

Yong Cao · Ari Shapiro · Petros Faloutsos · Frédéric Pighin

Motion Editing With Independent Component Analysis

the date of receipt and acceptance should be inserted later

Abstract We propose a novel method to edit motion data interactively based on motion decomposition. Our method employs *Independent Component Analysis* to decompose motion data into linear components that capture different aspects of the original motion. In this approach, 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 synthesize new motions, as well as to amplify or reduce the appearance of undesirable aspects in one motion. Motion can be synthesized quickly and visually. 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.

We propose a novel method for editing interactively motion data based on motion decomposition. Our method employs Independent Component Analysis to decompose motion data into linear components that capture different aspects of the original motion. In this approach, 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 synthesize new motions, as well as to amplify or reduce the appearance of undesirable aspects in one motion. Motion can be synthesized quickly and visually. Our approach is simple, efficient and intuitive since the compo-

nents are themselves motion data. We demonstrate that the proposed method can serve as an effective tool for interactive motion analysis and editing.

1 Introduction

Animating a digital object is often quite costly whether it is animated by hand using key-framing, with a motion capture system, or through some kind of physical simulation. As a result, much work is being done to develop techniques for editing motions. Thanks to these techniques a motion can be adapted to fit different situations and much time can be saved by reusing existing motions. Many editing techniques have been developed; each providing a different way to modify motions. For instance, some techniques are local and affect a small number of frames while others are global and affect the whole motion. Motion editing techniques have been developed for a wide variety of motion such as facial motions, full-body motions, and more recently fluid motions. However, most techniques only apply to a particular type of motion so that a system developed for editing facial motions is usually not appropriate for manipulating full-body motions. The technique we describe in this paper does not suffer from this drawback since it can be used for editing facial, full-body, and fluid motions. This is possible because our technique does not assume any specific parameterization of the input motions.

Our approach is based on a statistical modelling technique, called *Independent Component Analysis* (ICA), that decomposes motion data into linear combinations of independent components. These independent sources can then be modified through a set of editing operations, then recombined in order to synthesize a new motion. The decomposition yields components that can be intuitively interpreted without the need to understand frequency bands or other low-level concepts. ICA is similar to *Principal Component Analysis* (PCA) in the sense that it provides a dimensionality reduction of the data. However, the independent components have different semantics than the principal components and, often, they have more intuitive meaning. For instance, for

Yong Cao
Virginia Polytechnic Institute and State University, Department of
Computer Science
E-mail: yongcao@vt.edu

Ari Shapiro
University of California at Los Angeles, Department of Computer Science
E-mail: ashapiro@cs.ucla.edu

Petros Faloutsos
University of California at Los Angeles, Department of Computer Science
E-mail: pfal@cs.ucla.edu

Frédéric Pighin
Industrial Light and Magic
E-mail: fpighin@ilm.com

facial animation, a component can be associated with the motion of the eyebrows, whereas another one might control the amount of “happiness” presented in the motion. Moreover, since the resulting motion is a linear combination of these components, the modification of a component has an intuitive impact on the resulting motion.

The independent components are generated automatically without human intervention since ICA is an unsupervised learning technique. The animator then synthesizes new motions by performing a set of editing operations on the independent components. These operations include combining components, scaling individual components, substituting one component for another, and so on. Depending on the operations, the motion can be edited globally across every frame of the motion or locally over several consecutive frames. The spatial extent of the modification varies depending on the semantics of the components. Modifying some component might have a local spatial effect (e.g. a component associated with the motion of the mouth) whereas others might have a global effect (e.g. a component that affects the style of the entire face).

Our technique is different from other editing techniques which give the animator control over predefined parameters such as joint angles for full-body motion. Instead, our technique extracts a parameterization based on the different modes of variation found in the input data. This parameterization is based on the notion of statistical independence. Our work demonstrates that statistical independence is a good criterion for parameterizing motion data. The semantics of the different parameters are visually identified by the animator or analyzed by pre-defined numerical processes. While a skilled animator can use key-framing to obtain a specific animation, our technique is more coarsely-grained, operating on an intuitive and low dimension parameterization of the motions that requires little or no animation skill. Also, our editing approach differs from methods such as frequency band analysis, which provides no intuition on how modifying one band would affect the appearance of the resulting motion. Common motion editing methods and their advantages and disadvantages are given in Table 1.

Another advantage of our technique is that it is fast and suitable for interactive applications. After the interactive manipulation of the components, the synthesis phase completes within seconds, since it only requires linear operations.

The remainder of the paper is organized as follows. Section 2 provides an overview of related work on motion editing and synthesis in both facial and full-body animation. Section 3 introduces ICA. Section 4 describes how we apply ICA to different types of motion data in order to generate components that represent style, content, emotion and other aspects of motion. Section 5 describes the editing operations that can be performed on these components. Section 6 gives an example of interactively extracting the style from a full-body animation and applying it to another motion. Section 7 presents our results using facial animation, full-body animation and particle-based animations as well as a discussion of the limitations of our approach. Section 8 discusses several

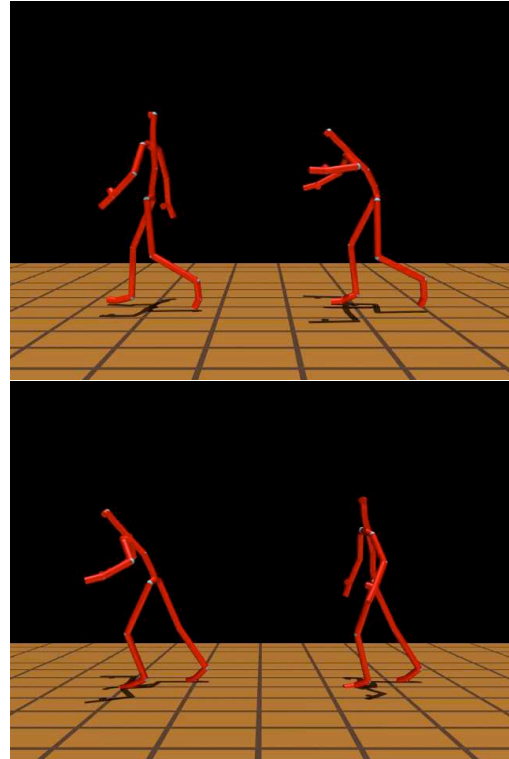


Fig. 1 Our editing technique allows animators to decompose motion into components that represent style, content and emotion. The extracted styles can then be reapplied in order to synthesize new motion. In the above figure, we edit a walking motion (top left) by adding a *sneaky* component in order to synthesize a *sneaky walk* (top right). We can also reverse the style transfer by adding a *walking* component to a sneaking motion (bottom left), yielding what appears to be a *tiptoeing* motion (bottom right).

issues involved in this paper. Lastly, Section 9 concludes the paper and discusses future work.

2 Related Work

Our study focuses on motion editing and synthesis in the context of facial animation, full-body animation and other sources of motion data. Previous work in these fields has often been specialized toward one of those motion categories. As such, we separate our review of related work into those related to facial animation, full-body animation and generic motion editing techniques.

