts and these can be corrected by an efficient postprocess. In the remainder of section 3, we assume that the skeletons of the three motions, A, A', and B are in the same topology.

3.1 Representation

For the purpose of efficient modeling and general usage, we represent motions in the standard skeletal format. Formally, a motion sequence $M = \{M(t_i)\}$ is a set of frames, which samples a continuous motion curve in time domain. $M(t_i) = (\mathbf{p}_i, \mathbf{q}_i^0, \mathbf{q}_i^1, ..., \mathbf{q}_i^{n-1})$ denotes the configuration of joints at the i^{th} frame, where $\mathbf{p}^i$ is the global position of root, and $\mathbf{q}_i^j$ is the orientation of j^{th} joint in its parent's coordinate system. Especially, $\mathbf{q}_i^0$ is the orientation of root in world coordinate system. Each component of M is called a Degree Of Freedom (DOF).

All joint angles are parameterized by exponential map, since this mapping can handle linear manipulation well on non-linear joint angle and can also avoid the discontinuities problem caused by Euler angle. We refer the readers to Grassia [7] for more discussions on Euler angle, quaternion and exponential map. The root position encodes a global translation, so it is not invariant to different ground-plane coordinate systems. To remove this variability, we represent root position relative to its position in the previous frame.

3.2 Time Alignment

Before modeling the difference between motions, a frame correspondence procedure must be performed to make the example motions, M_A and $M_{A'}$, have the same timing with M_B. Proper time alignment is important to motion analogies; for example, without time alignment the toes in two motions will contact the ground at different time. To align two motions, we construct a time-alignment curve, which maps frames of one motion to correspondence frames in another motion. Formally, the time-alignment curve is defined as follows:

$$S_{A\rightarrow B} \triangleq \{(t_i, t_j) : \text{the } i^{th} \text{ frame of } A \text{ corresponds to the } j^{th} \text{ frame of } B\} \qquad (2)$$

Moreover, the time-alignment curve must satisfy the constraints of continuity, causality and slope limitation; more details can be retrieved in Kovar et al [14].

To acquire the optimal alignment, first, the user need to select appropriate start frame and the end frame of each sequence in order to make each motion has good alignment on the boundary. Second, the user choose one DOF which can delegate the characteristic of the sequence. The correspondence procedure is then performed on the chosen DOF to compute the time alignment curve. According to our experience, the most closet joints to the root are usually being selected.

With time-alignment curves, we then warp the example motions to the target motion. For the following explanations, we assume that M_A is corresponded to M_B, as shown in Figure 3, and the warped motion is denoted by $\widetilde{M}_A$. There are three types of correspondences between M_A and M_B. The first case is that one $M_A(t_i)$ corresponds to one $M_B(t_j)$, we then set $\widetilde{M}_A(t_j) = M_A(t_i)$. The second case is that multiple $M_A(t_i)$, $(M_A(t_{i+k})$,$M_A(t_{i+k+1})$,..., $M_A(t_{i+k+m}))$, are related one $M_B(t_j)$, the warped frame is set by the average of all correspondent frames, $\widetilde{M}_A(t_j) = \frac{1}{m+1} \sum_{i+k+m}^{i+k} M_A(t_i)$. Lastly, one $M_A(t_i)$ maybe corresponds to multiple $M_B(t_j)$ $(M_B(t_{j+k})$,$M_B(t_{j+k+1})$,..., $M_B(t_{j+k+n}))$. In this case, we employee a cubic B-spline interpolation on $M_A(t_i)$ and its neighbor frames

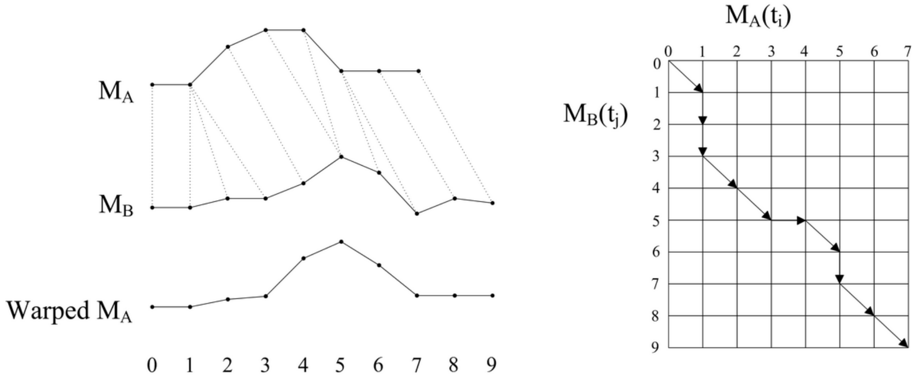


Fig. 3. Time alignment (align M_A to M_B)

to determine the values of $\widetilde{M}_A(t_{j+k})$,$\widetilde{M}_A(t_{j+k+1})$,..., $\widetilde{M}_A(t_{j+k+n})$. In addition, the warping procedure, not alignment, is performed on each DOF independently.

