Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2011)
A. Bargteil and M. van de Panne (Editors)

A Puppet Interface for Retrieval of Motion Capture Data

Naoki Numaguchi[1] Atsushi Nakazawa[1,2] Takaaki Shiratori[3] Jessica K. Hodgins[4,3]

[1] Osaka University [2] PRESTO, JST [3] Disney Research, Pittsburgh [4] Carnegie Mellon University

Abstract

Intuitive and efficient retrieval of motion capture data is essential for effective use of motion capture databases. In this paper, we describe a system that allows the user to retrieve a particular sequence by performing an approximation of the motion with an instrumented puppet. This interface is intuitive because both adults and children have experience playacting with puppets and toys to express particular behaviors or to tell stories with style and emotion. The puppet has 17 degrees of freedom and can therefore represent a variety of motions. We develop a novel similarity metric between puppet and human motion by computing the reconstruction errors of the puppet motion in the latent space of the human motion and those of the human motion in the latent space of the puppet motion. This metric works even for relatively large databases. We conducted a user study of the system and subjects could find the desired motion with reasonable accuracy from a database consisting of everyday, exercise, and acrobatic behaviors.

Categories and Subject Descriptors (according to ACM CCS): I.3.1 [Computer Graphics]: Hardware Architecture—Input devices I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Publicly available libraries of motion capture data [CMU] and free or inexpensive software for editing [Ani, Smi] allow even novices to create animations and share their creations on the internet. To find an appropriate sequence from these databases, users must inspect each motion clip in the database based on associated tags or thumbnail videos. The same is true for professional animators or game developers. The ability to easily re-use data for a sports video game, for example, could result in significant savings in both capture and cleanup of the data. However, these queries are difficult to properly specify, particularly for large databases.

In this paper, we present an instrumented puppet that functions as an intuitive interface for retrieving motion capture data from a database. The human-like puppet is instrumented with one orientation sensor and 14 potentiometers to measure the orientation of the body and joint angles. The puppet has 17 degrees of freedom (DOFs). The query to the motion retrieval system is sensor readings as the user manipulates the puppet. The sensor data are retargeted to a human skeleton and matched to each behavior primitive in the database. Finally, a few behaviors with the highest matching scores are shown so that the user can select the best match.

This interface is intuitive even for novices, because most people grew up playacting with puppets, dolls and stuffed animals. Children often bounce a stuffed animal around to indicate running or crash them into each other to portray fighting. The motions can be quite expressive showing both style of the behavior and the character's current emotion. Complicated motions are a bit more difficult because the user has only two hands with which to manipulate the puppet. We address this problem by allowing the user to change the tightness (friction) of each joint and by giving them two ways to hold the puppet, either grasping it in their hands or fixing it to a stage. The user's performance of the behavior does not need to be perfect, and it is sufficient if it merely distinguishes the desired class of motions from other similar motions in the database. For example, a punch must look different from a baseball pitch.

Another benefit of the puppet interface is the ability to easily represent acrobatic or superhero motion. Motion capture systems require the users to actually perform an acrobatic motion in the real world. That is, the input query motion is limited by the skills of the user. In contrast, the puppet interface enables the users to easily represent humanly impossible motion, and could be used to search a database of keyframed motion.

The major challenge in developing a retrieval system with the puppet interface is matching a given motion of the puppet to the appropriate sequence in the motion capture database. The significant differences between the motion dynamics of the puppet and a human actor make this problem intractable for any kind of direct matching algorithm based on joint positions, angles or other sensor values. Matching is further complicated by the variability in performance between users. We describe a new retrieval algorithm called the *dual subspace projection method*. This algorithm calculates the distance between the human and input puppet motion by combining the reconstruction errors of the puppet motion in the human motion latent space and the human motion in the puppet motion latent space. This technique is robust to individual styles of puppet manipulation, and fast enough for interactive motion design.

This paper is organized as follows. We first cover prior work in Section 2. Then Section 3 describes the design of the puppet hardware, and Section 4 describes motion retrieval algorithms. The experimental results are shown in Section 5. We conclude this paper by discussing the limitations and possible future directions in Section 6.

2. Prior Work

Retrieval from motion databases has been explored using many different interfaces including a mouse or tablet, the user's body, and instrumented puppets. Various similarity metrics have also been used to retrieve the motion. We now briefly describe these two bodies of related work.