2.1 Facial Animation - Editing & Synthesis

Two most important parts of facial animation are speech motion and facial expression. Recent works on facial animation are primarily related to the problems of how to edit and synthesize one of these two motion components.

Table 1 Common Motion Editing Techniques

Editing Method	Advantages	Disadvantages
Key-framing	visual, fine-grain control	tedious, requires expert user for good results
Inverse Kinematics	visual, high level control over specific aspects of motion	coarse-grained control, does not control all aspects of motion
Frequency band manipulation	low-level control	no intuitive representation of motion
ICA component editing	visual, high-level control over abstract qualities of motion	coarse-grained control, does not always allow control over specific aspects of motion

2.1.1 Speech motion

Most speech animation systems exploit the fact that speech can be reliably segmented into units (e.g. phonemes). The voice track is manually [39] or automatically [33,7] segmented into phonemic representations which are then mapped to lip-shapes. Of particular importance here is the problem of *co-articulation*. Co-articulation means that the mouth shape used to produce a particular phoneme depends not only on the current phoneme but also on the phoneme before and after the current one. *Hidden Markov Models* (HMM) have been used extensively to represent transitions between phonemic representations with proper co-articulation.

The synthesis of speech animation requires a library of lip-shapes that can be matched with speech units. This library can be designed in several ways. One option is to create manually each shape. For realistic animation it however preferable to record these shapes using video or motion capture data. *Video Rewrite* [7] is a representative example of such techniques. It constructs a large database of audiovisual basis units, based on triphones. Given a novel input utterance, the corresponding facial motion is constructed by concatenating the appropriate triphones from the database. In order to be useful the method requires a large database of triphones, which leads to a scaling problem. To eliminate the need for large example databases, a statistical face motion can be estimated from the data. *Voice Puppetry* [5] develops a mapping from voice to face by learning a model of a face's observed dynamics. The model takes into account the position and the velocity of facial features and learns a probability distribution over the different facial configurations. Ezzat et al [17] develop a *Multidimensional Morphable Model* for the voice to face mapping focusing on lip-syncing. Head and upper face motion is dealt with in an ad hoc fashion.

2.1.2 Facial expression and emotion

While the previous techniques can generate high quality speech motion, they generally do not provide the animator with intuitive control over the emotional state of the talking face. They focus on the mapping of the audio and visual speech signal and effects such as co-articulation. In contrast, our work develops an unsupervised learning approach that learns two separate mappings, one between the phonemic content

of the audio signal and the motion of the face and another between the audio signal and the emotional content of the speech.

Motion capture allows the recording a high fidelity motions from live actors. This technique spurred a wealth of research efforts in motion analysis.

Chuang et al [11] present an interesting attempt to separate visual speech into content and style (emotion). Their method based on factorization [22,46] produces a bilinear model that extracts emotion and content from input video sequences. However, their approach normalizes the signals losing important temporal information and it is tailored to video data. It is not clear whether it would transfer to 3D.

The pattern recognition community has performed a significant amount of work on facial expression analysis. Expressions are typically based on tracking the motion of particular facial elements such as the eyes, the rigid body motion of the face, or transient features such as wrinkles [15, 14, 16, 13, 4, 35, 3]. These systems are quite effective for recognition, however, it is not clear how they can be used to synthesize or edit facial motion.

2.2 Full-body Motion - Editing & Synthesis

Motion capture systems and recorded data are 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 characters might need to perform. Motion synthesis 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 a 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 [2, 34, 32, 29] 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 [34] 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. [8] apply signal processing operations to provide stylistic variations of the original motion. [47] use Fourier decomposition to change the style of human gaits. [1] extracts emotion by analyzing neutral and non-neutral motions. Using a small number of key-frames [50] warps motion to satisfy new constraints. [21] also proposes an interesting warping technique. The goals of our method are very similar to those outlined in [47] and [1]. 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 [1]. We extract style components from dissimilar motions as well.

C. Rose et. al. [42] 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 [20,44] mainly solve the problem of motion retargeting, applying motions to characters with different body proportions effectively changes some aspects of the original motion.

M. Vasilescu et. al. [48] uses multilinear analysis to extract stylistic aspects of motion from three different actors and reapply the style to each other. Their technique is based on applying PCA in multiple but pre-defined dimensions and produces a tensor form of the original data.

M. Brand et. al. and K. Pullen et. al. [6,45] 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. [41] 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 [49, 40, 19, 28, 36, 26] 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 [24, 31, 18] and hybrid methods [51].

Within the domain of statistical modelling, of particular relation to our work are the techniques that provide editing parameters through motion decomposition. [12] propose a factorization method that separates visual speech into style and content components. [10] uses *Independent Component Analysis* to capture the emotional content of visual speech for editing purposes. They 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 Independent Component Analysis for editing and synthesizing human walking has been proposed in [38].

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 modelling 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.

3 Independent Component Analysis

Independent Component Analysis is an unsupervised learning technique [27] 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*. [10] 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.

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

$$\mathbf{x} = \mathbf{A}\mathbf{u}, \quad (1)$$

where $\mathbf{x} = [x_1, \dots, x_n]^T$ and $\mathbf{u} = [u_1, \dots, u_n]^T$.

A number of ICA algorithms exist to estimate the mixing matrix \mathbf{A} . Estimating \mathbf{A} is sufficient, because if the matrix is known, inverting Equation 1 yields the independent components $\mathbf{u} = \mathbf{W}\mathbf{x}$. We use the publicly available Matlab[37] implementation of the FastICA [25] algorithm.

Applying ICA involves a two stage pre-processing. First, the data is centered around its statistical mean $E\{\mathbf{x}\}$. Then the centered data is decomposed into a set of uncorrelated variables, typically using Principal Components Analysis. The complete model is as follows:

$$\mathbf{x} = E\{\mathbf{x}\} + \mathbf{P}\mathbf{A}\mathbf{u}, \quad (2)$$

where $E\{\mathbf{x}\}$ is the expectation of \mathbf{x} and \mathbf{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.

4 Applying ICA to Motion

There are two main ways that we have applied ICA to motion data. We have formatted a single data set into a representation appropriate for decomposition and then executed the ICA algorithm. By decomposing a single motion, we attempted to parameterize the motion into independent components. These components can be recombined to form the original motion.

We have also aligned two different motions, M_1 and M_2 , with similar structure and run ICA algorithm, which produced components that represent the differences between the two motions.

The former technique is useful for editing motion that cannot be correlated with another motion. For example, motion represented simulated fluids suffers from a registration problem: particles from one fluid motion cannot be effectively correlated with particles of another motion since it is not known which particles correspond exactly to each other. By contrast, motion capture data of facial animation can be correlated since the markers of one motion capture session can be associated with the same region as another session. Barring an exact position match of the markers, the facial animation can also be retargeted so that the points correspond to the same relative location on the face. This is also true for full-body motion data, which can be retargeted in order to match joint and bone positions.