The dynamic time warping algorithm we used for alignment is presented by Sederberg et al [25], but there are some other algorithms that also work well, such as Dontcheva et al [5], Kovar et al [14], and Hsu et al [12].

3.3 Analogies Model

In previous section, we warp M_A and $M_{A'}$ in order to align them to M_B, and the three motions are of the same length now. In this section, we use these motions to estimate the parameters of transformation model. To distinguish from the original motions, we denote the warped motions as $\widetilde{M}_A$ and $\widetilde{M}'_A$ and the alignment curves by $S_{A\to B}$ (aligns A to B) and $S_{A'\to B}$ (aligns A' to B) respectively.

A motion sequence is a space-time curve in high-dimensional motion space. Since directly handle the analogies problem in such space is difficult, we project motion curve on each DOF and process each DOF independently. On each DOF, we model the transformation from $\widetilde{M}_A$ to $\widetilde{M}'_A$ as follows:

$$\widetilde{M}_{A'}(t_i) = a(t_i)\widetilde{M}_A(t_i) + b(t_i), \tag{3}$$

where t_i is the frame index. The parameters **a** and **b** are all vectors that respectively measure the scale and offset of linear transformation, and we use the two parameters to model the difference between two motions.

The idea of such a model is inspired by the work of Witkin et al. [30]. The main difference between our model and theirs are that we define the two parameters on the whole time domain while they define only on a few key-frames. Their method need to specify the key-frames by users or by automated motion segmentation which is also complicated, and our method is free of this problem.

3.4 Estimation

A naive solution of Equation 3 is derived by directly solving it. However, **a** and **b** solved in this way lack of smoothness and are highly sensitive to data noise. In order to get fine distributions of **a** and **b**, we define and minimize the following objective function:

$$E(\mathbf{a},\mathbf{b}) = ||\mathbf{a}\widetilde{\mathbf{M}}_{\mathbf{A}'} + \mathbf{b} - \widetilde{\mathbf{M}}_{\mathbf{A}}||^2 + \omega_a||\mathbf{Sa}||^2 + \omega_b||\mathbf{Sb}||^2, \tag{4}$$

where

$$\mathbf{S} = \frac{1}{2}\begin{pmatrix} -2 & 2 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & -2 & 2 \end{pmatrix}. \tag{5}$$

Here, **S** is a smooth matrix that approximates the first derivatives, and the smoothness terms $||\mathbf{Sa}||^2$ and $||\mathbf{Sb}||^2$ are balanced by two weights ω_a and ω_b. Through this equation, we try to make a tradeoff between the accuracy of the linear approximation and the smoothness of the two parameters. In this paper, we set $\omega_a = \omega_b = 2$. To solve this equation, we set $\partial E/\partial\mathbf{a} = \partial E/\partial\mathbf{b} = 0$ and reformulate Equation 4 as follows:

$$\begin{bmatrix} \widetilde{\mathbf{M}}^{\mathbf{T}}_{\mathbf{A}'}\widetilde{\mathbf{M}}_{\mathbf{A}'} + \omega_a\mathbf{S}^{\mathbf{T}}\mathbf{S} & \widetilde{\mathbf{M}}^{\mathbf{T}}_{\mathbf{A}'} \\ \widetilde{\mathbf{M}}_{\mathbf{A}'} & \mathbf{I} + \omega_b\mathbf{S}^{\mathbf{T}}\mathbf{S} \end{bmatrix}\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{M}}^{\mathbf{T}}_{\mathbf{A}'}\widetilde{\mathbf{M}}_{\mathbf{A}} \\ \widetilde{\mathbf{M}}_{\mathbf{A}} \end{bmatrix}. \tag{6}$$

By using Gaussian elimination, the above linear system can be easily solved. The form of Equation 4 is similar to the one used in Hsu et al. [12]. They define a similar function with an additional time warping matrix to align motions, and the parameters **a** and **b** are useless at last, in their algorithm. We adopt the definition of the objective function and use these parameters to model the transformation between motions.

