Style Components

Ari Shapiro[*] Yong Cao[†] Petros Faloutsos[‡]

University of California, Los Angeles

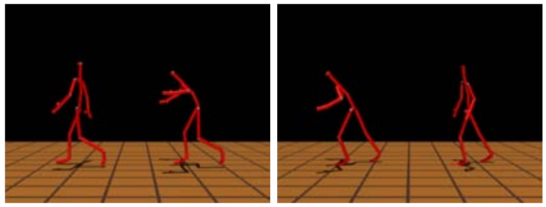

Figure 1: A sneaky style component is added to a normal walk (left image, left character) to synthesize a sneaky walk (left image, right character). The style component transfer is reversed, applying an upright and casual walking style to a sneaking motion (right image, left character) to produce a walk-like sneak, which appears as a tiptoeing motion (right image, right character).

Abstract

We propose a novel method for interactive editing of motion data based on motion decomposition. Our method employs Independent Component Analysis (ICA) to separate motion data into visually meaningful components called style components. The user then interactively identifies suitable style components and manipulates them based on a proposed set of operations. In particular, the user can transfer style components from one motion to another in order to create new motions that retain desirable aspects of the style and expressiveness of the original motion. For example, a clumsy walking motion can be decomposed so as to separate the clumsy nature of the motion from the underlying walking pattern. The clumsy style component can then be applied to a running motion, which will then yield a clumsy-looking running motion. Our approach is simple, efficient and intuitive since the components are themselves motion data. We demonstrate that the proposed method can serve as an effective tool for interactive motion analysis and editing.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation.

Keywords: motion capture, motion editing, animation

1 Introduction

Motion capture data is commonly used to animate interactive characters. It produces realistic and high quality synthetic motion. However, producing variations of the motion to satisfy new situations and constraints is not intuitive, and often results in unnatural motion.

A great amount of research work aims to provide the animators with tools to manipulate motion. The proposed techniques range from simple key-framing and signal processing to different forms of space time optimization and statistical modeling. Such techniques are often computationally expensive or not intuitive for animators that are not technically oriented. In any case, most of these techniques either adjust motion based on a set of constraints or they abstract recorded motion through statistical modeling. There are few techniques that allow the animator to edit directly the style of a motion in intuitive ways. This is the focus of our work.

We introduce a novel method for decomposing motion into various components which can represent the style and expressiveness of a motion without the need to key-frame animation or to analyze frequency bands. The resulting components can, in turn, be applied to other motions through a variety of editing operations, generating new motions that retain the basic content of the original motion while adding the style of the component motion. Thus, motion representing a person walking in a sneaky manner can be decomposed so as to extract the *sneakiness* of the motion. This *sneakiness style component* can then be applied to a normal walk in order to create a sneaky-looking walk. Conversely, our method allows the reciprocal application of style to the above example. The characteristics of a walking motion can be extracted as a separate *style component* and in turn added to a sneaky motion, yielding a walk-like sneaking motion. In addition, the amount of the *style component* can be interpolated so as to create a continuum of different motions between the original motion and the new stylized motion. Thus, the original walk from the example above could be combined with a sneaky *style component* in order to create a motion that is halfway between sneaking and walking. Thus, we can create transitions between the original motion and the new, stylized motion.

Motion decomposition is performed automatically through Independent Component Analysis (ICA). A user then interactively selects one or more of the resulting *style components* that best represent the style of the desired motion. The components resulting from the decomposition are displayed visually for the user. These components can be combined together with a variety of visual editing functions to better represent the expressiveness and nuances of the motion. The chosen *style components* are then applied to the original motion yielding a new, stylized motion. Unlike previous methods for stylizing motions, our method is completely visual and requires no knowledge of key framing, frequency bands or statistical analysis. Our approach is the basis of a simple and intuitive interactive tool for analyzing and editing motion data.

The remainder of the paper is organized as follows. Section 2 provides an overview of related work and background information. Section 3 describes our motion decomposition method. Section 4 explains how we interactively edit motions. Section 5 presents our results and discusses the limitations of our approach. Lastly, Sec-

[*]ashapiro@cs.ucla.edu
[†]abingcao@cs.ucla.edu
[‡]pfal@cs.ucla.edu

tion 6 concludes the paper.

2 Related Work

Motion capture systems and recorded data are readily available. Applying recorded motion to virtual characters produces high quality motion efficiently and easily. However, it not practical or even possible to capture the entire range of motions that an interactive character might need to perform.