4.1 Data Representation

Applying ICA to motion data is straightforward. A motion, M , is represented as a time series $\mathbf{x}(t)$ that typically indicates the Euclidean coordinates of motion capture markers in time. For simulated motion, such as fluids, smoke or other particle data, $\mathbf{x}(t)$ represents particle positions over time. However, the data could also be a representation appropriate for the type of system modelled. For example, we could use joint angles instead of Euclidean coordinates when decomposing full-body motion that has been fit onto a hierarchical skeleton.

The time series can be interpreted as a set of samples of random variable, \mathbf{x} , which can be substituted into equation 1. For this reason, a motion M can be represented by both time-series $\mathbf{x}(t)$ and random variable \mathbf{x} . A frame of motion M at time t_i can be represented as $\mathbf{x}(t_i)$.

4.2 Decomposition Using Different Representations

For full-body motion capture data, 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 (if appropriate), or the 3) quaternions that represent the rotation of the joints. Motion data for facial animation and particle data do not have hierarchical

structures, so these data sets must use the point representation.

Since the ICA algorithm results in a linear decomposition of the input data, decomposing data consisting of Euler angles will produce visually unintuitive results. 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. [23] 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. Depending on the data, however, the synthesized motion may violate constraints of the system. For example, for full-body motion data, the ICA decomposition and subsequent synthesis can result in a change in the length of an animated character's limbs, since the point representation does not preserve the implicit constraints of the system, such as distance between joints.

We used the point representation for most of our experiments and for most of the different types of motion data. When using full-body motion data, we are concerned only with kinematic animation and the visual quality of the final animation, and 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 [30].

Similarly, our ICA editing tool can produce exaggerated expressions when used with facial animation data, such as an exaggerated smile or eyebrows that are raised too high. Since our tool relies upon visual confirmation of the results, the animator can discard motion that contains these undesirable artifacts. However, our tool and editing method has no automatic way to enforce such constraints.

4.3 Interpreting Components

There are two ways we can detect the semantic meaning of each independent component: checking visually or analyzing numerically.

The components can be interpreted visually by treating them as motion data. Although each component only represents part of the original motion, a visual representation of the components gives clues to the animator as to their effect. Indeed, components with distinct characteristics can be easily detected by visual inspection. For example, a component derived from facial animation that mainly raises the eyebrows will appear to the user as a non-expressive face that raises and lowers the eyebrows as the motion is replayed. Many components do not have a clearly definable function as just described, however. The user must perform editing operations on these components in order to better understand their impact on the final operation. These operations are discussed in section 5.

To visually interpret the meaning of a independent component u_i , we reconstruct motion M , represented by \mathbf{x}_i , from only this component by using Equation 3.

$$\mathbf{x}_i = E\{\mathbf{x}\} + \mathbf{P}\mathbf{A} \begin{bmatrix} 0 \\ 0 \\ u_i \\ \dots \\ 0 \end{bmatrix}. \quad (3)$$

The motion \mathbf{x}_i can be thought as the visual representation of component u_i . Although each component only represents part of the original motion, a visual representation of the components gives clues to the animator as to their effect. Indeed, components with distinct characteristics can be easily detected by visual inspection. For example, a component derived from facial animation that mainly raises the eyebrows will appear to the user as a non-expressive face that raises and lowers the eyebrows as the motion is replayed. Many components do not have a clearly definable function as just described, however. The user must perform editing operations on these components in order to better understand their impact on the final operation.

The semantics of a component can also be interpreted by numerical analysis. In order to give a detailed description of this process, we use the following facial motion case as an example. In this example, we decompose facial motion data into several components. We want to associate each component either with style (emotion) or content (speech). In what follows we describe how we associate specific meaning to the independent components.

4.3.1 Emotion

We recorded the motion of an actor’s face while he was uttering a set of sentences multiple times, each time expressing a different emotion. Let us denote as $(\mathbf{x}^i, \mathbf{y}^i)$, p pairs of motions that corresponds to the same sentence but two different emotions. Applying ICA to each pair of motions in

our dataset, results into pairs of corresponding independent component sets, $(\mathbf{u}^i, \mathbf{v}^i)$. We would expect that the independent components related to emotion differ significantly between two speech motions that have the same content but different emotion. In contrast, if an independent component is not related to emotion, its value in time for two corresponding motions should be the same except some timing differences. In order to verify this property, we align each pair of corresponding motions using a *Dynamic Time-Warping* (DTW) algorithm[43]. Let us denote $(\mathbf{u}^i, \mathbf{v}^i)$ the independent components of two aligned motions after time warping. We compute their difference using the Root Mean Square (RMS) error as follows:

$$d_{emotion,j} = \left(\frac{1}{\sum q_i} \left(\sum_{i=1}^p \left(\sum_{k=1}^{q_i} (u_j^i(t_k) - v_j^i(t_k))^2 \right) \right) \right)^{\frac{1}{2}},$$

where q_i is the number of aligned time samples for pair i . The distance $d_{emotion,j}$ is designed such that it should be large if component j is related to emotion.

Figure 2(a) shows a plot of the $d_{emotion,j}$ values of 6 independent components estimated from 32 pairs of sentences of *Frustrated* and *Happy* motions. This data totals 11883 frames or 99 seconds. A clear peak can be observed for the third component. This strongly indicates that this component is related to emotional variations. The other components participate to a lesser degree to the emotional content of the motions. This shows that speech motion cannot be strictly separated into statistically independent components. Our approach is albeit a successful approximation. As further proof, in Figure 3 we plot the evolution of the different components over time for a set of five pairs of motions. On the timeline, we alternate *Frustrated* and *Happy* motions. The behavior of the third component appears very much related to changes in emotions (illustrated with different gray levels).

4.3.2 Content

We define content as the part of the motion associated with the formation of speech independent of expressiveness. For this case we only consider the motion of the markers in the mouth area (12 markers in our dataset).

Let us define a distance metric between two motions that have been reconstructed using two subsets of independent components, A and B .

$$d_{mouth}(\mathbf{x}_A, \mathbf{x}_B) = \left(\frac{1}{q} \sum_{k=1}^q \left(\frac{1}{r} \sum_{l=1}^r (x_A^l(t_k) - x_B^l(t_k))^2 \right) \right)^{\frac{1}{2}} \quad (4)$$

where \mathbf{x}_A and \mathbf{x}_B are the motions reconstructed using component subset A and B respectively, q is the number of time samples of both motions, r is the number of the markers considered for the mouth region (12 markers).