3.5 Transformation

In the previous step, we obtain the parameters of transformation model and now we can apply these parameters on M_B to generate a new motion. Because the sequences involves in the transformation procedure are all align to M_B(M_B certainly aligns to itself, $\widetilde{M}_B = M_B$), the result motion also aligns to M_B and we denote it by $\widetilde{M}_{B'}$. The transformation equation can be written as follows:

$$\widetilde{M}_{B'}(t_i) = a(t_i)M_B(t_i) + b(t_i) \tag{7}$$

It should not be surprising that the structure of transformation equation is identical to Equation 3, because the manner of transforming target motion is analogized as the examples. Additionally, we transform each DOF independently, in this step, and finally combine them into a whole motion sequence.

3.6 Inverse Time Alignment

After transforming the target motion M_B, we acquire sequence $\widetilde{M}_{B'}$. However, this sequence is not our desired result motion, as it is align to the target motion M_B, or more intuitively, we can say that $\widetilde{M}_{B'}$ is as long as M_B. Formally, we must inverse the time alignment procedure to recover the original rhythm of $\widetilde{M}_{B'}$.

The key to recover the correct timing is solving the inverse time alignment curves $S_{B\to B'}$ which aligns M_B to $M_{B'}$. Unfortunately, there is a conceptual difficulty that we can't calculate $S_{B\to B'}$ before we own $M_{B'}$. To solve this problem, we treat the time alignment curve as an extra DOF of the motions because it is reasonable that the transformation of timing can also be analogized from the manner that M_A transform into $M_{A'}$. We replace the motion curve terms with time alignment curve terms in Equation 3 and 7 to get equations as below:

$$S_{A'\to B}(t_i) = a(t_i)S_{A\to B}(t_i) + b(t_i), \tag{8}$$

and

$$S_{B'\to B}(t_i) = a(t_i)S_{B\to B}(t_i) + b(t_i). \tag{9}$$

Obviously, we have $S_{B\to B}(t_i) \equiv t_i$. By using the method presented in §3.4 and §3.5, we can solve the time-align curve $S_{B'\to B}$. However, time align curves are not uniform-sampled, which means the capacities of $S_{A\to B}(t_i)$ and $S_{A'\to B}(t_i)$ may be different. So, we align $S_{A\to B}(t_i)$ to $S_{A'\to B}$ with the method presented in §3.2 before we model the temporal transformation.

With $S_{B'\to B}$, we can obtain the inverse time alignment curve $S_{B\to B'}$ according to the following definition:

$$S_{B\to B'} \triangleq \{(t_i, t_j) : \text{where}(t_j, t_i) \in S_{B'\to B}\}. \tag{10}$$

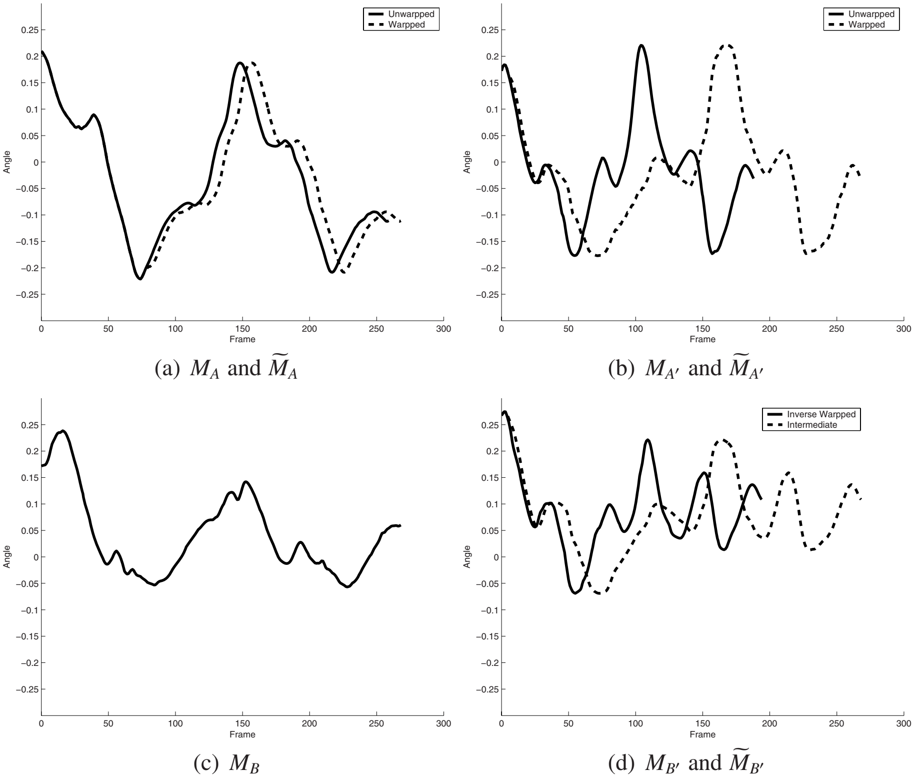
In addition, we refine the curve $S_{B'\to B}$ with uniform quadratic B-spline fitting to satisfy the constraints of continuity, causality and slope limitation, and similar strategy is used by Kovar et al [14]. By warping the intermediate motion $\widetilde{M}_{B'}$ with the inverse time-alignment curve $S_{B\to B'}$, we derive the desired motion $M_{B'}$.

