

# Learning the Stylistic Similarity Between Human Motions

Yu-Ren Chien and Jing-Sin Liu

Institute of Information Science  
Academia Sinica, Taiwan

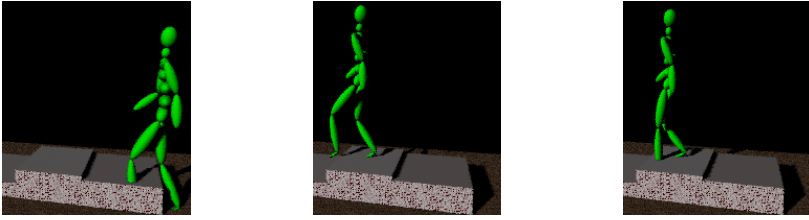
yrchien@ntu.edu.tw, liu@iis.sinica.edu.tw

**Abstract.** This paper presents a computational model of stylistic similarity between human motions that is statistically derived from a comprehensive collection of captured, stylistically similar motion pairs. In this model, a set of hyper-surfaces learned by single-class SVM and kernel PCA characterize the region occupied by stylistically similar motion pairs in the space of all possible pairs. The proposed model is further applied to a system for adapting an existing clip of human motion to a new environment, where stylistic distortion is avoided by enforcing stylistic similarity of the synthesized motion to the existing motion. The effectiveness of the system has been verified by 18 distinct adaptations, which produced walking, jumping, and running motions that exhibit the intended styles as well as the intended contact configurations.

## 1 Introduction

Human motions come in various types or styles, such as walking, limping, running, jumping, etc. Animators are always going to great pains to ensure that the produced character motions exhibit styles that match the exact styles they intend. Even if motions matching animators' stylistic intentions occasionally exist in motion capture databases, such canned motions usually cannot be directly reused due to the inability of these very motions to (geometrically) fit in the current scenes. What underlies an animator's effort to check a produced style against an intended or canned style, is a perceptual similarity between two motions, which we call the *stylistic similarity* between human motions. This work aims to automate this similarity judgment made by animators—we believe that such automation would greatly facilitate character animation. Specifically, automated stylistic similarity judgments can be applied to *human motion adaptation*, where a canned or captured motion with the desired style, which we call the *example motion*, is adapted to the geometry of the current scene (see Figure 1) with stylistic details preserved. We present a framework for human motion adaptation where stylistic similarity of the synthesized motion to the example motion is ensured by building the automated similarity judgment into the spacetime constraints formulation [1].

In this work, we regard the similarity judgment as a classification task, where the observer decides whether two motions are stylistically similar or not. Instead of gathering training motion pairs of both classes and adopting a two-class pattern recognition technique like SVM, we utilize a comprehensive collection of stylistically similar motion pairs made available by motion capture, and take an unsupervised learning approach



**Fig. 1.** A system for human motion adaptation adapts a user-supplied (example) motion to a new environment while preserving its stylistic details. **Left:** A typical example motion (walking on the flat ground, with the left foot raised off the ground in this snapshot), which is geometrically undesirable with respect to the intended environment shown here, in that the foot-contact configurations deviate from the targets on the marble steps so much that severe penetrations into the marble steps occur. **Middle:** The intended new motion for this particular scenario (example motion and intended environment), which has been adapted to the surface of the marble steps without being stylistically distorted. **Right:** A stylistically undesirable new motion for the same scenario, where the foot contacts have been adapted to the surface of the marble steps while the left foot penetrates into the marble steps due to the abnormally straight pose of the left leg.

to automating this similarity judgment, using single-class SVM (1-SVM) [2] and kernel PCA (KPCA) [3] to model the distribution of stylistically similar motion pairs. Moreover, in order to extract from two motions appropriate features for use with these statistical techniques, we propose a set of rules for segmentation of human motions and alignment of segmented motions, which rules form a structural paradigm for treating human motions that is novel in the world of character animation.