In order to remedy this shortcoming, motion synthesis applications [2, 19, 18, 16, 15] use a finite database of motion segments which can be synthesized into longer motions. The focus of these algorithms is efficient searching of the motion database for motion segments that satisfy the control parameters, such as a user-defined path or annotations, and a range of physical constraints. Note, that Li et al [19] use a Linear Dynamic System (LDS) to abstract the motion database and provide a search algorithm that works in the LDS space.

Retargeting transfers motions generated for one character onto another. Gleicher [8] and Shin et al [27] mainly solve the problem of motion retargeting by applying motions to characters with different body proportions. While the process of retargeting effectively changes some aspects of the original motion, its goal is to transfer movement content, not movement style.

Vasilescu and Terzopoulos [31] use multilinear analysis to extract stylistic aspects of facial motion from three different actors and reapply the style to each other. This method was not performed on full-body animation. Pullen and Bregler [23] create statistically similar variations of an original motion using a multi-level sampling technique. The animator can key frame a subset of the DOF of a character and automatically provide key frames for the remaining DOFs.

Motion editing, which is the focus of our work, is a challenging problem that has received a lot of attention. Earlier work exploits mostly ideas from signal processing. Bruderlin and Williams [5] apply signal processing operations to provide stylistic variations of the original motion. Using a small number of key frames, Witkin and Popović [33] warp motion to satisfy new constraints. Gleicher [9] also proposes an interesting warping technique.

Of closest relation to our method are those works that seek to transfer the style of one motion to another. Thus, our goals are very similar to those outlined by Unuma et al [29], which uses Fourier techniques to change the style of human gaits through extrapolation and interpolation of different motions. Our method differs in that it is entirely visual and relies upon a user's decision to determine which stylistic aspects to retain.

Urtasun et al [30] use PCA to decompose sets of motion data, such as walking and running at different speeds, and use PCA coefficients to synthesize motions with different cadences and styles. Stylistic variations can be synthesized by relating a new motion to the coefficients that are stored from the decomposed motion sets. Our work differs in that we do not need to generate examples of motion data first; we are able to transfer style using only one example motion. Also, our method is not able to synthesize movement at different speeds.

Amaya et al [1] extract emotion by comparing neutral and non-neutral motions. The difference between the motions is then added to another neutral motion to give that third motion the style of the non-neutral motion. Our method differs in that we do not require one motion to be neutral while the other represents the non-neutral emotion. Rather, our method allows you to use any two motions with varying amounts of emotional or stylistic content and transfer visually selected stylistic aspects between them.

Rose et al [24] borrow the verb-adverb paradigm from speech to annotate motions into basic, verbs, and modifications, adverbs. Interpolation between motions yields a convex set of variations. Our method differs in that we do not interpolate between two motions, but extract specific stylistic markers related to one motion and apply it to another.

Brand and Hertzmann [4] and Tanco and Hilton [28] both use Hidden Markov Models to capture the style of recorded motions. These styles can then be reapplied to novel motions. However, because the motion primitives are related to the hidden states of the models they cannot be edited explicitly.

Most recently, Hsu et al [12] performed style translation between motions by learning a translation between input and output models. Their method uses Iterative Motion Warping to compute correspondences between motions. Unlike their method which keeps degrees of freedom (DOF) separate in order to extract stylistic aspects of motion, the ICA decomposition described in this paper groups DOF together in order to find correlations between them.

Physically-based methods have also been used in order to apply style from one motion to another. The method from Liu et al [20] constructs a physical model and uses optimization to synthesize a new motion using physical parameters from a different motion. These physical parameters encode the stylistic variations. Although physically-based methods are able to handle constraints that are otherwise inaccessible to kinematic motions, an extremely detailed physical model would be needed to extract subtle nuanced gestures that are part of an in-depth stylistic transfer. For example, our method can extract the shaking hands of an old man (see Section 5.4). This shaking would be difficult to achieve with only a coarse physical model.

Of particular relation to our work within the domain of statistical modeling are the techniques that provide editing parameters through motion decomposition. Chuang et al [7] propose a factorization method that separates visual speech into style and content components. Wang et al [32] separate facial expressions from facial content using dimensionality reduction. Cao et al [6] use Independent Component Analysis to capture the emotional content of visual speech for editing purposes. The authors extract facial emotion components by automatically examining the regions of the face that they affect. In contrast, we allow the user to interactively choose aspects of the motion that represent style or emotion from any part of the body. The idea of using ICA for editing and synthesizing human walking has been proposed by Mori and Hoshino [22]. Saisan and Bissacco [25] show that ICA can be used to model subtleties of cyclic motion data for human bodies. Also, Bissacco et al [3] use a modified version of ICA to generate dynamic models of human walking.