2.1. Interface Devices

One of the most common interface devices is the mouse or pen tablet, which allow the user to draw 2D trajectories and poses. Thorne and colleagues [TBvdP04] developed such an interactive system. Given a pre-determined mapping between pen strokes and motion patterns, combinations of strokes allowed for character navigation. Li and colleagues [LGY*06] built a pen-based retrieval system in which viewing directions for 2D drawings were estimated to calculate errors of pose and trajectories between 3D motion capture data and the 2D drawings.

Because of the ambiguity between 2D and 3D, such systems often require domain knowledge. One solution is to use a *motion graph* [KGP02, LCR*02, AF02], where pairs of frames of motion data are connected when a smooth and natural transition between them is available. Retrieval of appropriate paths in the motion graph allows the synthesis of natural animation of characters. Lee and colleagues [LCR*02] demonstrated that the users could control characters by sketching a path with the mouse or selecting among a set of proposed paths. Later, the retrieval algorithms were improved with annotations [AFO03] or user models [MP07]. These approaches have been used primarily for realtime interactive systems such as video games, while our goal is to allow the user to design animations.

Capturing the whole-body motion of the user also allows the specification of a search query for a motion database. Chai and Hodgins [CH05] used a small marker set and a stereo-vision system to obtain approximate human poses and synthesized a matching motion by combining similar poses from a database. They introduced a fast nearest neighbor search method for realtime retrieval. Ren and colleagues [RSH*05] built a three camera system that extracted silhouettes of the user to find a matching sequence of motion in a motion graph. They utilized AdaBoost to build a metric for composing silhouettes. Multiple accelerometers on a shirt were used to select motion capture data from a database [SH08]. Yin and colleagues [YP03] used a motion capture database and a measure of foot pressure to estimate the user's pose.

A few researchers have used an instrumented puppet to retrieve motion as we do. Johnson and colleagues [JWB*99] developed an instrumented puppet to control a bird-like character. The user's manipulation of the puppet was recognized with HMMs to select among a set of pre-determined motion patterns. However, HMMs require a large amount of data in the learning step. Esposito and colleagues [EPO95] created an instrumented puppet called the *Monkey* that allowed the users to specify human or character poses intuitively. A similar instrumented puppet was also developed by Knep and colleagues [KHSW95] to specify poses of dinosaurs for movie production. Yoshizaki and colleagues [YSC*11] developed an active puppet interface that has an actuator for each joint. Actuating joints allows gravity compensation and biomechanical constraints, which improves the usability. Feng and colleagues [FGDJ08] used an instrumented puppet to retrieve human motion from a database. They attached color markers to puppet joints and estimated the positions with a stereo rig. Unlike these systems, our interface enables the users to specify poses while the they dynamically manipulate the puppet. Therefore, the specified poses can be more natural.

2.2. Similarity Metrics

A good similarity metric is essential for motion retrieval. Bruderlin and Williams [BW95] demonstrated that dynamic time warping (DTW) can be used to evaluate the similarity between two motion capture sequences. Later, Kovar and Gleicher [KG04] extended DTW to retrieve similar motions for interpolation. Müller and colleagues [MRC05, MR06] developed a content-based method for motion retrieval, where human motion is abstracted with geometrical relationships of body part pairs. Ishigaki and colleagues [IWZL09] built a performance interface that translated the user's intentions to human motion by applying a subspace method with principal component analysis (PCA) to a motion cap-