One concern about our model of stylistic similarity would be the lack of a formal definition of what we mean by “style,” which term always derives its definition from some subjective taxonomy implied by its context. Since the algorithms we adopt for building the model, i.e. 1-SVM and KPCA, generalize entirely from the training data, the “styles” considered in this work are actually defined by the exact labeling system used in the CMU Graphics Lab Motion Capture Database. In practice, we collected walking, running, and jumping data exactly as they are labeled in the database. We believe that the proposed technique for modeling stylistic similarity can be applied to any specific setting of stylistic taxonomy, or even any other perceptual (perhaps not visual) modalities, as long as appropriate data and features can be provided.

## 2 Related Work

Extraction of stylistically relevant features from human motion is essential to the effectiveness and efficiency of the following two tasks:

- **Style-based posture estimation:** In [4], each transition pose in jumps was represented by some centers of body-part mass while estimated. In [5], joint angles, vertical components in pelvis orientation, and their time-derivatives were extracted for estimation of the probability density of human posture in a particular style of motion; and

- **Human activity recognition:** In [6], joint positions were extracted for short-time recognition of motion styles. In [7], a fixed number of frames were randomly sampled from the 2-D motion; from each of these frames, global 2-D angular poses/velocities were extracted. In [8], feature extraction from an arm motion was performed by sampling the 2-D hand trajectory at every impulse of the hand acceleration signal.

There has been much research into human motion adaptation. Certain details in the example motion can be preserved in the new motion by minimizing a weighted sum of squared changes in motion parameters [9,10,11,12], or by iteratively modifying the (initially example) motion with the gradient method [13] or with a non-statistical variant of the Kalman filter [14]. In [15], [16], and [17], force patterns were extracted from the example motion and enforced in the new motion.

### 3 Segmentation and Alignment

In this work, we decompose each motion into a set of *motion segments*, rather than a sequence of postures, as the basic units of processing. The rationale behind this choice is that motion segments are more psychologically relevant than postures—we do not perceive and control our motions as individual postures, but as such basic movements as steps and twists [18]. Also inspired by the fact that we would at some times observe the motions of various body parts as a whole, and at other times focus on the motion of a particular limb, we break each motion both into *whole-body motion segments* and into *limb-specific motion segments*.

Segmentation of the motion of a specific limb is based on the interaction of its end-effector with the environment. There are two states to this interaction: the in-contact state and the free state. The motion is segmented at every state transition, e.g. at the heel-down frame and the toe-off frame of each foot contact, so that the resulting motion segments sequentially alternate between the two states.

Segmentation of a whole-body motion is achieved by *fusing limb-specific motion segmentations*, which in turn consists in grouping by concurrence all the limb-specific motion segments (originally grouped by limb). Let the sets  $A$  and  $B$  denote the frame numbers spanned by two limb-specific motion segments, and assume that  $|A| \geq |B|$  ( $|\cdot|$  denotes the set cardinality). The two segments are declared to be in the same group if and only if  $|A \cap B| > \frac{|B|}{2}$ , where  $\cap$  denotes the set intersection. We define a whole-body motion segment for each of the resulting (mutually exclusive) groups by assigning the *union* of the limb-specific frame spans to the whole-body frame span.

Consider two motions  $\mathcal{M}_1$  and  $\mathcal{M}_2$  of the same limb that share the same state sequence of length  $N$ , i.e. that start with the same state of interaction with the environment and both undergo exactly  $N - 1$  state transitions. To relate  $\mathcal{M}_1$  and  $\mathcal{M}_2$  in a structural manner, we align them by the state sequence, giving  $N$  pairs of aligned motion segments. In general for two alignable motions, the  $i$ -th (whole-body or limb-specific) segment in one motion can only be aligned with the corresponding ( $i$ -th) segment in the other. A pair of aligned motion segments extracted from a motion capture database generally have different durations.

## 4 A Framework for Human Motion Adaptation

### 4.1 Parameterization

The new motion to be optimized is represented by a vector made up of the following variables:

- 6 rigid DOFs per frame for pelvis motion; and
- 60 standardized *principal components* for each motion segment of each limb with which the character touches the environment. Any limb or torso that never touches the environment inherits its motion from the example motion without taking up any variables.