Reconstructing the motion of the mouth markers using all the independent components produces \mathbf{x}_{all} . In general this is different from the captured motion because of the compression done in the preprocessing step (Section 3). In order

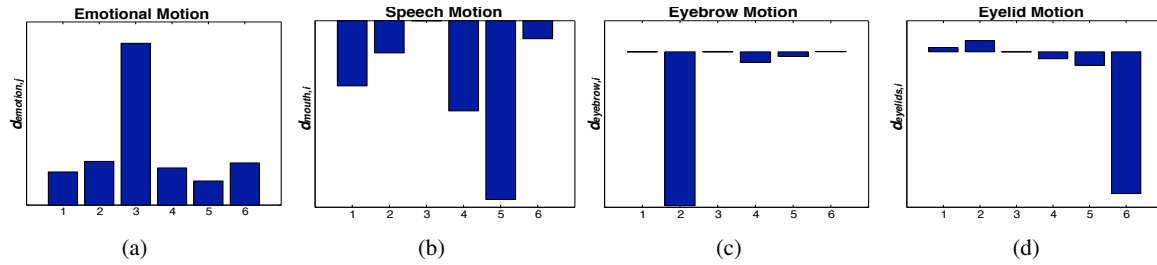


Fig. 2 These graphs illustrate the classification of independent components. Each graph illustrates a category of motion: (a) for emotions, (b) for speech, (c) for eyebrows, and (d) for eyelids. The horizontal axis represents the index of independent components. The vertical axis shows the distance metrics that we described in Equation 4.

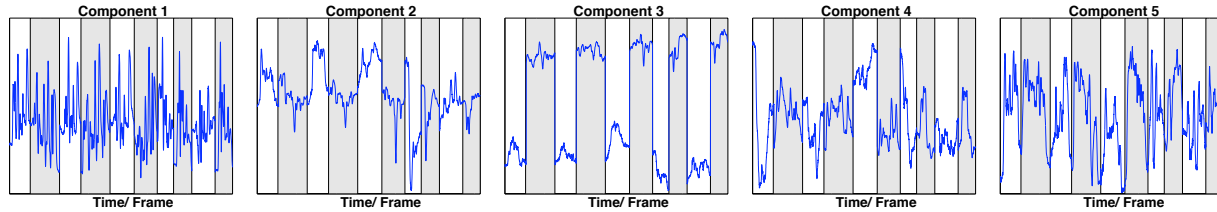


Fig. 3 These graphs present the evolution in time of five independent components corresponding to five pairs of *Frustrated* and *Happy* motions. The timeline alters between Frustrated (light) and Happy (dark) motions. Notice how the third component oscillates between extreme values when the emotion changes.

to evaluate how much independent component i contributes to the mouth motion we compute the following metric

$$d_{mouth,i} = d_{mouth}(\mathbf{x}_{E \cup \{i\}}, \mathbf{x}_{all}) - d_{mouth}(\mathbf{x}_E, \mathbf{x}_{all}) \quad (5)$$

where E is the subset of independent components responsible for emotion and $vecx_E$ is the marker motion reconstructed from subset E .

In Equation 5 $d_{mouth,i}$ quantifies the influence of independent component i on the motion of the mouth. The larger in absolute value this number is, the more influence component i has over the mouth motion. Figure 2(b) shows the value of $d_{emotion,i}$ for six independent components. Notice how large $d_{mouth,1}$, $d_{mouth,4}$, and $d_{mouth,5}$ are compared to the rest of the components. We can visually verify that the motion $x_{\{1\} \cup \{4\} \cup \{5\}}$ reconstructed using components 1, 4 and 5 captures most of the speech motion.

4.3.3 Blinking and non-emotional eyebrow motions

Our experiments show that some independent components cannot be associated with emotion or content. We have experimentally determined that we can further classify such components into two groups: one for blinking motion and the other for non-emotional eyebrow motion. The later refers to eyebrow motion that reflects stress and emphasis in the speech rather than the emotional state of the speaker.

In order to identify the components related to these two types of motion we use the same method employed for finding content related components. We define $d_{eyebrow}$ and $d_{eyelids}$ according to Equation 4 while considering only the markers on the eyebrows and the eyelids respectively. We use these two metrics to define $d_{eyebrow,i}$ and $d_{eyelids,i}$ from Equation 5 for the eyebrows and the eyelids respectively.

Figure 2(c) shows the value of the distance metric $d_{eyebrow,i}$ for six independent components. Notice how much larger $d_{eyebrow,2}$ is compared to the distance metric of the rest of the components. Clearly component 2 captures most of the eyebrow motion. Similarly, Figure 2(d) shows the value of the distance metric $d_{eyelids,i}$ for each of the six components. In this case, $d_{eyelids,6}$ dominates the rest of the components. We conclude that component 6 captures most of the eyelid motion.

5 ICA Editing

Before we describe the intuitive editing operations in ICA space, we like to introduce a set of notations. These notations can simplify the description of the editing operations and make the representation of the operations easier to understand.

As presented in Section 4.1, a motion can be represented by time series $\mathbf{x}(t)$ or random variable \mathbf{x} . After applying ICA to the motion, its corresponding independent components can be represented by time series $\mathbf{u}(t)$ or random variable \mathbf{u} , where $\mathbf{u} = [u_1, \dots, u_m]^T$. We represent an independent component u_i , ($1 \leq i \leq m$), by a vector \mathbf{u}_i ,

$$\mathbf{u}_i = u_i \mathbf{e}_i, (1 \leq i \leq m),$$

where \mathbf{e}_i is the vector in the canonical basis of the ICA mixing matrix that corresponds to the i th component. A frame of independent component \mathbf{u}_i at time t_j is represented as $\mathbf{u}_i(t_j)$. Given this vector representation, the independent components \mathbf{u} can be described as linear combination of every independent component vector \mathbf{u}_i ,

$$\mathbf{u} = \mathbf{u}_1 + \dots + \mathbf{u}_m.$$

5.1 Editing Operations

Now we can present ICA editing operations we apply to motion data. These operations are very intuitive and efficient which can be used in any interactive application. We also encourage readers to develop addition operations based on the requirement of their applications.

5.1.1 Scaling

The new component, \mathbf{u}'_i , can be generated by multiplying a scalar α by the original component \mathbf{u}_i . Scaling is used in order to amplify or exaggerate an effect. The trivial scaling, $\alpha = 0$, is another editing operation *zeroing*.

$$\mathbf{u}'_i = \alpha \mathbf{u}_i.$$

After scaling a single component without changing the other other components, the resulting motion \mathbf{x}' can be expressed as:

$$\mathbf{x} = E\{\mathbf{x}\} + \mathbf{PA}(\mathbf{u} + (\alpha - 1)\mathbf{u}_i).$$

5.1.2 Addition

A component vector may be added to another component in order to combine two components together.

$$\mathbf{u}'_i = \mathbf{u}_j + \mathbf{u}_k \quad (6)$$

Where \mathbf{u}'_i is the synthesized component, \mathbf{u}_j and \mathbf{u}_k are the two components that are combined. Components that are automatically extracted from the motion often have no intuitive semantic meaning. By combining components together, the user can better visualize the meaning of the new component in the context of the motion. The Addition operation does not change the resulting motion.

The *Addition* operation in combination with the *Scaling* operation allows the user to create linear combinations of components:

$$\mathbf{u}'_i = \alpha \mathbf{u}_j + \beta \mathbf{u}_k \quad (7)$$

5.1.3 Translation

Translation operation allows us to add a constant value, c , to an independent component \mathbf{u}_i .