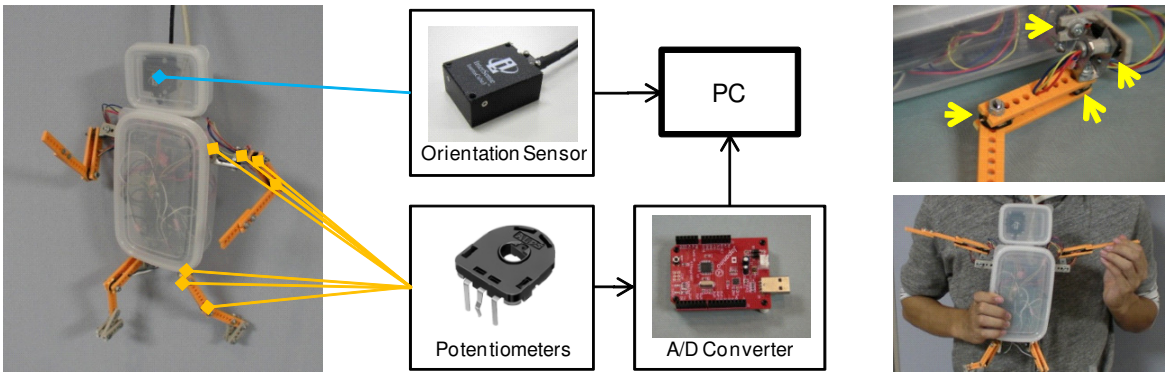


Figure 1: *Hardware setup of our retrieval system. Left: Structure of the puppet interface. The puppet has 17 DOFs consisting of 14 rotary potentiometers and one orientation sensor. The potentiometers are attached to each joint and their resistance values are read by Arduino microcontroller. The orientation sensor (InertiaCube3 by InterSense Inc.) is installed in the head (fixed with the torso) and is directly read by a PC. The sensor readings are updated at 60 Hz. Right-top: Joint structure of left shoulder and elbow. Each arrow indicates a potentiometer. Right-bottom: A standard manipulation style of our interface.*

ture database. Deng and colleagues [DGL09] used probabilistic PCA to segment and learn motion capture data for retrieval. Ho and Komura [HK09] retrieved interactions between characters based on topological information derived from knot theory. In addition to the subspace methods, Yamamoto and colleagues [YFM98] developed a mutual subspace method (MSM) to compare time series of images. They used the canonical angles of subspaces as a similarity metric. Though this method has not been applied to human motion, the results for face identification showed the effectiveness of the algorithm.

3. Puppet Hardware

Figure 1 shows the hardware configuration of our puppet interface. We designed the puppet so that the user can easily use his or her hands to manipulate the limbs of the puppet. The size is 33 cm tall, 44 cm wide with arms outstretched, and the weight is 400 g.

Sensors: The puppet has 17 DOFs; 14 rotary potentiometers (three for each shoulder, one for each elbow, two for each hip and one for each knee) and an orientation sensor at the torso/head. The potentiometers measure joint angles, and the sensor readings are converted into digital signals by an Arduino microcontroller and sent to a PC at 60 Hz. The orientation sensor we used is InertiaCube3 by InterSense Inc., and consists of an accelerometer, gyroscope and magnetometer to measure orientation with respect to the Earth's magnetic field.

Joints: The right top of Figure 1 shows the structure of the joints. The potentiometer is surrounded by neighboring bones and fixed by a bolt. The user can tighten or loosen the joints with the bolt so that the stiffness can be adjusted to achieve complicated poses. For example, the user may want to hold a pose with the legs while freely manipulating the arms.

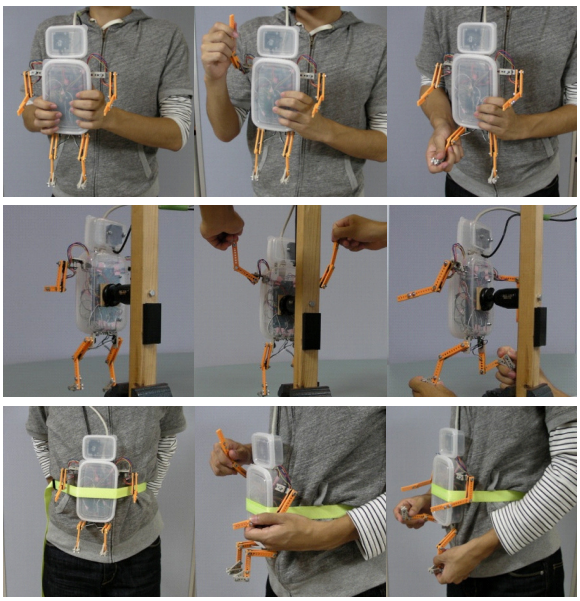


Figure 2: *Three control styles. Top: Hold in the hand, middle: attach to a stage, and bottom: attach to the user's body with a strap.*

Calibration: The calibration process of the puppet interface is simple. First, the users specify the "T-pose," where the arms are outstretched sideways, and the legs are vertical. The sensor readings from the T-pose correspond to 0°. Then, each joint is rotated by 90°. Scaling parameters ob-

tained from these two sensor readings provide the required calibration data.