A concatenation of a 25-sample version of the angular signals represented by the 60 components can be computed by transforming the components according to a learned subspace [18].

For subspace learning, we have collected 98 motion clips of such types as climbing, jumping, running, walking, etc. By collecting limb-specific motion segments across all the clips, we have as training data 514 (507, resp.) segments for the left leg (right leg, resp.). After performing PCA on the training patterns extracted from the training segments [18], we found the first 60 of the 175 principal components for each (7-DOF) leg to account for about 99.9% of the total variation.

The low dimensionality of this motion representation is intended to avoid any prohibitive amount of computation time or failure of convergence in the optimization, which would generally be incurred by simply representing the motion by tens of postural variables per frame.

### 4.2 Constraints and Objectives

The synthesized motion  $\mathbf{x}^*$  minimizes the function

$$F(\mathbf{x}) = w_p \cdot P(\mathbf{x}) + w_c \cdot C(\mathbf{x}) + w_q \cdot Q(\mathbf{x})$$

subject to the constraints  $\mathbf{S}(\mathbf{x}) \geq \mathbf{0}$  and  $\mathbf{b}_l \leq \mathbf{V}(\mathbf{x}) \leq \mathbf{b}_u$ .

The objective function  $P(\cdot)$  represents the *stylistic similarity* of the new motion to the example motion, giving the sum-square value of  $N_l + N_w$  nonnegative distortion scores based on the KPCA, where  $N_l$  is the number of limb-specific motion segments represented by  $\mathbf{x}$ , and  $N_w$  is the number of whole-body motion segments in each of the new motion and the example motion. Each distortion score measures the stylistic distortion of a motion segment in the new motion from its aligned counterpart in the example motion, so that the score is close to zero if and only if the aligned pair is stylistically similar. In addition,  $C(\cdot)$  represents the sum-square distance of end-effectors from user-specified targets during contacts, and  $Q(\cdot)$  penalizes 1) excessive contact torques about centers of support [4,19], and 2) loss of balance [4] at frames specified by the user for balancing.

The constraint function  $\mathbf{S}(\cdot)$  also represents the *stylistic similarity* of the new motion to the example motion, giving a vector of  $N_l + N_w$  similarity scores based on the 1-SVM. Each similarity score measures how a pair of aligned motion segments (a segment in the

new motion and its aligned counterpart in the example motion) are stylistically similar to each other, so that the score is nonnegative if and only if the pair is stylistically similar. Moreover,  $\mathbf{V}(\cdot)$  is intended to realize joint limits, intersegmental continuity, smoothness of pelvis motion, and flight-phase dynamics.

Since  $F(\cdot)$  has a sum-square structure, this particular nonlinear program can be treated as a constrained nonlinear least squares problem.

## 5 Stylistic Similarity

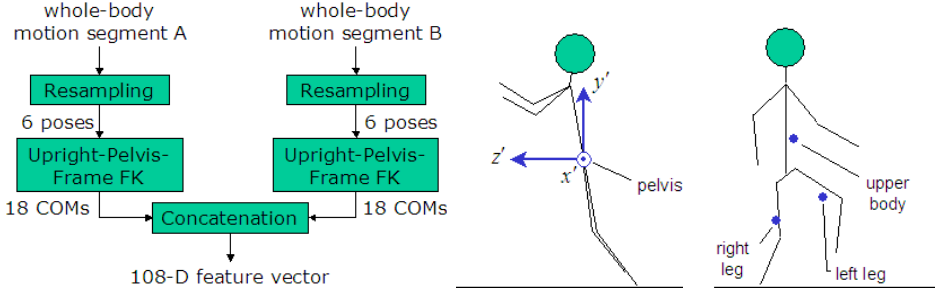
In this section, we develop algorithms for constructing from motion capture data computational models of the stylistic similarity between motion segments. These algorithms are based on the abstraction of a space of all possible motion segment pairs, and the characterization of a dichotomy in that space that represents the difference between stylistic similarity and stylistic distortion. To be specific, we take advantage of the abundance of instances of stylistic similarity in a motion capture database to characterize the region occupied by all possible stylistically similar pairs, which we call the *similarity region*. Our intended models of stylistic similarity will be given by approximations of the similarity region.