2.1 Independent Component Analysis

Independent Component Analysis is an unsupervised learning technique [14] that separates a set of observed random variables into a linear mixture of hidden random variables that are statistically independent. We call these new random variables *independent components*. Cao et al [6] provides a good description of ICA and a comparison with the more well-known decomposition method, Principal Component Analysis (PCA). In this work, we follow their notation.

The mathematics of ICA are straightforward. Given a set of n random variables $x_1,\ldots,x_n$ each of them can be written as a linear mixture of n latent or hidden variables $u_1,\ldots,u_n$, such that

$$x_j = \sum_{i=1}^{n} a_{ji} u_i,$$

or in matrix notation

$$x = Au. \tag{1}$$

A number of ICA algorithms exist to estimate the mixing matrix A. Estimating A is sufficient, because if the matrix is known, inverting

Equation 1 yields the independent components $u = Wx$. We use the publicly available Matlab [21] implementation of the FastICA [13] algorithm.

Applying ICA involves a two stage pre-processing. First, the data is centered around its statistical mean $E[x]$. Then the centered data is decomposed into a set of uncorrelated variables, typically using PCA. The complete model is as follows:

$$x = E\{x\} + PAu, \quad (2)$$

where $E\{x\}$ is the expectation of x and P is the $n \times m$ PCA matrix.

The number of principal components determines the number of independent components. We can decide to keep $m < n$ independent components, effectively reducing the dimension of our data.

3 Motion Decomposition

We can specify motion capture data in terms of Euclidean coordinates or using joint angles. The Euclidean coordinate representation specifies the location of the markers in Euclidean space for each captured frame. Hierarchical angle representation models the character as a set of hierarchical joints. Data is typically represented by a set of Euler angles and offsets from the parent joints. The results of the ICA decomposition vary according to the format of the motion capture data.

3.1 ICA Performance With Different Representations

The ICA algorithm works on a matrix whose rows represent the individual frames of a motion, and whose columns represent the different channels or degrees of freedom of the motion. Thus, the ICA decomposition can be performed on either the 1) Euclidean coordinate point representation of the motion, the 2) Euler angles representing the rotation of the joints, 3) quaternions that represent the rotation of the joints, or 4) an exponential map representing the rotation of the joints. Similarly, a transformation matrix could be derived from the Euler angles and, in turn, submitted to the ICA decomposer.

However, since the ICA algorithm results in a linear decomposition of the input data, it will produce visually unintuitive results when the input consists of a series of Euler angles. This is likely related to the problem of Gimbal lock, where a linear combination of Euler angles does not always result in a smooth interpolation of the desired angle. Thus, the synthesized component motion shows sporadic twists and turns that greatly disrupt the appearance of the motion. This makes Euler angles a poor choice for the ICA decomposition. Quaternions can be used by submitting the four values of a quaternion to the ICA decomposer. The motions decomposed from quaternions do not suffer from the extreme rotations that we see with the Euler angles. However, the quaternion representation results in subtle rotations that differ slightly from the original motion, since the process of linear combination does not properly separate the quaternion in a meaningful way, either. The results of quaternion decomposition are more visually intuitive than those of Euler angle decomposition. Grassia [10] shows that exponential maps are a good representations for rotations, but are not well suited for parameterizations across continuous frames and thus are not used.

The Euclidean coordinate representation, since it does not involve rotations, does not suffer from the same problem as the rotational representations indicated above. Since the input to the ICA decomposer consists of points in Euclidean space, the ICA decomposition and motion synthesis gives visually meaningful results. Euclidean space can be linearly interpolated without strange side effects. The synthesized motion does, however, result in slight changes in the length of our animated character's limbs, since the point representation does not preserve the distance between joints. This problem is caused by the ICA decomposition which also does not preserve bone lengths. In addition, editing the independent components can result in exaggerated motions that violate bone length constraints. By replacing one *style component* u_1 of motion m_1 with *style component* u_1 of motion m_2, we potentially alter the implicit fixed distances between joints.

We used the Euclidean coordinate representation for most of our experiments. Since we are concerned only with kinematic animation and the visual quality of the final animation, we are not concerned with slight changes in the lengths of the bones of our character. Although, the change of limb length impacts foot plants and also creates occasional foot skating or violation of floor constraints, bone lengths can be easily made globally consistent among frames. In addition, an inverse kinematics solver can be used to satisfy foot plant constraints. Note that altering bone lengths has been used by Kovar et al [17] on kinematic motion for the purpose of correcting foot skating. Also, Harrison et al [11] makes an argument for retaining length changes and measures to what extent these changes can be made without being noticed by the observer.