Control styles: The design of the puppet allows a few different kinds of manipulation:

1. Use one hand to hold the puppet (Figure 2 top): The user holds the puppet with one hand and controls a limb with the other hand. With this approach, the orientation of the puppet and one body limb can be controlled.
2. Attach to a stage (Figure 2 middle): The user can control limbs with each hand. Though rotational motion cannot be expressed, multiple limbs can be controlled.
3. Attach to user's body with a strap (Figure 2 bottom): Some users prefer to tie the puppet to their body with a strap and control the limbs with their hands and the orientation by rotating their body. This approach keeps the orientation of the puppet aligned with that of the user and perhaps makes the limb control more intuitive as a result. This manipulation is similar to the system developed by Mazalek and colleagues [MCN*11].

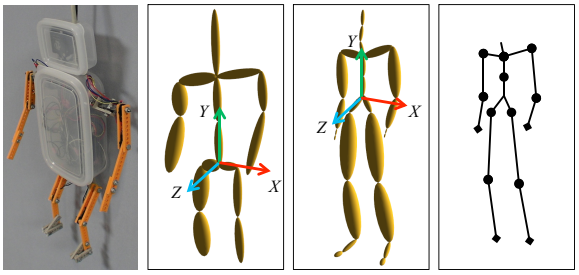
4. Motion Retrieval Algorithm

To understand how people manipulate the puppet, we conducted an informal subject test. First, we showed a video created from motion capture data to the subjects, and then asked the subjects to express the motion with the puppet. The subjects commented that manipulating the puppet was an intuitive way to express the motion. We obtained the following key findings: 1) the subjects represented the motion with positions of the end-effectors and/or joints rather than joint angles, 2) joint trajectories differed significantly between the puppet and human motion because of differences in the range of motion, and 3) puppet motion did not obey physical laws because the puppet was supported by the user or a stage. These findings suggest that conventional approaches to compute a similarity metric between human motion with similar skeletons may not work.

Based on these findings, we designed the algorithm to robustly compute a similarity between motion with different skeletons and range of motion:

1. To handle differences between the puppet and human skeletons, puppet motion is retargeted to the dimensions of a human skeleton calculated from motion capture data.
2. The algorithm uses joint positions in a body centered coordinate system.
3. To compensate for the different range of motion between the puppet and human, we introduced the dual subspace projection method (DSPM). The method evaluates similarities based on the pose distribution in their subspaces. We also consider an anomaly detection framework for robustness.

Because our method is based on a subspace method, we can choose any latent space representation for DSPM. We tested PCA and Gaussian process latent variable models (GP-LVM). In the following section, we explain a conventional subspace method which is the basis of our algorithm, provide details of each component mentioned above and describe how to incorporate PCA and GP-LVM into DSPM.



(a) (b) (c) (d)

Figure 3: *Coordinate systems and joint positions used to obtain feature vectors. (a) Joint angles of the puppet are read from the potentiometers. (b) A reconstructed posture using the puppet skeleton. (c) A reconstructed posture using the human skeleton. The arrows indicate axes of a body centered coordinate system. (d) Black dots indicate joint positions considered for a feature vector.*

4.1. Data Preprocessing

Feature vectors: Given joint angles and body orientation from the sensors of the puppet, the system directly maps the angles to a human skeleton calculated from motion capture data. Positional or acceleration data is not used, and the root remains at the origin of the world coordinate system. Then, a pose of the human skeleton is calculated from the joint angles of the puppet using forward kinematics. The joint positions considered are neck, shoulders, elbows, hands, hips, knees and feet (14 in total). These positions are converted into a body centered coordinate system, where the Y axis points upward in the world coordinate system, and the X axis points sideways, and the Z axis points the facing direction of the root (Figure 3). Fourteen joint positions for each frame are stored in a single feature vector.

Behaviors in a database: Because our system is designed to retrieve short human behaviors, we segment long motion capture sequences and store them in a database. We use a probabilistic PCA-based algorithm that can segment motion capture data into distinct behaviors [BSP*04].

4.2. Subspace Method

A subspace method (SM) computes a reconstruction error of the input data in a latent space of target data. This method is a fundamental algorithm for pattern recognition and has been used for motion retrieval. For example, Ishigaki and colleagues applied SM for recognizing human be-

haviors by matching the user motion data and a motion database [IWZL09]. Here, we briefly explain SM using PCA and how to incorporate anomaly detection.