To approximate the geometry of a region in the 3-D space, one might sample the region and simply fit a surface to the boundary samples; however, if the region is part of any surface, it would be more effective to also fit one or two surfaces to all the samples. In view of this, we apply two approximation techniques to the similarity region: the 1-SVM [2] and the KPCA [3], which respectively solve the above two classes of approximation problems, giving a *supporting hypersurface* and a set of *fitted hypersurfaces* for the sampled similarity region.

### 5.1 Feature Extraction

In this section, we describe the actual features extracted to represent each point in the space of all possible motion segment pairs. Specifically, we consider multiple spaces of motion segment pairs: one for pairs formed by whole-body motion segments, and the others for pairs formed by limb-specific motion segments (one space per limb). The features extracted for each space will underlie the approximation of the corresponding similarity region.

**Whole-Body Features.** The procedure defined in this section takes a pair of whole-body motion segments (A, B) as input, and gives as output a feature vector that encodes the Cartesian postural evolutions of major body parts in A and B. As shown in Figure 2, the extraction process starts by linearly interpolating each of A and B at  $K$  uniformly spaced instants to give  $K$  postures (totally  $2K$  postures for A and B). For each such posture, consider a reference frame that has origin at pelvis, y-axis upward, and z-axis anterior (see Figure 2), which serves to remove all the global information, except the balance information, from the features for the posture. The process then computes (using the mass distribution data reported in [20]) centers of upper-body, left-leg, and right-leg mass (see Figure 2) with respect to the reference frame for each of the  $2K$  postures. Concatenating the COM position vectors for all the three body parts and for



**Fig. 2.** Whole-body feature extraction. **Left:** The procedures. (FK stands for forward kinematics.) **Middle:** The reference frame. The character exactly faces the left. **Right:** The COMs computed.

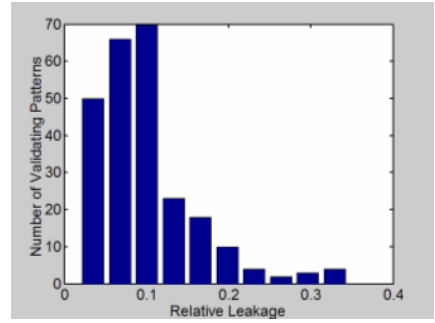
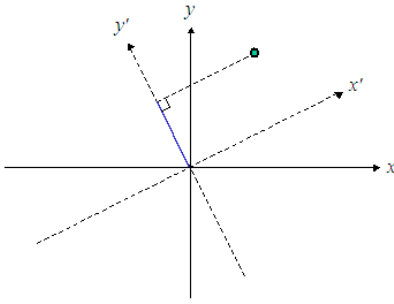
all the postures, the process finally produces an  $18K$ -dimensional vector as the feature vector. We set  $K$  to 6 in all experiments.

**Limb-Specific Features.** The feature vector for a pair of motion segments (A,B) of a limb is 30-dimensional, composed of the first 15 standardized principal components for each of A and B. The 15 components are extracted from each segment by the analysis procedures given in Section 4.1, except with a further truncated subspace.

## 5.2 Building the Models

To sample the similarity region in the space defined by each extracted set of features, i.e. to gather training patterns, we have collected about 70 pairs of captured motion clips, each pair formed by two motions *of the same type* (walking, jumping, or running) and with the same sequence of end-effectors in contact with the environment (e.g. two walks both starting with a left-foot contact and lasting for five steps, or two jumps both composed of a doubly supported takeoff, a flight stage, and a doubly supported landing, so that their segmentations can be aligned). To extract three sets of training patterns from the clips (one dataset for whole-body features, one for left-leg features, and the other for right-leg features; in each dataset we have approximately 90 patterns from walking, 70 from jumping, and 90 from running), we applied the whole-body feature extraction procedure described in Section 5.1 to each pair of aligned whole-body motion segments found in each pair of clips, and the limb-specific procedure in Section 5.1 to each pair of aligned limb-specific (leg) motion segments found in each pair of clips.