3.7 Postprocessing

Given good example and target motions, our method can generate various stylistic motions, but sometimes a few abnormal frames occurs, which do not satisfy kinematic constraints. The most common phenomenon is footskate, which has been mentioned many times in motion editing and synthesis publications. Fortunately, Kovar et al [17] provide an efficient method to clean up it. Our method on footskate cleaning is similar to theirs, but differs at two points.

First, footskate cleaning problem is also inverse kinematic (IK) problem, and they use analytic method which can solve kinematic constraints of typical skeleton very fast. For general usage purpose, we choose the numerical routine to solve kinematic constraints, which allows our system to handle the case of different skeletal topologies.

Second, in order to avoid sharping adjustments to joint rotations, they allow small changes in the bone lengths. Although the slight changes can hardly be discovered by



(a) M_A and $\widetilde{M}_A$

(b) $M_{A'}$ and $\widetilde{M}_{A'}$

(c) M_B

(d) $M_{B'}$ and $\widetilde{M}_{B'}$

Fig. 4. Example of analogies motion curve of one DOF

the users, the result motions must be saved in the format that support variable bone length. To remedy this shortcoming, we slightly violate the constraints of joints' rotations to match the footprints without introducing visual artifacts, For example, we treat some 1-DOF joints as 3-DOFs joints, and allow these joints rotate slightly on extra DOFs. This technique works well with a lot of data formats, such as BVH, and it can also be seamlessly integrated into traditional algorithms.

4 Results

In this section, we demonstrate some results of our algorithm and all of them can be found in the supplement. We use two motion capture databases from different sources on the internet. One database is shared by `http://people.csail.mit.edu/ehsu/`, and it was used by Hsu et al [12] in their style translation experiments. The other database is shared by `http://www.animazoo.com`. The skeleton of Hsu's data has 18 joints and the latter's contains 23 joints.

There are two main factors that determine the speed of our algorithm. One is the the amount of DOFs and the length of motion sequences. As we treat all the DOFs independently, one more DOF only means the estimate and transform procedures are executed one more time and the time cost of each single procedure does not increase. The length of sequences also affect the time cost, as long sequences need more time to be aligned and transformed and the estimate procedure spend more time to optimize parameters through long sequences. According to our experience, the computational expenditure of our method has a near-linear relation with this factor.

The slope limitation of align path is another factor (cf. the right image of Figure 3). With a small slope, the time warping algorithm is faster, but sometimes failed to converge to a global optimal. And a large value of slope means good aligned motions and more time for calculation. In our experiments, we set the slope limitation = 3. In addition, we use the standard time align algorithm without any optimization in this paper because our object is mainly to validate the analogies framework. The discussion on how to accelerate the time warping procedure can be found in [25].

4.1 Style Analogies

For this sample, we show the analogies of motion styles. We choose a walking motion (A) and a jogging motion (A') as examples. Our system captures the difference between them and apply it to a crouching motion (B) to generate a jog-like crouching motion (B'). The analogical procedure transforms the motion style from walking to jogging, according to A and A', while keeping the character's back bending like B, Figure 1.