Let $\mathcal{Q}$ be puppet motion consisting of N feature vectors $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$, and $\mathcal{H}$ be human motion, respectively. Applying PCA to the human motion provides eigenvectors $\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_M$, where M is the number of dimensions. Defining a latent space matrix $\mathbf{H} = [\,\mathbf{h}_1\,\mathbf{h}_2 \cdots \mathbf{h}_{M'}]$ with dimension M', the reconstruction error Err for an input feature vector $\mathbf{x}_n$ is calculated as

$$Err(\mathbf{H}, \mathbf{x}_n) = \left|\left| \mathbf{x}_n - \left(\mathbf{H}\mathbf{H}^{\mathsf{T}} \left(\mathbf{x}_n - \bar{\mathbf{x}}^{\mathcal{H}} \right) + \bar{\mathbf{x}}^{\mathcal{H}} \right) \right|\right|, \quad (1)$$

where $\bar{\mathbf{x}}^{\mathcal{H}}$ is the mean feature vector of the human motion $\mathcal{H}$.

To robustly evaluate a distance D_{SM} between $\mathcal{Q}$ and $\mathcal{H}$, we employ a subspace anomaly detection approach [FYM05]. In this framework, sample datasets are projected onto the subspace of other datasets, and then are reconstructed in the original space. The maximum reconstruction error is used as the distance of the two datasets, namely,

$$D_{SM}(\mathcal{Q}, \mathcal{H}) = \max\left(Err(\mathbf{H}, \mathbf{x}_1), \cdots, Err(\mathbf{H}, \mathbf{x}_N)\right). \quad (2)$$

$D_{SM}(\mathcal{Q}, \mathcal{H})$ indicates larger error, The human motion that produces the minimum distance to the query puppet motion is selected as the best retrieval result.

4.3. Dual Subspace Projection Method

SM computes reconstruction errors by projecting the feature vectors of the query puppet motion onto the latent space of human motion, i.e. forward reconstruction errors. Therefore, the reconstruction errors become small in the case where the entire query puppet motion is similar to only part of the human motion but different from the rest. In our informal subject test, this phenomenon occurred frequently due to the difference in range of motion between puppet and human.

The dual subspace projection method evaluates backward reconstruction errors as well as forward reconstruction errors. Backward reconstruction errors are computed by projecting human motion onto the latent space of query puppet motion (Figure 4). In the case of the issue with SM, the backward reconstruction errors will be high, when only part of the puppet motion is similar to the human motion.

Given query puppet motion $\mathcal{Q}$ and human motion $\mathcal{H}$, PCA provides the latent spaces $\mathbf{Q}$ and $\mathbf{H}$ for those motions, respectively. DSPM computes the distance, D_{DSPM}, as

$$D_{DSPM}(\mathcal{Q}, \mathcal{H}) = \max(Err(\mathbf{H}, \mathbf{x}_1), \cdots, Err(\mathbf{H}, \mathbf{x}_N), Err(\mathbf{Q}, \mathbf{x}_1^{\mathcal{H}}), \cdots, Err(\mathbf{Q}, \mathbf{x}_{N^{\mathcal{H}}}^{\mathcal{H}})), \quad (3)$$

where $\mathbf{x}^{\mathcal{H}}$ is a feature vector of $\mathcal{H}$, and $N^{\mathcal{H}}$ is the number of frames of the human motion. To reduce the computational cost for interactive retrieval, we randomly choose K frames from both the query puppet motion and human motions. Human motion that has a minimum distance with the query puppet motion is retrieved as the best matching result.

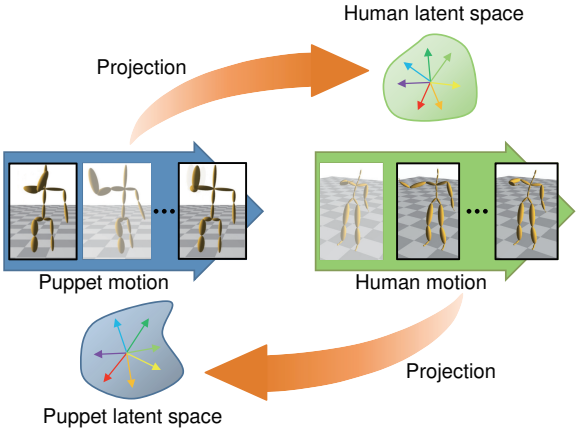


Figure 4: *The dual subspace projection method considers both forward reconstruction error by projecting the query puppet motion onto a latent space of each human motion and backward reconstruction error by projecting human motion onto a latent space of the query puppet motion.*

4.4. Gaussian Process Latent Variable Models

We can also use Gaussian process latent variable models [Law04] as a latent space representation for SM and DSPM. GP-LVM is a non-linear latent space that maps input feature vectors to a latent space by combining probabilistic PCA and a kernel method. Because of the performance of high dimensional data modeling, it is widely used for learning human motions and has been applied to many applications such as stylized inverse kinematics [GMHP04], and transferring human motion to non-humanoid characters [YAH10].