**Models Based on the 1-SVM.** As the first part of our modeling efforts, for each dataset  $\{\mathbf{p}_i \in \mathbb{R}^n : i = 1, \dots, I\}$ , we fit a hypersurface  $\{\mathbf{p} \in \mathbb{R}^n : f_s(\mathbf{p}) = 0\}$  to the boundary samples in the dataset such that  $f_s(\mathbf{p}_i) \geq 0, \forall i \in \{1, \dots, I\}$ , i.e. all the training patterns lie on the same side of the *supporting hypersurface*, using the *single-class SVM* training routine in the OSU Support Vector Machines Toolbox, which is based on the LIBSVM library. 1-SVM [2] and the better-known two-class SVM differs in that instead of finding a separating hypersurface between the respective supports of two groups of training patterns, the 1-SVM algorithm simply estimates the support of a single group of training patterns, i.e. its purpose is to detect novelty given a set of usual patterns.



**Fig. 3. Left:** Illustrating the feature-space distance of a testing pattern from the subspace spanned by the lower-order principal axes. The  $x$ - $y$  plane represents the  $\infty$ -D feature space. The  $x'$ - and  $y'$ -coordinates represent the lower-order and the higher-order principal components, respectively. The dot represents the testing pattern. The blue line segment on the  $y'$ -axis represents the distance. **Right:** Cross-validation result for the hypersurfaces fitted to the whole-body dataset.

In the theory of 1-SVM, the supporting hypersurface can be represented by a supporting hyperplane in an infinite-dimensional space related to the original space by a radial basis function kernel,  $f_k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \cdot \|\mathbf{x} - \mathbf{y}\|^2)$ . Here we adopt this kernel function with the parameter  $\gamma$  tuned to give the best cross-validation accuracy possible. To estimate the modeling accuracy of each supporting hypersurface learned by the 1-SVM algorithm, we have conducted 20-fold cross-validation on the corresponding dataset. The probability of leakage of sample  $\mathbf{p} \in R^n$  beyond the whole-body (left-leg, right-leg, resp.) supporting hypersurface,  $\text{Prob}[f_s(\mathbf{p}) < 0]$ , is estimated to be 0.048 (0.0332, 0.0413, resp.) with the 95% confidence interval calculated to be  $[0.025, 0.0823]$  ( $[0.0144, 0.0644]$ ,  $[0.02, 0.0747]$ , resp.).

**Models Based on the KPCA.** As the second part of our modeling efforts, we fit a set of hypersurfaces to the samples in each dataset by means of the KPCA algorithm [3]. In the theory of KPCA, one can perform PCA on the training patterns in the same  $\infty$ -D feature space as defined for 1-SVM modeling above. For each resulting principal component that has small variance, we have a corresponding hyperplane that is fitted to the training patterns in the  $\infty$ -D space:  $\{\mathbf{p} \in R^\infty : \mathbf{e} \cdot (\mathbf{p} - \boldsymbol{\mu}) = 0\}$ , which is perpendicular to the corresponding principal axis  $\mathbf{e} \in R^\infty$ , passes through the mean  $\boldsymbol{\mu} \in R^\infty$ , and defines a corresponding input-space *fitted hypersurface*. One may find principal components to have small variances starting from order  $O_c$ . For all these small components, which we call *higher-order principal components*, we have a corresponding set of input-space hypersurfaces that are all fitted to the training patterns, so that each training pattern is known to lie on the intersection of the set of hypersurfaces. Consider a testing pattern represented in the  $\infty$ -D space by the principal components  $\mathbf{z} = (z_1, z_2, \dots)$ . We measure how close the pattern is to the intersection of input-space fitted hypersurfaces by the proportion of its feature-space squared norm contributed by the higher-order principal components ( $\|\mathbf{z}\|^2 - \sum_{i=1}^{O_c-1} z_i^2$ )/ $\|\mathbf{z}\|^2$ , which we call the *relative leakage* of the pattern with respect to the fitted hypersurfaces. As depicted in Figure 3, the numerator can be interpreted as the squared distance of the pattern from the subspace spanned by the

lower-order principal axes. Note that as an inner product in the centered  $\infty$ -D space,  $\|\mathbf{z}\|^2$  can be computed via  $f_k(\cdot, \cdot)$  and centering [3]. In our experiments, the cut-off order  $O_c$  is set to 16 for the whole-body dataset and to 22 for each single-leg dataset.