4.2 Content Analogies

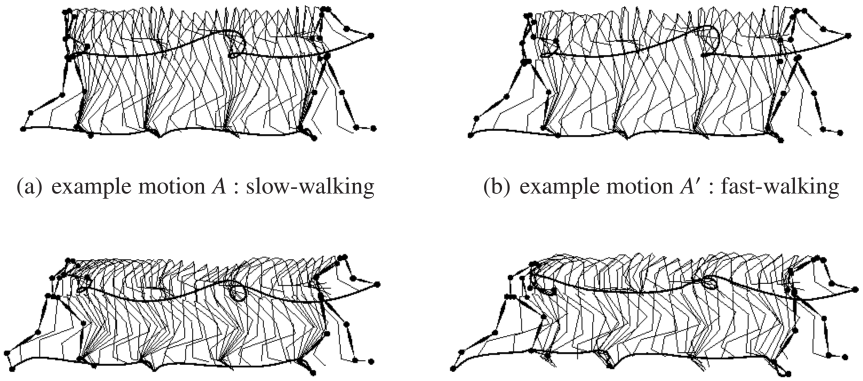
In this sample, we demonstrate the analogies of motion contents. We select a crouching motion (A) and a lateral-crouching motion (A') as examples. We apply the change on a normal forward walking motion (B) to generate a motion that walk in a lateral manner. Notice that the character in B' does not bend its back like it does in A and A', because our analogies model extracts only the difference and certainly eliminates the common features appear in both examples, while most motion blending algorithms are not able to distinguish desired and undesired features.

4.3 Rhythm Analogies

Here, we show a sample of rhythm analogies. We compare a slow-walking motion (A) with a fast-walking motion (A') and apply the rhythmic change on a slow-swaying motion (B). The new motion (B') sways faster while retains other features of (B) except a slight change in arms' swing, according to the examples, see Figure 5.

4.4 Multiple Features Analogies

Lastly we perform a complicated experiment that analogize multiple features. A normal walking motion and a drunk walking motion are be set as A and A' respectively. The two motions are differ in many ways including style, speed and path. We want to transform all of them of a third motion B, in which a man is making a phone call while strolling. The objective was successfully achieved; we get a man in teetering, who is so drunk that he cannot keep the cell phone beside his ear.



(a) example motion A : slow-walking

(b) example motion A' : fast-walking

(c) target motion B : slow-swaying

(d) result motion B' : fast-swaying

Fig. 5. Rhythm analogies: slow-swaying (B) is transformed to fast-swaying (B'), according to speed change between slow-walking (A) and fast-walking (A'). Notice that the content of walking motion, in A and A', is not be transformed to B, because the feature of walking is removed by our analogies model by comparing the two examples.

5 Conclusion

In this paper, we present a novel method for example based motion editing. We treat motion sequences as high-dimensional space-time curves and solve the motion analogies problem as a space-time curve analogies problem. Given two examples , our system can extract the difference between them and apply it to the other motions to generate sequences with new contents and styles while retain their original characteristics.

With our method, animators can easily modify a batch of existing sequences to attach one or more new characteristics. In addition, the animators can also generate animations with highly sophisticated styles by layered transferring original data. Actually, it is not our objective to design a brand new system to replace current motion editing and synthesis approaches. Our goal is to provide an efficient tool which can be integrated into larger animation systems.

As nothing is perfect, our method still has following limitations. Like most motion editing and synthesis methods, our system cannot guarantee that the result motions are valid under physical principle. We believe that physically-based techniques can enhance the reality of motions in postprocessing stage. For example, Liu et al. [20] present a physical model to handle constraints that cannot be solved by kinematic routines. Furthermore, applying our method in cartoon animations maybe encounter some problems. Because the actions of cartoon characters are very exaggerated sometimes and these actions cannot be recorded by motion capture system. Whether our system does work in such cases has not been proved yet. We hope to validate our method on non-human motion data in the future.

Acknowledgements. Motion data are courtesy of http://people.csail.mit.edu/ehsu/ and http://www.animazoo.com. This project is supported in partial by 973 Program of China (No.2002CB312102), 863 Programs of China (No.2006AA01Z335) and NSFC (No.60505001).

