

# Stylistic Motion Decomposition

Ari Shapiro Yong Cao Petros Faloutsos

---

## Abstract

*We propose a novel method for interactive editing of motion data based on motion decomposition. Our method employs Independent Component Analysis to decompose motion data into meaningful components that capture different aspects of that motion. The user interactively identifies suitable components and manipulates them based on a proposed set of operations. In particular, the user can transfer components from one motion to another in order to create new and novel 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 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.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

---

## 1. Introduction

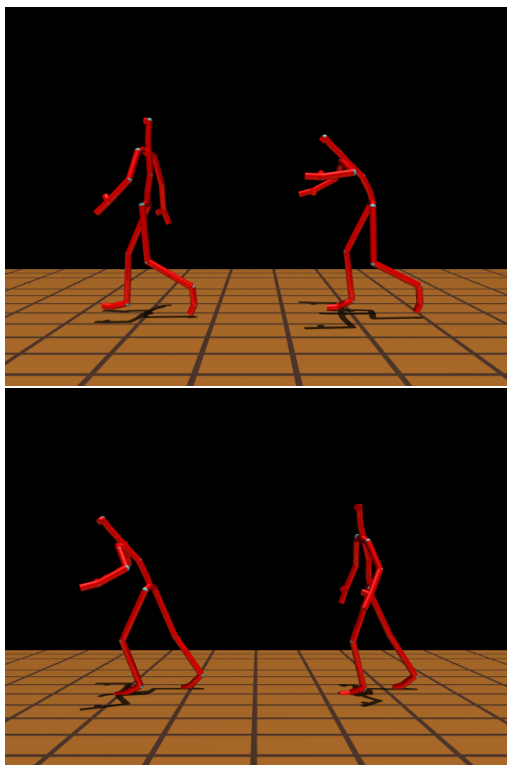
Motion capture data is commonly used to animate interactive characters. It produces realistic and high quality synthetic motion. However, producing variations of the data to satisfy new situations and constraints is not intuitive, and often results in unnatural motion.

A great amount of research work aims to provide the animators with tools to manipulate motion data. 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.

Inspired by [CFP03], 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 operation, generating new motions that retain the basic content of the original motion while adding the style of the component motion.

Thus, motion capture data representing a person walking in a sneaky manner can be decomposed so as to extract the *sneakiness* of the motion. This *sneakiness* 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 separate 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 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 components that best represent the style of the desired motion. 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



**Figure 1:** *Walking and the same walking motion with a sneaky style (top). Sneaking and the same sneaking motion with a walking style, which results in a tiptoeing motion (bottom).*

requires no knowledge of key framing, frequency bands or statistical analysis. Furthermore, since the components themselves are motion data, they can be subject to all the available techniques for manipulating motion data, for example retargeting. 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 the limitations of our approach. Lastly, Section 6 concludes the paper and discusses future work.

## 2. Related Work

Motion capture systems and recorded data are becoming readily available. Applying recorded motion to virtual characters produces high quality motion efficiently and easily. However, for interactive characters, it not practical or even possible to capture the entire range of motions that the char-

acters might need to perform. Most applications usually have a finite database of motion segments, based on which they have to synthesize longer motions. Such approaches have two main challenges. Given some form of control parameters, for example a motion path, they must synthesize appropriate motions by assembling suitable motion segments from the database. Often, the available motion segments cannot be used as they are and have to be edited in some way. For example, we may want to produce a walking motion with style that does not exist in the database.

A number of successful techniques have been proposed that assemble motion segments from a database into longer motions with different levels of control. The focus of these algorithms [AFO03, LWS02, LCR\*02, KGP02] is efficient searching of the motion database for motion segments that satisfy the control parameters, for example a user-defined path or annotations, and a range of physical constraints. Note, that [LWS02] use Linear Dynamic System (LDS) to abstract the motion database and provide a search algorithm that works in the LDS space.

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. [BW95] apply signal processing operations to provide stylistic variations of the original motion. [UAT95] use Fourier decomposition to change the style of human gaits. [ABC96] extracts emotion by analyzing neutral and non-neutral motions. Using a small number of key-frames [WP95] warps motion to satisfy new constraints. [Gle01] also proposes an interesting warping technique. The goals of our method are very similar to those outlined in [UAT95] and [ABC96]. However, our decomposition based on motion components is more intuitive than a frequency-based one. In addition, we are not limited to extracting emotion from two similar motions as in [ABC96]. We extract style components from dissimilar motions as well.