Editing can be expressed as:

$$\mathbf{u}'_i = \mathbf{u}_i + c\mathbf{e}_i,$$

where c is a scalar that quantifies the amount of translation in the component.

editing.tex The edited motion \mathbf{x}' can be expressed as:

$$\mathbf{x}' = E\{\mathbf{x}\} + \mathbf{PA}(\mathbf{u} + c\mathbf{e}_i).$$

5.1.4 Replacement

Unlike the previous editing operations, the *Replacement* operation can only be applied to a single frame of the components. We can use this operation to replace the value of component \mathbf{u}_i at time frame t_1 with the value of the same component at time frame t_2 . This manipulation can then be written as follows:

$$\mathbf{u}_i(t_1) = \mathbf{u}_i(t_2).$$

The edited motion \mathbf{x}' at frame t_1 can be expressed as:

$$\mathbf{x}'(t_1) = E\{\mathbf{x}\} + \mathbf{PA}(\mathbf{u}(t_1) - \mathbf{u}_i(t_1) + \mathbf{u}_i(t_2)).$$

Using *Replacement* operation, we can select a movement from a small period of a motion. We can then replace the other period of the motion with our selected movement, in order to duplicate that movement to different places in the motion. For facial animation, we can change the emotional state of the animated face by using a component that represents the underlying emotional state of the speaker. For full-body animation, we can copy and replace a special walking style onto different parts of a walking motion.

5.1.5 Copy and Add

We can also add a component was not present in the original motion.

Let's consider two motions \mathbf{x} and \mathbf{y} . Applying ICA to motion \mathbf{x} , we can get independent component \mathbf{u} and its mixing matrix $(\mathbf{PA})_1$. Applying ICA to motion \mathbf{y} , we get independent component \mathbf{v} and its mixing matrix $(\mathbf{PA})_2$. We can add the motion represented by independent component \mathbf{v}_i to the motion \mathbf{x} . The resulting motion is expressed as:

$$\mathbf{x}' = E\{\mathbf{x}\} + (\mathbf{PA})_1\mathbf{u} + (\mathbf{PA})_2\mathbf{v}_i.$$

Note that this editing operation, unlike the *Replacement* operation described above, will not remove any aspect of the original motion.

Notice that all the editing operations we have described so far are applied to motions that are already in the training set used to estimate the ICA model. In order to edit a motion \mathbf{x} that does not belong to the training set, we can project it to extract the independent components:

$$\mathbf{u} = (\mathbf{PA})^+(\mathbf{x} - E\{\mathbf{x}_{training}\}),$$

where $^+$ indicates the pseudo-inverse of a matrix and $\mathbf{x}_{training}$ the expectation of the motions in the training set. After projection, the motion can be edited in ICA space.

5.2 Preparing Input Data

The mixing matrix \mathbf{A} and independent components \mathbf{u} in Equation 1 can be learned by applying ICA to input data. The semantics of the resulting components is fully determined by statistical independence hidden inside the input motion

data. We can not expect ICA to give us a component that has certain semantic meaning while the input training data doesn't include that type of semantics. For example, if an input facial motion doesn't have mouth movement, there is no resulting component from ICA can be related to speech motion. Therefore, during input data preparation step, firstly we need to decide what kind of semantics we are expecting. We then make sure that the input data we prepare for ICA algorithm includes those semantics.

In some cases the movement we are interested with is not presented inside input motion data. For instance, if we want to edit a walking motion by changing the walking style to running style, we will find that there is no movement of changing style in the input walking motion. Therefore, we can not have any independent component that represents style difference. In these cases, our solution is to align and combine another motion into original input motion, so that the combined motion includes the movement we are interested with. We then use the combined input motion as the training data for ICA algorithm.

A motion can be aligned and combined with another motion only if the two motions can be correlated with each other. Motion capture data of facial animation can be correlated since the markers of one motion capture session can be associated with the same region as another session. Barring an exact position match of the markers, the facial animation can also be retargeted so that the points correspond to the same relative location on the face. This is also true for full-body motion data, which can be retargeted in order to match joint and bone positions. By contrast, motion represented simulated fluids is difficult to be aligned with another motion. Because it suffers from a registration problem: particles from one fluid motion cannot be effectively correlated with particles of another motion since it is not known which particles correspond exactly to each other.

6 Style Transfer

One of the most important motion editing operations is transferring style between motions. In this section, we describe style transfer operation in detail.

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 4 summarizes our interactive motion editing approach. Our system is shown as applied to full-body motion capture data.

6.1 Editing For Full-Body Animation

The remainder of this section explains the steps depicted in the figure and enumerated here:

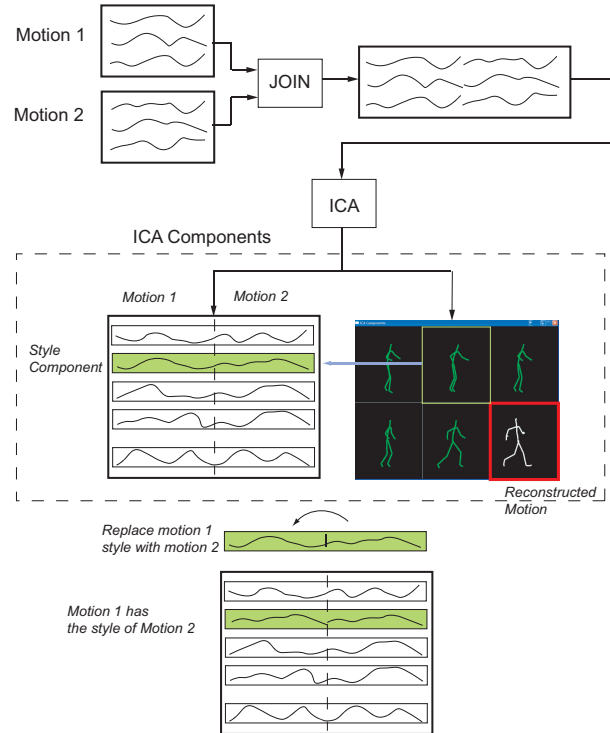


Fig. 4 Example of using our ICA-based interactive editing system. Shown above is a *Copy & Replace* operation on two motions. Using these steps, we can add the style of the second motion to the original motion. Note that the components generated from the combination of two motions are visually represented, with the final synthesized motion shown on the lower right.

1. *Pre Processing* Global translation is removed from the motions.
2. *Motion Combination*. Two motions are combined together.
3. *Component Generation*. The combined motion is decomposed into components.
4. *Style Selection*. The user selects components of interest to them.
5. *Component Merging*. The user combines components together to better represent the desired characteristics of motion.
6. *Tool Editing*. Components may be edited with standard motion editing tools.
7. *Transferring Style*. The selected components are transferred in order to create a newly synthesized motion.
8. *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.

6.2 Pre Processing

The global translation DOF are removed before the ICA decomposition since the decomposition has no intrinsic knowl-

edge 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.