In the learning step, a model parameter set Φ and latent coordinates $\mathbf{L}$ for a given kernel matrix $\mathbf{K}$ which defines the GP-LVM latent space are estimated from feature vectors. Given an observation matrix $\mathbf{X}$ consisting of N^X feature vectors $\mathbf{x}_1, \cdots, \mathbf{x}_{N^X}$, the parameters are optimized by maximizing the probability of generating $\mathbf{X}$ as

$$p(\mathbf{X}|\mathbf{L}, \Phi) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{X}_d; \mathbf{0}, \mathbf{K}), \quad (4)$$

where $\mathbf{X}_d$ is d-th dimension of $\mathbf{X}$ and $\mathcal{N}$ is a Gaussian distribution.

In the retrieval step, when the feature vector of query motion $\mathbf{x}_q$ is given, the probability in the learned latent space can be used as an error metric. With respect to each latent coordinate $\mathbf{l} \in \mathbf{L}$ of the learned motion $\mathcal{M}$, the probability is

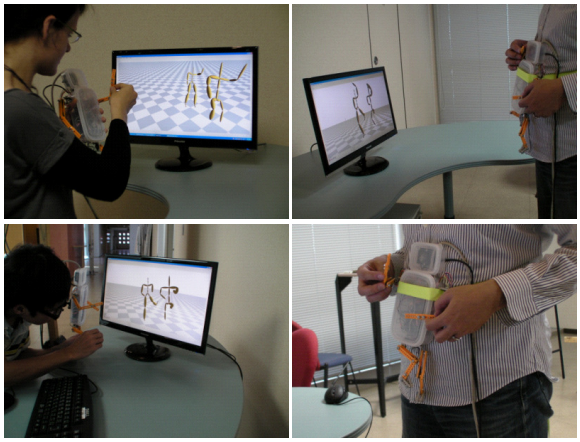


Figure 5: *Experiments with the puppet interface.*

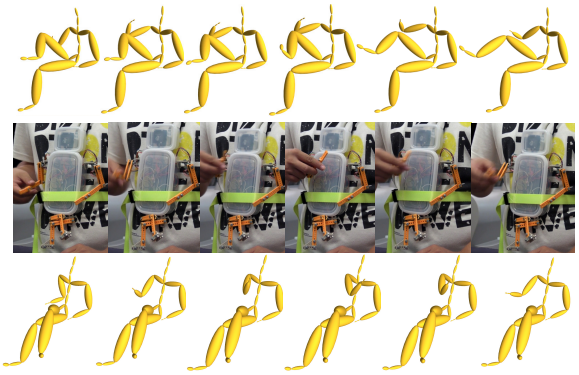


Figure 6: *Target human motion for "Hand on chin" (top) and the puppet manipulation (middle). The input puppet motion is retargeted to the human skeleton (bottom).*

formulated as

$$P(\mathbf{x}_q|\mathbf{l},\Phi) = \mathcal{N}(\mathbf{m}_\mathbf{l}, \sigma^2_\mathbf{l}), \tag{5}$$

where

$$\mathbf{m}_\mathbf{l} = \mathbf{X}\mathbf{K}^{-1}\mathbf{k}(\mathbf{l}), \quad \sigma_\mathbf{l} = k(\mathbf{l},\mathbf{l}) - \mathbf{k}(\mathbf{l})^\mathsf{T}\mathbf{K}^{-1}\mathbf{k}(\mathbf{l}),$$

and $k(\mathbf{l}_i,\mathbf{l}_j)$ is the (i,j)-th element of the kernel matrix, and $\mathbf{k}(\mathbf{l})$ is a vector whose i-th element is $k(\mathbf{l},\mathbf{l}_i)$. Finally, the error function between the query feature vector $\mathbf{x}_q$ and $\mathcal{M}$ that is modeled with Φ and $\mathbf{L}$ is defined as

$$\begin{aligned} &Err([\Phi,\mathbf{L}],\mathbf{x}_q) \\ &= \min(-\log(P(\mathbf{x}_q|\mathbf{l}_1,\Phi),\cdots,-\log(P(\mathbf{x}_q|\mathbf{l}_{N^d},\Phi)). \end{aligned} \tag{6}$$

By using Equation (6) in Equations (2) and (3), the distance between the input puppet and the target human motion can be computed for SM and DSPM, respectively.