References

1. Amaya, K., Bruderlin, A., Calvert, T.: Emotion from motion. In: Graphics Interface 1996, pp. 222–229 (1996)
2. Arikan, O., Forsyth, D.A., O'Brien, J.F.: Motion synthesis from annotations. ACM Transactions on Graphics 22(3), 402–408 (2003)
3. Brand, M., Hertzmann, A.: Style machines. In: ACM SIGGRAPH 2000, pp. 183–192 (2000)
4. Bruderlin, A., Williams, L.: Motion signal processing. In: SIGGRAPH 1995, pp. 97–104 (1995)
5. Dontcheva, M., Yngve, G., Popović, Z.: Layerd acting for character animation. ACM Transactions on Graphics 22(3), 409–416 (2003)
6. Gleicher, M.: Retargetting motion to new characters. In: ACM SIGGRAPH 1998, pp. 33–42 (1998)
7. Grassia, F.S.: Practical parameterization of rotation using the exponential map. Journal of Grahpics Tools 3(3), 29–48 (1998)
8. Heck, R., Gleicher, M.: Parametric motion graphs. In: Symposium on Interactive 3D Graphics and Games (2007)
9. Heck, R., Kovar, L., Gleicher, M.: Splicing upper-body actions with locomotion. Computer Graphics Forum 24(3), 459–466 (2006)
10. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: ACM SIGGRAPH 2001, pp. 327–340 (2001)
11. Hertzmann, A., Oliver, N., Curless, B., Seitz, S.M.: Curve analogies. In: Proceedings of the Thirteenth Eurographics Workshop on Rendering, pp. 233–245 (2002)
12. Hsu, E., Pulli, K., Popović, J.: Style translation for human motion. ACM Transaction on Graphics 24(3), 1082–1089 (2005)
13. Ikemoto, L., Forsyth, D.A.: Enrich a moton collection by transplanting limbs. In: The 2004 Symposium on Computer animation, pp. 99–108 (2004)
14. Kovar, L., Gleicher, M.: Flexible automatic motion blending with registration curves. In: ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 214–224 (2003)
15. Kovar, L., Gleicher, M.: Automated extraction and parameterization of motions in large data sets. ACM Transactions on Graphics 23(3), 559–568 (2004)
16. Kovar, L., Gleicher, M., Pighin, F.: Motion graphs. ACM Transactions on Graphics 21(3), 473–482 (2002)
17. Kovar, L., Schreiner, J., Gleicher, M.: Footskate cleanup for motion capture editing. In: ACM SIGGRAPH Symposium on Computer Animation, pp. 97–104 (2002)
18. Lee, J., Shin, S.Y.: A hierarchical approach to interactive motion editing for human-like figures. In: SIGGRAPH 1999, pp. 39–48 (1999)
19. Li, Y., Wang, T., Shum, H.-Y.: Motion texture: A two-level staticstical model for character motion synthesis. ACM Transactions on Graphics 21(3), 465–472 (2002)
20. Liu, C.K., Hertzmann, A., Popović, Z.: Learning physics-based motion styl with nonlinear inverse optimization. ACM Transcations on Graphics 24(3), 1071–1081 (2005)
21. Monzani, J.S., Baerlocher, P., Boulic, R., Thalmann, D.: Using an intermediate skeleton and inverse kinematics for motion retargeting. In: Eurograhics 2000 (2000)

22. Mukai, T., Kuriyama, S.: Geostatistical motion interpolation. ACM Transactions on Grahics 24(3), 1062–1070 (2005)
23. Park, S.I., Shin, H.J., Kim, T.H., Shin, S.Y.: On-line motion blending for real-time locomotion generation. Computer Animation and Virtual Worlds 15(3-4), 125–138 (2004)
24. Rose, C., Cohen, M.F., Bodenheimer, B.: Verbs and adverbs: Multidimensional motion interpolation. IEEE Computer Graphics and Applications 18(5), 32–41 (1998)
25. Sederberg, T., Greenwood, E.: A physically-based approach to 2-d shape blending. In: SIGGRAPH 1992, vol. 26, pp. 26–34 (1992)
26. Shapiro, A., Cao, Y., Faloutsos, P.: Style components. In: Graphics Interface 2006, pp. 33–39 (2006)
27. Shin, H.J., Lee, J., Gleicher, M., Shin, S.Y.: Computer pupperty: An importance-based approach. ACM Transactions on Graphics 20(2), 67–94 (2001)
28. Unuma, M., Anjyo, K., Tekeuchi, R.: Fourier principles for emotion-based human figure animation. In: SIGGRAPH 1995, pp. 91–96 (1995)
29. Urtasun, R., Glardon, P., Boulic, R., Thalmann, D., Fua, P.: Style-based motion synthesis. Computer Graphics Forum 23(4), 1–14 (2004)
30. Witkin, A., Popović, Z.: Motion warping. In: SIGGRAPH 1995, pp. 105–108 (1995)