[RCB98] 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. However, it is not clear if the method can scale to large databases. Although [Gle98, SLGY01] mainly solve the problem of motion retargeting, applying motions to characters with different body proportions effectively changes some aspects of the original motion.

More recently, [BH00, TH00] use Hidden Markov Models to capture the style of recorded motions. However, because the motion primitives are related to the hidden states of the models they cannot be edited explicitly. [PB02] 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.

Dynamic approaches to motion editing and synthesis [WK88, PW99, FP03, KB96, LP02, HJS03] aim mainly to

ensure that the resulting motion is realistic and satisfies given physical constraints. Editing the style of the motion can be tedious or time consuming. The same applies for approaches based on dynamic control as such [HWBO95, LvDPF96, FvT01] and hybrid methods [ZH02].

Within the domain of statistical modeling, of particular relation to our work are the techniques that provide editing parameters through motion decomposition. [CDB02] propose a factorization method that separates visual speech into style and content components. Our work is inspired from [CFP03] who use Independent Component Analysis to capture the emotional content of visual speech for editing purposes. Cao et al. extracts 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 Independent Component Analysis for editing and synthesizing human walking has been proposed in [MH02]. However, the length of the paper and its exposition does not allow us to evaluate its results.

Unlike physics-based and hybrid approaches our work is focused on style editing, rather than satisfying given physical constraints. Our main objective is to extract the style of recorded motion and apply it to different motions. Our work falls within the realm of statistical modeling and in particular motion decomposition. Statistical models such as LDS, Hidden Markov models and Bayesian networks are either difficult to edit or not intuitive to work with. In contrast, our proposed technique decomposes the motion into components that, unlike frequency bands and complex mathematical models, are themselves motion data. They are therefore a familiar model for animators and they can be subject to all available motion capture manipulation techniques. Our interactive editing tool allows the user to interactively examine, edit and apply these components to other motions.

## 2.1. Independent Component Analysis

*Independent Component Analysis* is an unsupervised learning technique [HKO01] 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*. [CFP03] provides an excellent description of ICA and a comparison with the more well-known decomposition method, Principal Component Analysis. In this work, we follow their notation.

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

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

or in matrix notation

$$x = Au. \quad (1)$$

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 [Mat] implementation of the FastICA [HGSH98] 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 Principal Components Analysis. 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 points or using joint angles. Point 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) three-dimensional point representation of the motion, the 2) Euler angles representing the rotation of the joints, or the 3) quaternions that represent 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 un-intuitive 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. [Gra98] provides an excellent discussion on different rotation parameterizations.

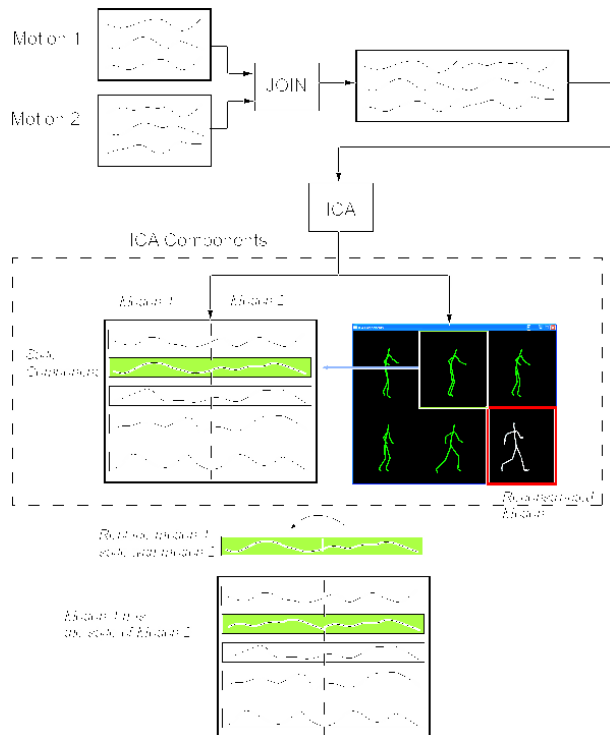
The three-dimensional point representation, since it does not involve rotations, does not suffer from the same problem as the rotational representations indicated above. Since the input to the ICA decomposer consists of points in Euclidean space, the ICA decomposition and motion synthesis gives visually 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 component  $u_1$  of motion  $m_1$  with component  $u_1$  of motion  $m_2$ , we potentially alter the implicit fixed distances between joints.