To estimate the modeling accuracy of each set of fitted hypersurfaces learned by the KPCA algorithm, we have conducted 20-fold cross-validation on the corresponding dataset. For the hypersurfaces fitted to the whole-body dataset, we plot the histogram of the relative leakage over all validating patterns in Figure 3; for the hypersurfaces fitted to the single-leg datasets, the same histograms are again computed, as not shown here due to the space limit. For every set of fitted hypersurfaces, the average relative leakage over all the validating patterns is found to be roughly 0.1, and about 90% of the patterns are found to have relative leakages below 0.2.

## 6 Adaptation Experiments

In our implementation of the adaptation system presented in Section 4, for better efficiency of the optimization, we apply the transformation proposed by Schittkowski [21] to the constrained nonlinear least squares problem defined in Section 4.2. The resulting nonlinear program is coded in ANSI Fortran 77, augmented with sparse derivative computation by the ADIFOR 2.0 automatic differentiation tool, and solved by the sparse nonlinear programming routine in the NAG Fortran Library, which is based on the SNOPT package described in [22], on a Pentium-4 1.8-GHz PC running Red Hat (7.2) Linux with 512-MB RAM. Note that low-dimensional representation of human motion serves here to prevent variables from significantly outnumbering constraints, thereby favoring the efficiency of SNOPT and ADIFOR-generated codes. In our experiments, every adaptation took less than thirty minutes.

To evaluate the effectiveness of our adaptation system, we have tested it on 18 distinct motion adaptation tasks. In these tasks, the example motions are of the types walking, jumping, and running (6 tasks for each type) and apart from those used in preparation of the datasets from which the models of stylistic similarity were learned, and most of them are each composed of 3 or 4 whole-body motion segments. We specified contact targets such that each example motion significantly deviates from the associated targets in the new environment, and the targets still look attainable for the specific activity in the example motion, thereby controlling the task difficulty. By playing back and watching each of the 18 synthesized motions, we found that each motion exhibits the intended style, as well as a realistic contact during each contact event. All these tasks were performed with the same parameter setting, except that in running tasks only 30 standardized principal components were used to represent each leg motion segment in order for the number of variables to roughly match the relatively small number of constraints in running tasks. Only ending frames in jumps were marked for balancing.

To verify the necessity of enforcing stylistic similarity to the example motion, we created a crippled version of our system by dropping from the nonlinear program the functions representing the similarity, i.e.  $\mathbf{S}(\cdot)$  and  $P(\cdot)$ , and repeated all the above 18 tasks using this crippled version instead. The new motions resulting from this verification were each found to be stylistically distorted. Manifestations of such distortion include abnormal poses, abnormal variations in speed of movement, and abnormal directions of



movement. On the other hand, robustness in attaining geometrical goals was also impaired—two of these motions were found to be geometrically highly undesirable. One interesting implication of this verification is that collisions and kinematical singularities are usually accompanied by stylistic distortion and can be adequately avoided by enforcing stylistic similarity to the example motion, which is obvious when we notice that abnormal poses brought about penetrations in several tasks, and that local minimums of the geometrical sum-square error are a specific form of kinematical singularities.

## 7 Discussion

We have presented a motion adaptation system that is sensitive to stylistic fidelity in its quest for the optimal solution. In this final part of the paper, we discuss the uniqueness of this system by providing some in-depth comparisons with the system proposed in [12]:

- Both muscle forces and contact forces are bound-constrained in [12]. We also constrain contact forces in our system. However, without computing muscle forces, we let the new motion imitate the example motion according to the data-driven model of stylistic similarity, which we believe has the effect of capturing not only the bounds on, but also the patterns of, muscle forces. Such pattern-following mechanism is also present in [12] in the form of minimum sum-square difference between Cartesian representations of the example motion and the new motion.
- Both systems generate new motions by solving spacetime optimization with the same sparse nonlinear programming solver, and, for better convergence of the optimization, simplify the underlying postural parameterization according to the specific behavioral characteristics in the example motion. We also build into our parameterization a data-driven set of constraints that account for the inherent spacetime characteristics of all human motions, thereby further lowering the dimensionality.
- Both systems are shown in extensive experiments to accurately adapt jumping/running examples to new environments while preserving stylistic quality. We additionally present style-preserving adaptations for walking motions.