6.3 Motion Combination

Given two motions, M_a and M_b , motion M_{ab} is produced by joining the frames of M_a and M_b . Thus, M_{ab} will have $f = f_a + f_b$ frames, where f_i is the number of frames for motion M_i . It is essential to combine the motions together in order for the ICA algorithm to find synchronized differences between the two motions.

6.4 Component Generation

Given motion M_{ab} , represented as random variable vector \mathbf{x}^{ab} , the user selects the number of components k in which to decompose \mathbf{x}^{ab} as well as a representation for the decomposition. The representation can be points, quaternions or Euler angles, see Section 4. Applying the ICA algorithm results into k independent components $\mathbf{u}_1^{ab} \dots \mathbf{u}_k^{ab}$ for the combined motion \mathbf{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.

Each component \mathbf{u}_i^{ab} is used to reconstruct part of the original motion, M_{ab}^i , as follows:

$$\mathbf{x}_i^{ab} = E\{\mathbf{x}^{ab}\} + \mathbf{P}\mathbf{A}\mathbf{u}_i^{ab}, i = 1, \dots, k \quad (8)$$

and the result is displayed in a separate window, shown in the middle of Figure 4.

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 4.

6.5 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 4, 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.

6.6 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.

Mathematically, merging two components \mathbf{u}_j and \mathbf{u}_k results in a combined motion as follows:

$$\mathbf{u}_i' = \mathbf{u}_j + \mathbf{u}_k.$$

6.7 Tool 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 [50] and motion retargeting [20].

6.8 Transferring style

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

Once a style component \mathbf{u}_s^{ab} has been selected, it is split into two segments, \mathbf{u}_a^s and \mathbf{u}_b^s , that represent the style components of the original two motions, M_a^s and M_b^s . We can then align (time-warp) either the style component \mathbf{u}_s^a to \mathbf{u}_s^b or vice versa depending on which motion's timing we wish to preserve. We align the motions by applying dynamic time warping [43] 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 every frame of component \mathbf{u}_s^a with \mathbf{u}_s^b . Transferring a style component from one motion to another can be mathematically expressed as follows:

$$\mathbf{x}'(t_i) = E\{\mathbf{x}\} + \mathbf{P}\mathbf{A}(\mathbf{u}^a(t_i) - \mathbf{u}_s^a(t_i) + \mathbf{u}_s^b(t_i)), \\ t_{start} \leq t_i \leq t_{end}.$$

6.9 Post Processing

Once the final motion has been generated, the global translation from M_a , 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.

7 Results

Our editing technique can be used on many different types of motion data. Below we show the results of our experiments with facial animation, full-body animation and particle animation.

7.1 Facial Animation

We recorded facial motion using a Vicon8 optical motion capture system. We used 109 markers to sample the face geometry fairly densely. The sampling rate of the data is 120 frame/sec. To drive a 3D textured face mesh, the markers are mapped to corresponding mesh points, and the rest of the mesh is deformed using *Radial Basis Functions* [9].

In our experiments the principal components correlate the speech related mouth motion with intense emotion related eyebrow motion. In contrast, the independent components are able to separate mouth and eyebrow motion to a much more meaningful degree. The independent component that captures the mouth motion contains limited eyebrow motion. We believe that this is correct since part of the eyebrow motion is actually related to the content of the speech, for example when stressing a point. In contrast, intense eyebrow motion is clearly related to emotion and not to the content of the speech.

The proposed method provides an intuitive decomposition of facial motion that allows us to edit the apparent emotion of visual speech. Figure 5 shows 3 rendered frames from an editing session. The *neutral* and *sad* independent components are mixed with different percentages. Figure 6 shows a emotion session that change the emotional content by translating between *neutral*, *sad* and *angry*.

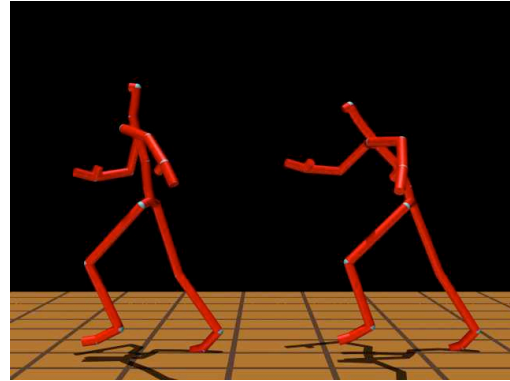


Fig. 7 Running (left) and a sneak-like run (right).

7.2 Full-body Animation

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.

7.2.1 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.

7.2.2 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 7.

7.2.3 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



Fig. 5 Three snapshots of editing visual speech. Each row shows the same speech content but different amount of *neutral* and *sad* emotions.

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 8, while our run-like walk resembles a power walk, Figure 9.

7.2.4 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 10 shows an interpolation between the sneak and the walk-like sneak (tiptoeing).

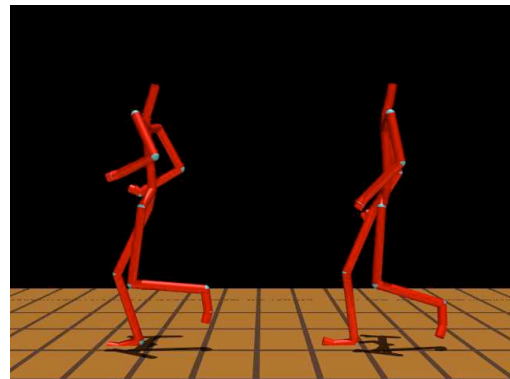


Fig. 8 Running (left) and running with a walking style - jogging (right).

7.3 Particle Motion

We performed experiments on particle data representing explosions. The purpose of our particle experiments is to extract independent components that represents various char-

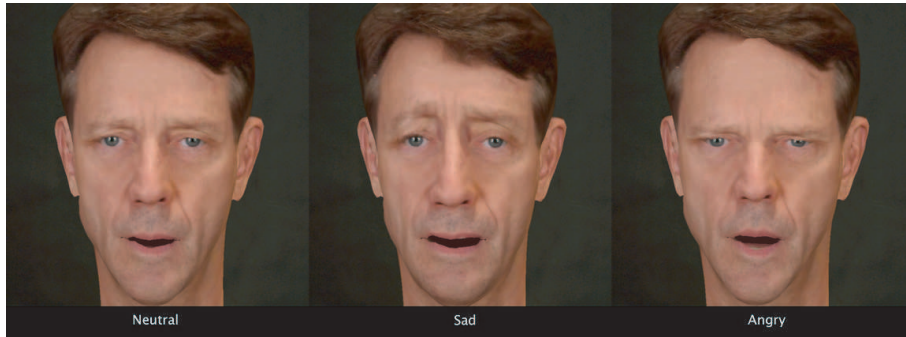


Fig. 6 The left face represents the original synthesized motion with a *neutral* expression. The middle and right faces represent the original synthesized face with the addition of a *sad* and *angry* component.