4 Interactive Editing

Our editing system allows the user to sequence two motions together and identify the independent components that best represent the style differences between them. Once the *style components* are found, the motions are split again and the individual *style components* can be subject to a number of editing operations. Figure 2 summarizes our interactive motion editing approach. The remainder of this section explains the steps depicted in the figure and enumerated here:

1. *Motion Combination.* Two motions are combined together.
2. *Style Component Generation.* The combined motion is decomposed into components.
3. *Style Component Selection.* The user selects components of interest to them.
4. *Style Component Merging.* The user combines components together to better represent the desired characteristics of motion.
5. *Transferring Style Components.* The selected components are transferred in order to create a newly synthesized motion.
6. *Post Processing.* The newly synthesized motion undergoes a motion clean-up phase.

Note that the interface to the system is entirely visual. The user chooses and transfers components by observing a visual representation of those components, and not a frequency-based one.

Each of the steps listed above is explained in greater detail below.

4.1 Motion Combination

Given two motions, x_a and x_b, motion x_{ab} is produced by joining the frames of x_a and x_b. Thus, x_{ab} will have $f = f_1 + f_2$ frames, where f_i is the number of frames for motion x_i. It is essential to combine the motions together in order for the ICA algorithm to find synchronized differences between the two motions.

4.2 Style Component Generation

Once x_{ab} is formed, the user selects the number of *style components* k in which to decompose x_{ab} as well as a representation for the decomposition. The representation can be points, quaternions or Euler angles, see Section 3. Applying the ICA algorithm results in

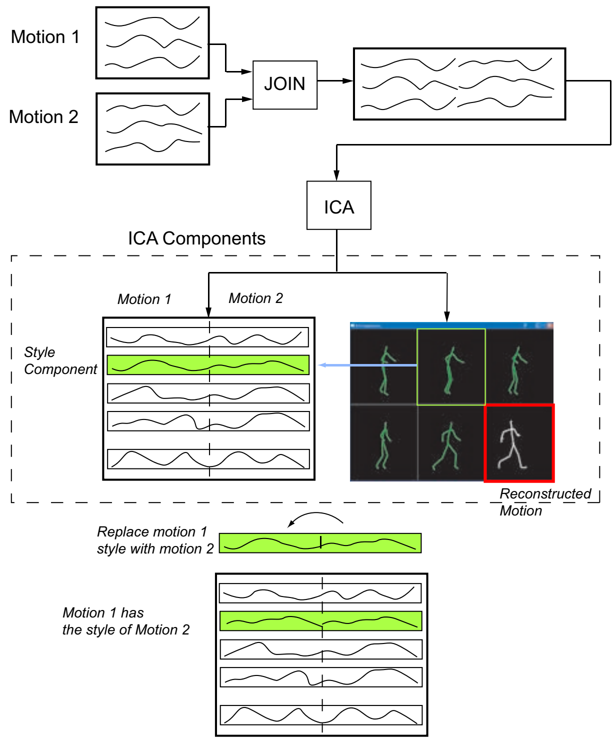

Figure 2: Overview of the ICA-based interactive editing system.

k independent components $u^1_{ab}..u^k_{ab}$ for the combined motion x_{ab}. It is usually sufficient to keep enough components to cover 95% of the variance in the data. However, experimenting with arbitrary numbers of *style components* often produces interesting results. We typically experiment with 3-5 components. Since the decomposition takes only a few seconds, it is trivial to adjust the number of components, view the results, then try a different number of components if needed.

Note that the global translation should be removed from the motion before we apply ICA. This is explained in more detail under Post Processing, Section 4.6.

Each *style component* u^1_{ab} is used to reconstruct part of the original motion as follows:

$$x^i_{ab} = E\{x_{ab}\} + PA(u^i_{ab}e^i), i = 1, ..., k \tag{3}$$

and the result is displayed in a separate window, shown in the middle of Figure 2.

Combining these motion reconstructs an approximation, m'_{ab} of the original motion, m_{ab}, which is shown at the bottom right of the screen captured window in Figure 2.

4.3 Style Component Selection

The user visually analyzes the reconstructed motions, m^i_{ab}, and identifies potentially interesting stylistic components. Good candidates for selection are *style components* that capture differences in posture, cadence as well as defining nuances that appear in one motion but not the other. In Figure 2, the user identifies the middle *style component* on the top row as a style component for use in a style transfer.

For example, in one of our experiments we apply this approach to a joint running+walking motion and we are able to extract a single *style component* that captures the forward lean and raising of the elbows during the running motion. The same *style component* captures the upright stance and dropped arms during the walking motion.

The user can experiment with different decompositions of the same motions by either choosing a different number of *style components* or by rerunning the decomposition algorithm with a different initial guess.