We used the point 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 create 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 on kinematic motion for the purpose of correcting foot skating [KSG03].

#### 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. *Component Generation.* The combined motion is decomposed into components.



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

3. *Style Selection.* The user selects components of interest to them.
4. *Component Merging.* The user combines components together to better represent the desired characteristics of motion.
5. *Component Editing.* Components may be edited with standard motion editing tools.
6. *Transferring Style.* The selected components are transferred in order to create a newly synthesized motion.
7. *Post Processing.* The newly synthesized motion undergoes a motion clean-up phase.

Note that 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.

#### 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. Component Generation

Once  $x_{ab}$  is formed, the user selects the number of 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 into  $k$  independent components  $u_{ab}^1 \dots u_{ab}^k$  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 components often produce interesting results. We typically experiment with 3-5 components.

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

Each component  $u_{ab}^1$  is used to reconstruct part of the original motion as follows:

$$x_{ab}^i = E\{x_{ab}\} + PA(u_{ab}^i e^i), i = 1, \dots, k \quad (3)$$

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 Selection

The user visually analyzes the reconstructed motions,  $m_{ab}^i$ , and identifies potentially interesting stylistic components. Good candidates for selection are components that capture the posture, cadence and nuances of the original motion, while maintaining its defining aspects. In Figure 2, the user identifies the middle component on the top row as a potential style component.

For example, during one of our experiments we apply this approach to a joint running+walking motion and we are able to extract a single component that captures the forward lean and raising of the elbows during the running motion. The same 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 components or by rerunning the decomposition algorithm with a different initial guess.

We can now define a set of operations that we can apply to the independent components that helps us alter the style of a motion.

## 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 components allows the animator to create a smaller set

of components that may be more representative or easier to work with. More importantly merged 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:

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

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

## 4.5. Component Editing

One of the most important features of our method is that the proposed decomposition produces components that are themselves motion data. We can therefore edit any of these components using published methods that work with motion data. For example, we can simply scale a component or apply more complex techniques such as motion warping [WP95] and motion retargeting [Gle98].

## 4.6. Transferring style

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

Once a style component  $u_{ab}^s$  has been selected, it is split into two segments that represent the style components of the original two motions,  $m_a^s$  and  $m_b^s$ . 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 [SK83] 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  $u_a^s$  with  $u_b^s$ . Following the notation of [CFP03] transferring a style component from one motion to another can be mathematically expressed as follows:

$$x = E\{x\} + PA(u_a + ((u_b^s - u_a^s)^T e^s)), \quad (5)$$

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

## 4.7. 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  $x_d$ , which was removed before applying the decomposition, is re-added to the motion. This process of recombining the original global translation along with time warping preserves the foot plants in the newly synthesized motion. The global translation for the base motion, and not the style motion, is added to the synthesized 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. 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 style between a walking motion and a sneaking motion. Joining motions and decomposing them into five independent components allowed us to successfully identify an interesting style component. This component models the difference between the hunched posture of the sneaking motion and the upright stance of the walking motion. Applying this 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 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.

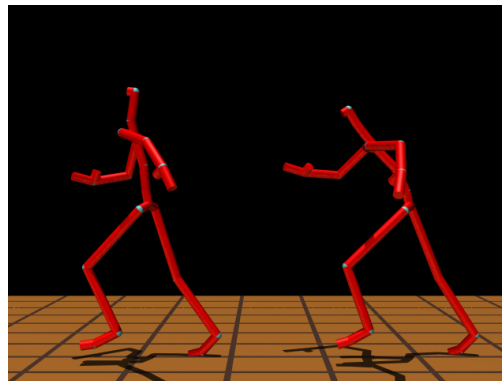


Figure 3: Running (left) and a sneak-like run (right).

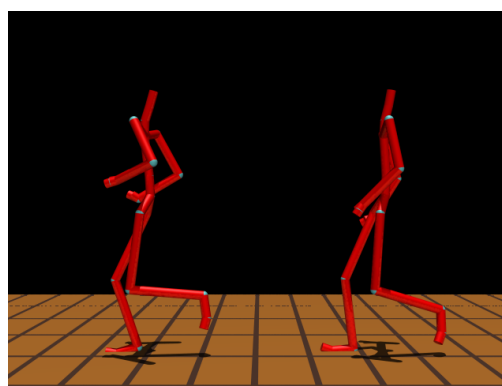


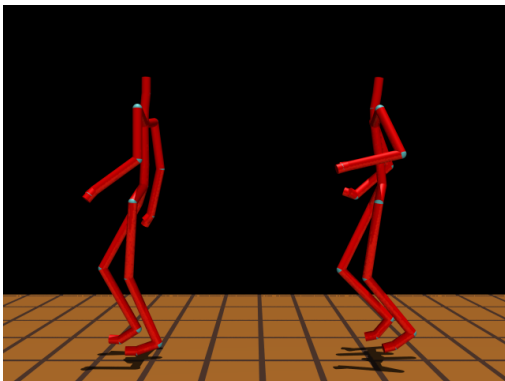
Figure 4: Running (left) and running with a walking style - jogging (right).