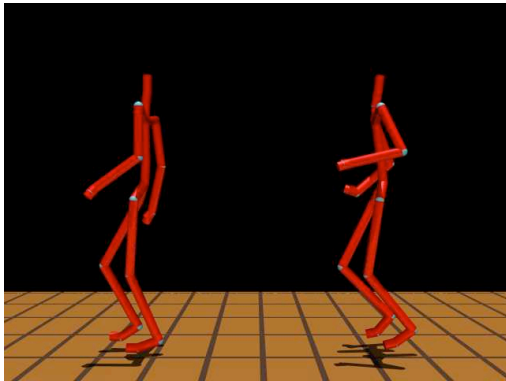


Fig. 9 Walking (left) and walking with a running style - power-walking (right).

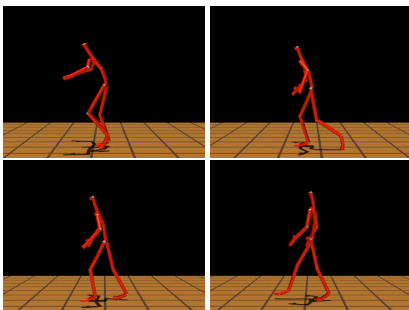


Fig. 10 Interpolating between a sneak and a walk-like sneak.

acteristics of the explosion. Like the facial animation and full-body motion experiments, we wish to obtain an effect analogous to the separation of style and content. We wish to extract various explosive characteristics, such as *burst speed of the explosion*, *expansiveness*, *mushroom-like appearance* and *explosion height*. Then using the component operations, we synthesize new explosions.

Particle-based motion data, unlike facial animation and full-body motion, suffers from a registration problem. Particles between two different motion set cannot be correlated, since the particles are themselves abstractions of a larger phenomena. Thus, it cannot be said with surety that a single particle in one explosion will exist in a different location

of a different explosion. This is in contrast to, say, facial animation, where a single marker can be correlated among different faces. Thus, our technique of combining similar motions and extracting components that represent the differences between motions will not work. However, *intercomponent* editing methods, such as *scaling* and *addition* allow us to synthesize new explosions without the need to simulate them.

Figure 11 shows a decomposition of a typical explosion. As the explosion is animated, component 1 (upper left) echoes the birth of the explosion, starting from a compact cluster of particles and then expanding those particles vertically. Components 2, 3 and 4 capture the mushroom-like folding and horizontal movement of the explosion during the later stages of the explosion. By scaling component 1, we are able to quickly synthesize a new explosion which rises higher and faster than the original explosion.

Subjective particle qualities were difficult to obtain via ICA decomposition. Burst speed and explosion girth were relatively easy to extract via the decomposition process since they can be mapped to relatively simple calculations of the data. More complicated qualities such as mushroom-like appearance could not be easily captured, in part because they do not decompose into clear, linear processes that ICA commonly generates. In addition, the authors do not know of a way to properly render an explosion without false coloring given that the density information, which is necessary for volumetric rendering, is no longer present. Future work would entail decomposing both the particle position data and the density information. However, we expect that one would get interesting results by decomposing other kinds of particle data, such as water and smoke.

8 Discussion

The human body is a highly non-linear control system. It is therefore counter-intuitive that linear methods such as LDS [34] 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 sim-

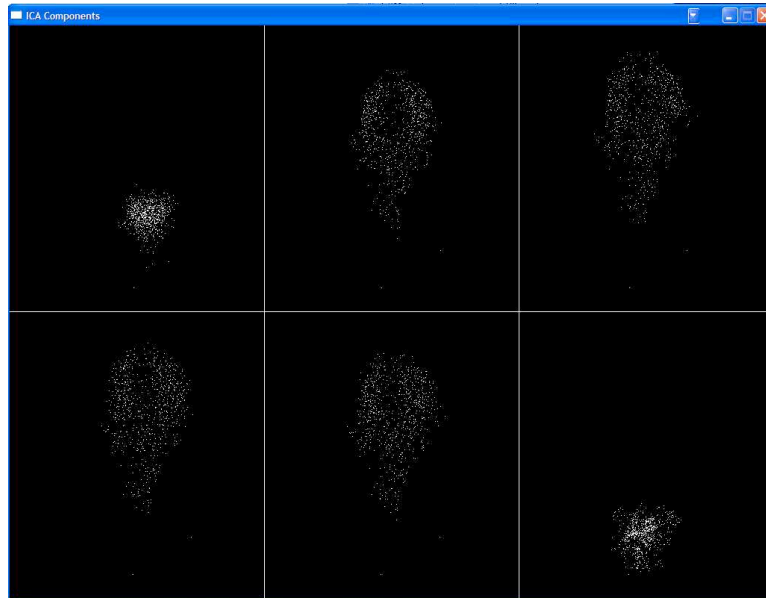


Fig. 11 An explosion is shown, represented as particles, with 5 components during the 5th frame at 33 fps. The reconstructed explosion is shown at the lower right. The first component (upper left) captures the sudden burst of the explosion in the opening frames. The other components remain relatively inert until later in the motion when the explosion expands and moves horizontally.

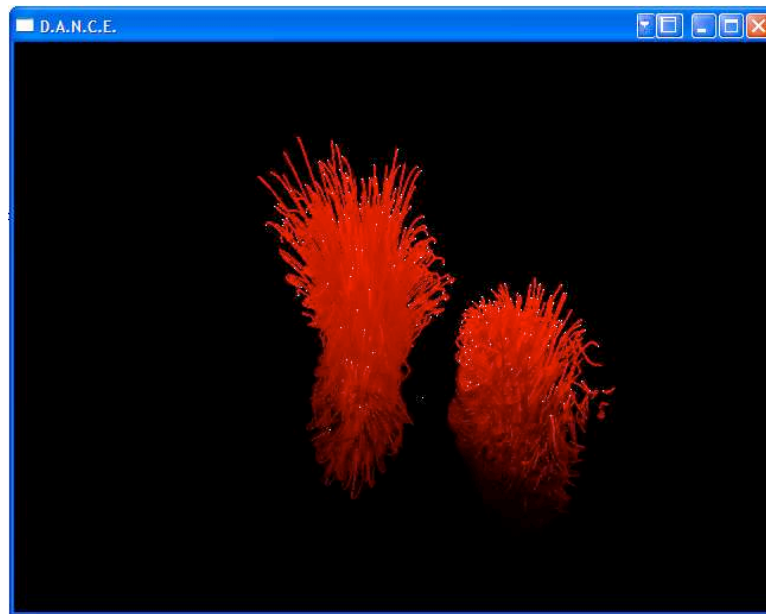


Fig. 12 By scaling one of the components that expressed the maximum height of the explosion, we synthesize a new explosion (left) that rises higher than our simulated one (right) over the same time period. The explosion trajectories are shown in red.

plifies 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 intu-

itive 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 [27] 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.

9 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.

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.

Acknowledgment

This work was partly funded by the NSF grant CCF-0429983 and the Department of the Army under contract number DAAD 19-99-D-0046. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the Department of the Army.

We would also like to thank Intel Corp. and Microsoft Corp. for their generous support through equipment and software grants.