## Acknowledgments

The data used in this project was obtained from `mocap.cs.cmu.edu`. The database was created with funding from NSF EIA-0196217. The ADIFOR 2.0 System was obtained from Rice University and the University of Chicago as Operator of Argonne National Laboratory. This work has been supported by NSC 93-2212-E-001-001, NSC 92-2212-E-001-001, and NSC 91-2212-E-001-001.

## References

1. Witkin, A., Kass, M.: Spacetime constraints. In: SIGGRAPH. (1988) 159–168
2. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. *Neural Computation* **13** (2001) 1443–1471

3. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* **10** (1998) 1299–1319
4. Liu, C.K., Popović, Z.: Synthesis of complex dynamic character motion from simple animations. In: *SIGGRAPH*. (2002) 408–416
5. Grochow, K., Martin, S.L., Hertzmann, A., Popović, Z.: Style-based inverse kinematics. *ACM Trans. Graph.* **23** (2004) 522–531
6. Arikian, O., Forsyth, D.A., O'Brien, J.F.: Motion synthesis from annotations. *ACM Transactions on Graphics (TOG)* **22** (2003) 402–408
7. Ben-Arie, J., Wang, Z., Pandit, P., Rajaram, S.: Human activity recognition using multidimensional indexing. *IEEE Trans. Pat. Anal. Mach. Intel.* **24** (2002) 1091–1104
8. Rao, C., Yilmaz, A., Shah, M.: View-invariant representation and recognition of actions. *International Journal of Computer Vision* **50** (2002) 203–226
9. Gleicher, M.: Motion editing with spacetime constraints. In: *Proceedings of the 1997 symposium on Interactive 3D graphics*. (1997) 139–ff.
10. Gleicher, M.: Retargetting motion to new characters. In: *SIGGRAPH*. (1998) 33–42
11. Lee, J., Shin, S.Y.: A hierarchical approach to interactive motion editing for human-like figures. In: *SIGGRAPH*. (1999) 39–48
12. Popović, Z., Witkin, A.: Physically based motion transformation. In: *SIGGRAPH*. (1999)
13. Sulejmanpašić, A., Popović, J.: Adaptation of performed ballistic motion. *ACM Trans. Graph.* **24** (2005) 165–179
14. Tak, S., Ko, H.S.: A physically-based motion retargeting filter. *ACM Trans. Graph.* **24** (2005)
15. Pollard, N.S., Behmaram-Mosavat, F.: Force-based motion editing for locomotion tasks. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. (2000)
16. Pollard, N.S.: Simple machines for scaling human motion. In: *Computer Animation and Simulation '99*. (1999)
17. Liu, C.K., Hertzmann, A., Popović, Z.: Learning physics-based motion style with nonlinear inverse optimization. *ACM Trans. Graph.* **24** (2005) 1071–1081
18. Fod, A., Matarić, M.J., Jenkins, O.C.: Automated derivation of primitives for movement classification. *Autonomous Robots* **12** (2002) 39–54
19. Fang, A.C., Pollard, N.S.: Efficient synthesis of physically valid human motion. *ACM Transactions on Graphics (TOG)* **22** (2003) 417–426
20. de Leva, P.: Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters. *J. Biomechanics* **29** (1996) 1223–1230
21. Schittkowski, K.: Solving constrained nonlinear least squares problems by a general purpose SQP-method. In Hoffmann, K.H., et al., eds.: *Trends in Math. Optim.* (1988) 295–309
22. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP algorithm for large-scale constrained optimization. Technical Report NA-97-2, Dept. Mathematics, UCSD (1997)