### 5.3. Running and Walking

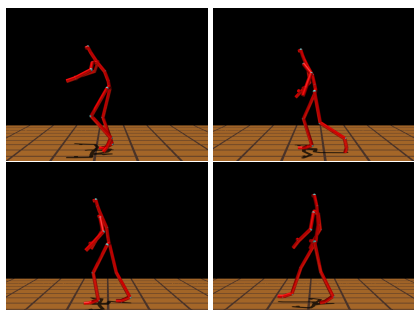
For this example we combine a running and a walking motion. A style component is found that captures the shrugged shoulders, the raised elbows and the bending of the knees of the running motion. The same component captured the upright stance and relaxed arms of the walking motion. By applying the walking 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. 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 the original motion in order to produce a continuum of motions that contain varying amount of style. Figure 6 shows an interpolation between the sneak and the walk-like sneak (tiptoeing).



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



**Figure 6:** Interpolating between a sneak and a walk-like sneak.

### 5.5. Discussion

The human body is a highly non-linear control system. It is therefore counter-intuitive that linear methods such as LDS [LWS02] and ICA prove to be effective tools for motion modeling and editing. 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.

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 [HKO01] algorithm that we currently use does not always converge to the globally optimal decom-

position. 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.

Motion editing is a difficult problem. We believe that our method solves another piece of the puzzle by providing a style modeling and editing tool which can be used standalone or in conjunction with other methods.

## 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. An important feature of our decomposition is that the resulting 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 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.

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. We plan to investigate such avenues in the future.

## References

- [ABC96] AMAYA K., BRUDERLIN A., CALVERT T.: Emotion from motion. In *Graphics Interface* (May 1996), pp. 222–229. 2
- [AFO03] ARIKAN O., FORSYTH D. A., O'BRIEN J. F.: Efficient synthesis of physically valid human motion. *ACM Transaction of Graphics* 22, 3 (2003), 417–426. 2
- [BH00] BRAND M., HERTZMANN A.: Style machines. In *Siggraph 2000, Computer Graphics Proceedings (2000)*, Akeley K., (Ed.), ACM Press / ACM SIGGRAPH / Addison Wesley Longman, pp. 183–192. 2
- [BW95] BRUDERLIN A., WILLIAMS L.: Motion signal