4.4 Component Merging

Our ICA decomposition produces a set of independent components which can be linearly combined to form the original data. It is therefore straightforward to linearly mix components together and produce combined components. Merging *style components* allows the animator to create a smaller set of *style components* that may be more representative or easier to work with. More importantly merged *style components* may provide a more suitable basis for aligning motions, which is often a necessary step for more complex operations.

Mathematically, merging two components u^1 and u^2 results in a combined motion u^{12} as follows:

$$x^{12} = E\{x\} + PA(u^1e^1 + u^2e^2), \tag{4}$$

where e^i is a vector in the canonical basis of A that corresponds to the ith-component.

4.5 Transferring Style

Perhaps the most interesting operation we can perform using our decomposition approach is to transfer style between motions.

Once a style component u^s_{ab} has been selected, it is split into two segments that represent the *style components* of the original two motions, m^s_a and m^s_b. We can then align (time-warp) either x_a to x_b or vice versa depending on which motion's timing we wish to preserve. We align the motions by applying dynamic time warping as described by Sankoff and Kruskal [26] on one of the degrees of freedom (DOFs) of the character. The user interactively selects the appropriate DOF based on her knowledge of the motion and the desired effect. For example, if the resulting motion needs to preserve foot contacts, a good choice is the hip swing degree of freedom. The user can experiment with different degrees of freedom and select the one that produces the desired result.

Once the motions are aligned, the user can generate new motions by replacing u^s_a with u^s_b. Following the notation of Cao et al [6], transferring a style component from one motion to another can be mathematically expressed as follows:

$$x = E\{x\} + PA(u_a + ((u^s_b - u^s_a)^T e^s), \tag{5}$$

where e^s is a unit vector in the canonical basis of A that corresponds to the selected style component.

4.6 Post Processing

The global translation DOF are removed before the ICA decomposition since the decomposition has no intrinsic knowledge of the correlation between foot plants and changes in position. Our tests show that ICA decomposition with the global translation DOF results in a distracting amount of foot skating. Once the final motion has been generated, the global translation from x_a, which was removed before applying the decomposition, is re-added to the motion. Assuming that the style component does not contain much lower body movement, then the process of recombining the original global translation along with time warping generally preserves the leg movements and thus indirectly preserves the foot plants in the newly synthesized motion. The global translation for the base

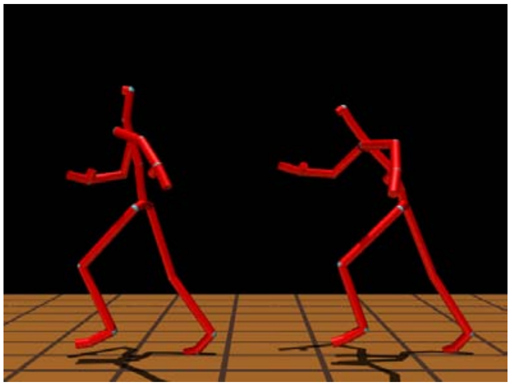

Figure 3: Running (left) and a sneak-like run (right).

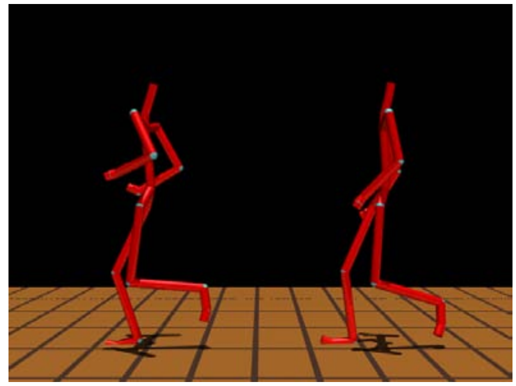

Figure 4: Running (left) and running with a walking style - jogging (right).

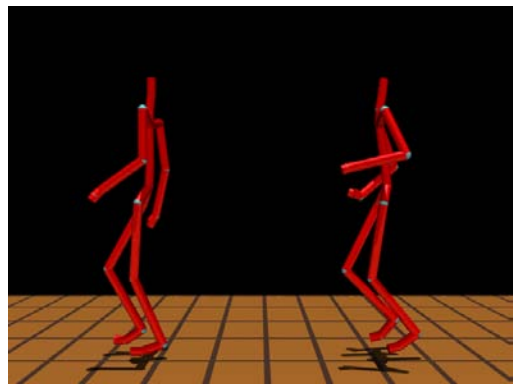

Figure 5: Walking (left) and walking with a running style - power-walking (right).

motion, and not the style motion, is added to the synthesized motion since the main movement corresponds to the base motion.