4.5. Joint Velocity Histogram

The subspace methods evaluate the similarities of human postures. To take into account the temporal joint motion, we can optionally include an M-dimensional joint velocity histogram, Jvh, that describes the magnitude of every joint motion.

$$Jvh = \sum_{n=1}^{N-1} \left(|x^1_{n+1} - x^1_n|, |x^2_{n+1} - x^2_n|, ..., |x^M_{n+1} - x^M_n| \right), \tag{7}$$

where x^k_n is the k-th dimensional component of the input feature vector $\mathbf{x}$ at time n. The similarity of $Jvh_\mathcal{A}$ and $Jvh_\mathcal{B}$ of motion $\mathcal{A}$ and $\mathcal{B}$ is given by the Bhattacharyya distance [Kai67].

$$Sim(Jvh_\mathcal{A}, Jvh_\mathcal{B}) = \sum_{i=1}^{M} \sqrt{\frac{Jvh^i_\mathcal{A}}{\sum_{j=1}^M Jvh^j_\mathcal{A}} \cdot \frac{Jvh^i_\mathcal{B}}{\sum_{j=1}^M Jvh^j_\mathcal{B}}}, \tag{8}$$

where Jvh^i is the i-th component of the histogram.

5. Experiments

The system consists of the puppet interface and a PC (Windows XP, Intel Core2 Duo 2.66 GHz, 4GB RAM). Data capturing and rendering were implemented with C++ and the motion retrieval algorithms were run on Matlab R2008b. We did not use any parallel computing techniques.

5.1. Evaluation of DSPM

We compare the performance of DSPM with conventional pose-based retrieval techniques. We prepared 37 motion clips (932 seconds in total) containing 29 everyday behaviors from Subject #14 of CMU Motion Capture Database [CMU], and applied the segmentation method [BSP*04] to the database. We obtained 269 distinct behaviors.

We recruited five subjects for this experiment (Figure 5). They did not have any experience with animation design or techniques. We conducted the experiments with the following two protocols:

Protocol 1: Target behaviors were chosen from the database and shown to the subjects through a 3D viewer. Then the subjects performed the behaviors using the puppet interface without looking at the motion.

Protocol 2: Verbal descriptions of the target behaviors were provided to the subjects. Then the subjects performed the behaviors with the puppet interface without looking at the descriptions.

Figure 6 shows an example of target human motion and the puppet manipulation by the subject.

We compared DSPM (PCA) and DSPM (GP-LVM) with the subspace methods, namely, SM (PCA) and SM (GP-LVM), MSM, and DTW. MSM evaluates the largest canonical angle of two subspaces as the similarity of two datasets. Given the PCA latent spaces of puppet motion $\mathcal{Q}$ and human motion $\mathcal{H}$ as $\mathbf{Q} = [\mathbf{q}_1\,\mathbf{q}_2\cdots\mathbf{q}_M]$ and $\mathbf{H} = [\mathbf{h}_1\,\mathbf{h}_2\cdots\mathbf{h}_M]$

Table 1: *Results with Protocol 1 (P1) and Protocol 2 (P2). The numbers with and without brackets indicate how many subjects thought the target motion was ranked within the top 10 and 5 retrieval results, respectively.*