- processing. *Proceedings of ACM SIGGRAPH: Computer Graphics* (August 1995), 97–104. 2
- [CDB02] CHUANG E., DESHPANDE H., BREGLER C.: Facial expression space learning. In *Proceedings of Pacific Graphics* (2002). 3
- [CFP03] CAO Y., FALOUTSOS P., PIGHIN F.: Un-supervised learning for speech motion editing. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), Eurographics Association, pp. 225–231. 1, 3, 5
- [FP03] FANG A. C., POLLARD N. S.: Efficient synthesis of physically valid human motion. *ACM Transaction of Graphics* 22, 3 (2003), 417–426. 2
- [FvT01] FALOUTSOS P., VAN DE PANNE M., TERZOPOULOS D.: Composable controllers for physics-based character animation. In *Proceedings of ACM SIGGRAPH 2001* (2001), pp. 251–260. 3
- [Gle98] GLEICHER M.: Retargeting motion to new characters. In *Proceedings of SIGGRAPH 98* (July 1998), Computer Graphics Proceedings, Annual Conference Series, pp. 33–42. 2, 5
- [Gle01] GLEICHER M.: Motion path editing. In *Proceedings of 2001 ACM Symposium on Interactive 3D Graphics* (2001), pp. 195–202. 2
- [Gra98] GRASSIA F. S.: Practical parameterization of rotations using the exponential map. *J. Graph. Tools* 3, 3 (1998), 29–48. 4
- [HGSH98] HURRI J., GÄVERT H., SÄRELÄ J., HYVÄRINEN A.: Fastica. Helsinki University of Technology, Laboratory of Computer Information Science, Neural Networks Research Centre, [www.cis.hut.fi/projects/ica/fastica/](http://www.cis.hut.fi/projects/ica/fastica/), 1998. 3
- [HJS03] HYUN JOON SHIN LUCAS KOVAR M. G.: Physical touch-up of human motion. In *Pacific Graphics* (2003), pp. 194–203. 2
- [HKO01] HYVÄRINEN A., KARHUNEN J., OJA E.: *Independent Component Analysis*. John Wiley & Sons, 2001. 3, 7
- [HWBO95] HODGINS J. K., WOOTEN W. L., BROGAN D. C., O'BRIEN J. F.: Animating human athletics. In *Proceedings of SIGGRAPH 95* (Aug. 1995), Computer Graphics Proceedings, Annual Conference Series, pp. 71–78. 3
- [KB96] KO H., BADLER N.: Animating human locomotion with inverse dynamics. *IEEE Computer Graphics and Applications* (March 1996), 50–59. 2
- [KGP02] KOVAR L., GLEICHER M., PIGHIN F.: Motion graphs. *ACM Transactions on Graphics* 21, 3 (July 2002), 473–482. 2
- [KSG03] KOVAR L., SCHREINER J., GLEICHER M.: Footskate cleanup for motion capture editing. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), Eurographics Association, pp. 97–104. 4
- [LCR\*02] LEE J., CHAI J., REITSMA P., HODGINS J., POLLARD N.: Interactive control of avatars animated with human motion data, 2002. 2
- [LP02] LIU C. K., POPOVIĆ Z.: Synthesis of complex dynamic character motion from simple animations. *ACM Transactions on Graphics* 21, 3 (July 2002), 408–416. 2
- [LvdPF96] LASZLO J. F., VAN DE PANNE M., FIUME E.: Limit cycle control and its application to the animation of balancing and walking. *Proceedings of SIGGRAPH 96* (August 1996), 155–162. 3
- [LWS02] LI Y., WANG T., SHUM H.-Y.: Motion texture: A two-level statistical model for character motion synthesis. *ACM Transactions on Graphics* 21, 3 (July 2002), 465–472. 2, 7
- [Mat] MATHWORKS: Matlab software package. 3
- [MH02] MORI H., HOSHINO J.: Independent component analysis of human motion. In *Proceedings of International Conference on Acoustics Speech and Signal Processing* (2002), vol. VI, pp. 3564–3567. 3
- [PB02] PULLEN K., BREGLER C.: Motion capture assisted animation: Texturing and synthesis. *ACM Transactions on Graphics* 21, 3 (July 2002), 501–508. 2
- [PW99] POPOVIC Z., WITKIN A.: Physically based motion transformation. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 11–20. 2
- [RCB98] ROSE C., COHEN M. F., BODENHEIMER B.: Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications* 18, 5 (1998), 32–41. 2
- [SK83] SANKOFF D., KRUSKAL J. B.: *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. CSLI Publications, 1983. 5
- [SLGY01] SHIN H. J., LEE J., GLEICHER M., Y. S. S.:



- Computer puppetry: and importance-based approach. *ACM Transaction of Graphics* 20, 2 (2001), 67–94. [2](#)
- [TH00] TANCO L. M., HILTON A.: Realistic synthesis of novel human movements from a database of motion capture examples. In *IEEE Workshop on Human Motion* (2000). [2](#)
- [UAT95] UNUMA M., ANJYO K., TAKEUCHI R.: Fourier principles for emotion-based human figure animation. *Computer Graphics* 29, Annual Conference Series (1995), 91–96. [2](#)
- [WK88] WITKIN A., KASS M.: Spacetime constraints. *Proceedings of ACM SIGGRAPH: Computer Graphics* 22, 4 (August 1988), 159–168. [2](#)
- [WP95] WITKIN A. P., POPOVIĆ Z.: Motion warping. In *Proceedings of SIGGRAPH 95* (Aug. 1995), Computer Graphics Proceedings, Annual Conference Series, pp. 105–108. [2](#), [5](#)
- [ZH02] ZORDAN V. B., HODGINS J. K.: Motion capture-driven simulations that hit and react. In *ACM SIGGRAPH Symposium on Computer Animation* (July 2002), pp. 89–96. [3](#)



**Figure 7:** A walking motion (left) is given a sneak-like style (2nd to left). The original sneaking motion (right) is stylized with an upright, walking style (2nd to right). Frames from top to bottom.