If the data represents marker positions instead of joint angles, the limb lengths of the character may lengthen or shorten between frames. To correct this, the system automatically employs a filter to restore the correct limb lengths according to the original data by preserving joint angles. In addition, low-pass filtering is automatically done to eliminate high-frequency motions. High-frequency motion is typically caused by the time-warping technique as a result of matching a high-speed motion, such as running, with a low-speed one, such as a very slow walk. *Style component* transfers in the opposite direction, from a low-speed motion to a high-speed motion, result in stiff movements, such as limbs that remain in the same place for an unnaturally long amount of time.

5 Results

Our system is able to decompose motion capture data regardless of the hierarchical structure of the character. We use two different skeleton hierarchies for our examples; a thirty-one joint, sixty-two DOF skeleton and a twenty-six joint, eighty-four DOF skeleton. All motions are displayed in real-time and decomposed with the ICA algorithm in less than 5 seconds. For most of our experiments we use five independent components. Once a *style component* is selected, the motion reconstruction takes less than two seconds.

5.1 Walking and Sneaking

In this example, we transfer a *style component* between a walking motion and a sneaking motion. Joining motions and decomposing them into five *style components* allowed us to successfully identify one of the components that models the difference between the hunched posture of the sneaking motion and the upright stance of the walking motion. Applying this *style component* to both original motions produces two new stylized variations. Figure 1(left) shows a sneaky walk, while Figure 1(right) a walk-like sneak. The latter motion appears to be the motion of a character tiptoeing in order to keep quiet, without the characteristic hunched posture of a sneaky motion.

5.2 Running and Sneaking

Here we combine a running motion with the previous sneaking motion. We find a similar *style component* that captures the hunched posture of the sneak, as in the previous example, and apply it to the run. The sneaky run is shown in Figure 3.

5.3 Running and Walking

For this example we combine a running and a walking motion. A *style component* is found that captures the shrugged shoulders, the raised elbows and the bending of the knees of the running motion. The same *style component* captured the upright stance and relaxed arms of the walking motion. By applying the walking style to the run, our resulting motion resembles a jogging motion, Figure 4, while our run-like walk resembles a power walk, Figure 5.

5.4 Old Man's Walk and An Approaching Fighter

We combine an old man's walk with a threatening, fighter-like approach. The old man's arms shake while he walks, and his legs move in a bow-legged manner. The fighter moves with a steady upper body and a deliberate gait. By finding a *style component* that captures the fighter's uprightness and raised hands, we synthesize a new motion showing an old fighter walking in an aggressive manner. The new motion retains the cadence, shaky arms and bow legged movement, but incorporates the raised hands and slightly raised head, Figure 6.

5.5 Motion Interpolation

The original and stylized motion retain very similar characteristics, including global translation and general movement speed. The alignment between these two motions eliminates problems such as foot-skating and phase differences when interpolating two different motions. Thus, the stylized motion can be linearly interpolated with

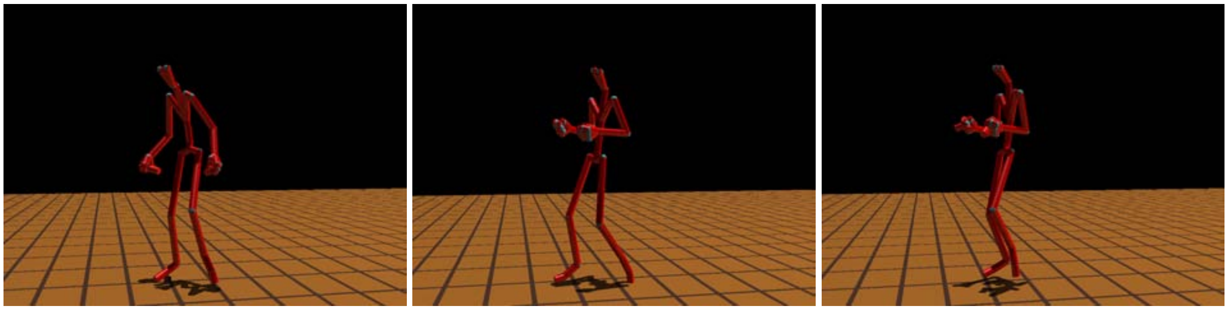

Figure 6: The old man (left), the fighter (middle) and the old man as a fighter (right).