References

- Kenji Amaya, Armin Bruderlin, and Tom Calvert. Emotion from motion. In *Graphics Interface*, pages 222–229, May 1996.
- Okan Arıkan, David A. Forsyth, and James F. O'Brien. Efficient synthesis of physically valid human motion. *ACM Transaction of Graphics*, 22(3):417–426, 2003.
- J.N. Bassili. Emotion recognition: the role of facial movement and the relative importance of upper and lower areas of the face. *Journal of Personality and Social Psychology*, 39:2049–2059, 1979.
- M. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motions. In *Proceedings, International Conference on Computer Vision*, pages 374–381, 1995.
- Matthew Brand. Voice puppetry. In *Proceedings of ACM SIGGRAPH 1999*, pages 21–28. ACM Press/Addison-Wesley Publishing Co., 1999.
- 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.
- C. Bregler, M. Covell, and M. Slaney. Video rewrite: driving visual speech with audio. In *SIGGRAPH 97 Conference Proceedings*, pages 353–360. ACM SIGGRAPH, August 1997.
- A. Bruderlin and L. Williams. Motion signal processing. *Proceedings of ACM SIGGRAPH: Computer Graphics*, pages 97–104, August 1995.
- Martin D. Buhmann. *Radial Basis Functions : Theory and Implementations*. Cambridge University Press, 2003.
- Yong Cao, Petros Faloutsos, and Francis 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.
- E. Chuang, H. Deshpande, and C. Bregler. Facial expression space learning. In *Proceedings of Pacific Graphics*, 2002.
- E. Chuang, H. Deshpande, and C. Bregler. Facial expression space learning. In *Proceedings of Pacific Graphics*, 2002.
- J. Cohn, J. Lien, A. Zlochower, and T. Kanade. Feature point tracking by optical flow discriminates subtle differences in facial expression. In *Proceedings, Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 396–401, 1998.
- I. Essa, S. Basu, T. Darell, and A. Pentland. Modeling, tracking and interactive animation of faces and heads using input from video. *Computer Animation Conference*, pages 68–79, June 1996.
- I. Essa and A. Pentland. A vision system for observing and extracting facial action parameters. In *Proceedings of IEEE Computer Vision Pattern Recognition Conference*, 1994.
- I. Essa and A. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):757–763, July 1997.
- Tony Ezzat, Gadi Geiger, and Tomaso Poggio. Trainable video-realistic speech animation. In *Proceedings of ACM SIGGRAPH 2002*, pages 388–398. ACM Press, 2002.
- Petros Faloutsos, Michiel van de Panne, and Demetri Terzopoulos. Composable controllers for physics-based character animation. In *Proceedings of ACM SIGGRAPH 2001*, pages 251–260, 2001.
- A. C Fang and Nancy S. Pollard. Efficient synthesis of physically valid human motion. *ACM Transaction of Graphics*, 22(3):417–426, 2003.
- Michael Gleicher. Retargeting motion to new characters. In *Proceedings of SIGGRAPH 98, Computer Graphics Proceedings, Annual Conference Series*, pages 33–42, July 1998.
- Michael Gleicher. Motion path editing. In *Proceedings of 2001 ACM Symposium on Interactive 3D Graphics*, pages 195–202, 2001.
- Z. Grahramani and M. I. Jordan. Factorial hidden markov models. *Machine Learning*, (29):245–275, 1997.
- F. Sebastin Grassia. Practical parameterization of rotations using the exponential map. *J. Graph. Tools*, 3(3):29–48, 1998.
- Jessica K. Hodgins, Wayne L. Wooten, David C. Brogan, and James F. O'Brien. Animating human athletics. In *Proceedings of SIGGRAPH 95, Computer Graphics Proceedings, Annual Conference Series*, pages 71–78, August 1995.
- Jarmo Hurri, Hugo Gävert, Jaakko Särelä, and Aapo Hyvärinen. Fastica. Helsinki University of Technology, Laboratory of Computer Information Science, Neural Networks Research Centre, www.cis.hut.fi/projects/ica/fastica/, 1998.
- Michael Gleicher Hyun Joon Shin, Lucas Kovar. Physical touch-up of human motion. In *Pacific Graphics*, pages 194–203, 2003.

27. A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.
28. H. Ko and N. I. Badler. Animating human locomotion with inverse dynamics. *IEEE Computer Graphics and Applications*, pages 50–59, March 1996.
29. Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. *ACM Transactions on Graphics*, 21(3):473–482, July 2002.
30. 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.
31. Joseph F. Laszlo, Michiel van de Panne, and Eugene Fiume. Limit cycle control and its application to the animation of balancing and walking. *Proceedings of SIGGRAPH 96*, pages 155–162, August 1996.
32. J. Lee, J. Chai, P. Reitsma, J. Hodgins, and N. Pollard. Interactive control of avatars animated with human motion data, 2002.
33. J. P. Lewis. Automated lip-sync: Background and techniques. *Journal of Visualization and Computer Animation*, 2:118–122, 1991.
34. 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.
35. J. Lien, J. Cohn, T. Kanade, and C.C. Li. Automatic facial expression recognition based on FACS action units. In *Proceedings, Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 390–395, 1998.
36. C. Karen Liu and Zoran Popović. Synthesis of complex dynamic character motion from simple animations. *ACM Transactions on Graphics*, 21(3):408–416, July 2002.
37. Mathworks. Matlab software package.
38. 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.
39. F. I. Parke. A model for human faces that allows speech synchronized animation. *Journal of Computers and Graphics*, 1(1):1–4, 1975.
40. Zoran Popovic and Andrew Witkin. Physically based motion transformation. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 11–20. ACM Press/Addison-Wesley Publishing Co., 1999.
41. Katherine Pullen and Christoph Bregler. Motion capture assisted animation: Texturing and synthesis. *ACM Transactions on Graphics*, 21(3):501–508, July 2002.
42. Charles Rose, Michael F. Cohen, and Bobby Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications*, 18(5):32–41, 1998.
43. David Sankoff and Joseph B. Kruskal. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. CSLI Publications, 1983.
44. H. J. Shin, J. Lee, M. Gleicher, and S. Y. Shin. Computer puppetry: and importance-based approach. *ACM Transaction of Graphics*, 20(2):67–94, 2001.
45. 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.
46. J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.
47. Munetoshi Unuma, Ken Anjyo, and Ryoza Takeuchi. Fourier principles for emotion-based human figure animation. *Computer Graphics*, 29:91–96, 1995.
48. M. Alex O. Vasilescu. Proceedings of the international conference on pattern recognition. pages 456–460, 2002.
49. A. Witkin and M. Kass. Spacetime constraints. *Proceedings of ACM SIGGRAPH: Computer Graphics*, 22(4):159–168, August 1988.
50. Andrew P. Witkin and Zoran Popović. Motion warping. In *Proceedings of SIGGRAPH 95*, Computer Graphics Proceedings, Annual Conference Series, pages 105–108, August 1995.
51. Victor B. Zordan and Jessica K. Hodgins. Motion capture-driven simulations that hit and react. In *ACM SIGGRAPH Symposium on Computer Animation*, pages 89–96, July 2002.