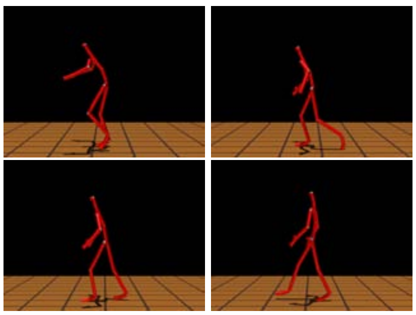

Figure 7: Interpolating between a sneak and a walk-like sneak.

the original motion in order to produce a continuum of motions that contain varying amount of style. Figure 7 shows an interpolation between the sneak and the walk-like sneak (tiptoeing).

5.6 Discussion

The human body can be considered generally as a highly non-linear control system. It is therefore counter-intuitive that linear methods such as LDS as proposed by Li et al [19] and ICA prove to be effective tools for motion modeling and editing. However, it seems that as the human body repeats and learns common motions, such as gaits, it optimizes and simplifies its control strategies. Thus, the observed dynamics of such motions can often be approximated with combinations of linear models.

It is helpful to consider ICA in relation to PCA. PCA and ICA are both statistical techniques that can reduce the dimensionality of data. PCA produces components that are orthogonal, uncorrelated and in often do not have intuitive interpretation when decomposing full-body motion in our method. In contrast, ICA assumes that the components are statistically independent. The first stage of ICA is the whitening stage which is done with PCA. ICA rotates each PCA component independently of each other, and therefore is often able to capture dimensions of the motion data in a more meaningful way than is PCA alone. This has been shown in the case of facial motion by Cao et al [6].

Extracting style is a difficult problem. It is difficult to prove that any style extraction scheme such as frequency band manipulation or other decomposition techniques work well in all cases. Most of these techniques are experimentally proven to work in a range of cases. The same holds for our proposed ICA decomposition method. Our method achieves interesting results within a certain data set of motions that represent movement in full-body animation. A similar scheme has proved successful for facial motion as well [6]. ICA has been used in the vision literature for motion recognition cases. Our technique is simple, fast and provides an intuitive decomposition into motion components, some of which can be identified as representing stylistic aspects of the motion.

Although, our method produced some surprising results with its ability to capture the difference in style of a range of motions, it has several limitations.

Our experiments show that our method is more effective with cyclic motions than with acyclic motions. This is probably due to the fact that aligning cyclic motions is more intuitive than aligning arbitrary motions. However, our decomposition method is often able to separate one-time events, such as gestures, from the cyclic aspects of a motion.

The FastICA [14] algorithm that we currently use does not always converge to the globally optimal decomposition. However, to our knowledge it is one of the most efficient algorithms, which is crucial for interactive editing.

We would also like to clarify that, in this work, we assume that motion data is already segmented into suitable pieces of singular motion. Automatic data segmentation is out of the scope of this paper.

6 Conclusion

We have presented a novel method for interactive motion editing. Our method, based on Independent Component Analysis, provides a meaningful decomposition of the original motion into reusable components called *style components*. An important feature of our decomposition is that the resulting *style components* are themselves motion data. Therefore, they are a familiar model for animators and can be subject to the growing number of techniques that work with motion data.

Based on the proposed decomposition we have defined a set of editing operations that can change the style of an original motion. Of special interest is the ability of our approach to extract stylistic aspects from one motion and apply it to another. At the same time, we can edit the *style components* themselves to reduce or exaggerate their effect on the motion. Using our interactive editing tool we are able to perform efficiently a series of examples that demonstrate the effectiveness of the method.

In summary, we believe that our work presents a fast and interactive technique that can often transfer style between motions. Animators can experiment to quickly obtain new motions.

We have just beginning to explore the possibilities offered by the ICA-based motion decomposition. We believe that it can be equally effective in a range of applications, such as motion segmentation, automatic motion annotation and motion recognition.

Acknowledgements

We would like to thank Frédéric Pighin for his many useful discussions and comments. The work in this paper was partially supported by NSF under contract CCF-0429983. We would also like to thank Intel Corp., Microsoft Corp., Alias Wavefront and ATI Corp. for their generous support through equipment and software grants.

REFERENCES

[1] Kenji Amaya, Armin Bruderlin, and Tom Calvert. Emotion from motion. In *Graphics Interface*, pages 222–229, May 1996.

[2] Okan Arikan, David A. Forsyth, and James F. O'Brien. Efficient synthesis of physically valid human motion. *ACM Transaction of Graphics*, 22(3):417–426, 2003.

[3] Alessandro Bissacco, Payam Saisan, and Stefano Soatto. Modeling human gaits with subtleties. In *Proceedings of 13th IFAC Symposium on System Identification*, August 2003.

[4] Matthew Brand and Aaron Hertzmann. Style machines. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 183–192. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[5] A. Bruderlin and L. Williams. Motion signal processing. *Proceedings of ACM SIGGRAPH: Computer Graphics*, pages 97–104, August 1995.

[6] Yong Cao, Petros Faloutsos, and Frederic Pighin. Unsupervised learning for speech motion editing. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 225–231. Eurographics Association, 2003.

[7] E. Chuang, H. Deshpande, and C. Bregler. Facial expression space learning. In *Proceedings of Pacific Graphics*, 2002.

[8] Michael Gleicher. Retargeting motion to new characters. In *Proceedings of SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, pages 33–42, July 1998.

[9] Michael Gleicher. Motion path editing. In *Proceedings of 2001 ACM Symposium on Interactive 3D Graphics*, pages 195–202, 2001.

[10] F. Sebastian Grassia. Practical parameterization of rotations using the exponential map. *of Graphics Tools*, 3(3):29–48, 1998.

[11] Jason Harrison, Ronald A. Rensink, and Michiel van de Panne. Obscuring length changes during animated motion. *ACM Transactions on Graphics*, 23(3):569–573, August 2004.

[12] Eugene Hsu, Kari Pulli, and Jovan Popović. Style translation for human motion. *ACM Transactions on Graphics*, 24(3):1082–1089, 2005.

[13] Jarmo Hurri, Hugo Gävert, Jaakko Särelä, and Aapo Hyvärinen. Fastica. Helsinki University of Technology, Laboratory of Computer Information Science, Neural Networks Research Centre, 1998.

[14] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

[15] Lucas Kovar and Michael Gleicher. Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics*, 23(3), July 2004.

[16] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. *ACM Transactions on Graphics*, 21(3):473–482, July 2002.

[17] Lucas Kovar, John Schreiner, and Michael Gleicher. Footskate cleanup for motion capture editing. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 97–104. Eurographics Association, 2003.

[18] Jehee Lee, Jinxiang Chai, Paul S. A. Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. Interactive control of avatars animated with human motion data. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 491–500, New York, NY, USA, 2002. ACM Press.

[19] Yan Li, Tianshu Wang, and Heung-Yeung Shum. Motion texture: A two-level statistical model for character motion synthesis. *ACM Transactions on Graphics*, 21(3):465–472, July 2002.

[20] C. Karen Liu, Aaron Hertzmann, and Zoran Popović. Learning physics-based motion style with nonlinear inverse optimization. *ACM Transactions on Graphics*, 24(3):1071–1081, 2005.

[21] Mathworks. Matlab software package.

[22] Hiroshi Mori and Jun'ichi Hoshino. Independent component analysis of human motion. In *Proceedings of International Conference on Acoustics Speech and Signal Processing*, volume VI, pages 3564–3567, 2002.

[23] Katherine Pullen and Christoph Bregler. Motion capture assisted animation: Texturing and synthesis. *ACM Transactions on Graphics*, 21(3):501–508, July 2002.

[24] Charles Rose, Michael F. Cohen, and Bobby Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications*, 18(5):32–41, /1998.

[25] P. Saisan and A. Bissacco. Image-based modeling of human gaits with higher-order statistics. In *Proc. of the Intl. Workshop on Dynamic Scene Analysis, Kopenhagen*, June 2002.

[26] David Sankoff and Joseph B. Kruskal. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. CSLI Publications, 1983.

[27] H. J. Shin, J. Lee, M. Gleicher, and Shin. S. Y. Computer puppetry: and importance-based approach. *ACM Transaction of Graphics*, 20(2):67–94, 2001.

[28] L. M. Tanco and A. Hilton. Realistic synthesis of novel human movements from a database of motion capture examples. In *IEEE Workshop on Human Motion*, 2000.

[29] Munetoshi Unuma, Ken Anjyo, and Ryozo Takeuchi. Fourier principles for emotion-based human figure animation. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 91–96, New York, NY, USA, 1995. ACM Press.

[30] R. Urtasun, P. Glardon, R. Boulic, D. Thalmann, and P. Fua. Style-based Motion Synthesis. *Computer Graphics Forum*, 23(4):1–14, 2004.

[31] M.A.O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *ICPR(2)*, pages 511–514, 2002.

[32] Yang Wang, Xiaolei Huang, Chan-Su Lee, Song Zhang, Zhiguo Li, Dimitris Samaras, Dimitris Metaxas, Ahmed Elgammal, and Peisen Huang. High resolution acquisition, learning and transfer of dynamic 3-d facial expressions. *Eurographics 2004*, 23(3), August 2004.

[33] Andrew P. Witkin and Zoran Popović. Motion warping. In *Proceedings of SIGGRAPH 95*, Computer Graphics Proceedings, Annual Conference Series, pages 105–108, August 1995.