	DTW		**SM (PCA)**		SM (GP-LVM)		**MSM**		**DSPM (PCA)**		DSPM (GP-LVM)	
	P1	P2	P1	P2	P1	P2	P1	P2	P1	P2	P1	P2
right punch	5 (4)	4 (4)	5 (5)	4 (3)	5 (5)	5 (5)	4 (3)	3 (3)	5 (3)	1 (1)	3 (3)	3 (2)
squats	0 (0)	0 (0)	4 (3)	3 (2)	0 (0)	0 (0)	2 (2)	4 (2)	4 (4)	3 (1)	0 (0)	2 (0)
wash window 1	4 (3)	3 (3)	5 (3)	5 (5)	2 (0)	3 (2)	5 (5)	5 (5)	5 (5)	5 (5)	3 (2)	3 (3)
wash window 2	3 (1)	4 (4)	5 (5)	5 (5)	4 (1)	5 (1)	5 (5)	5 (5)	5 (5)	5 (5)	5 (4)	3 (3)
side twists	4 (3)	3 (3)	4 (4)	5 (5)	0 (0)	0 (0)	4 (4)	5 (5)	5 (5)	5 (4)	1 (1)	1 (1)
climb 2 steps	3 (3)	3 (2)	4 (3)	5 (3)	0 (0)	0 (0)	5 (4)	5 (5)	5 (5)	4 (3)	4 (3)	1 (1)
point by right hand	3 (1)	2 (0)	5 (4)	5 (3)	2 (0)	3 (2)	5 (5)	5 (5)	5 (5)	5 (5)	4 (2)	3 (1)
sit on high stool	3 (1)	3 (3)	5 (5)	5 (5)	0 (0)	0 (0)	5 (4)	4 (3)	4 (3)	5 (4)	4 (4)	3 (1)
sit and swing legs	5 (5)	5 (5)	4 (4)	5 (5)	0 (0)	0 (0)	5 (5)	5 (5)	5 (5)	5 (5)	2 (2)	3 (2)
hand on chin	5 (2)	2 (2)	3 (3)	2 (2)	0 (0)	0 (0)	1 (1)	3 (1)	5 (4)	5 (4)	4 (4)	4 (4)
drink soda	3 (3)	2 (2)	4 (2)	3 (2)	0 (0)	0 (0)	5 (5)	5 (4)	5 (4)	5 (5)	1 (0)	0 (0)
Average	3.5 (2.4)	2.8 (2.5)	**4.4 (3.7)**	**4.3 (3.6)**	1.2 (0.5)	1.5 (0.9)	**4.2 (3.9)**	**4.5 (3.9)**	**4.8 (4.4)**	**4.4 (3.8)**	3.4 (2.3)	2.4 (1.6)

Table 2: *Objective evaluation with Protocol 1. The numbers show the average ranks of the ground-truth behaviors in the retrieval results.*

	DTW	SM (PCA)	SM (GP-LVM)	MSM	**DSPM (PCA)**	DSPM (GP-LVM)
right punch	15.8	8.6	75.8	84.8	5.6	25.2
squats	186.8	8.4	66.8	27	7.6	45.8
wash window 1	37.4	61.2	134.4	17.6	8.8	117.0
wash window 2	43.2	30.6	126.4	50.4	10.2	48.2
side twists	31.2	22.0	83.0	10.0	1.8	62.4
climb 2 steps	47.8	32.6	80.2	9.0	9.0	13.6
point by right hand	16.2	6.6	135.8	2.4	4.8	47.0
sit on high stool	67.4	12.4	143.8	19.0	23.6	70.4
sit and swing legs	1.8	1.2	106.6	1.2	1.0	33.4
hand on chin	5.6	12.8	86.2	33.8	4.2	6.0
drink soda	15.6	12.8	179.4	1.0	5.2	106.4
Average	42.6	19.0	110.8	23.3	**7.4**	52.3

respectively, the squared cosine value of the first canonical angle of the two subspaces can be computed as the largest eigenvalue of a matrix $\mathbf{H}^{\mathsf{T}}\mathbf{Q}\mathbf{Q}^{\mathsf{T}}\mathbf{H}$. This largest eigenvalue is considered as a distance metric of MSM. We used five dimensional subspaces for the PCA-based methods, and a radial basis function kernel and a two dimensional latent space for the GP-LVM-based methods. We used $K = 50$ for random sampling of DSPM. We tested other parameters for these experiments, namely, five dimensional latent space for the GP-LVM and $K = 100$ and 150 sample frames in DSPM, however, they did not change the retrieval results significantly.

We asked the subjects if the desired behaviors were included within top 5 and 10 retrieval results. The results of Protocols 1 and 2 are summarized in Table 1. SM (PCA), DSPM (PCA) and MSM achieved better results than DTW, SM (GP-LVM) and DSPM (GP-LVM).

We also performed an objective evaluation with Protocol 1 by checking where the ground-truth behavior was ranked in the retrieval results (Table 2). Similarly to the results of the subjective evaluation, DSPM (PCA) outperforms other methods, followed by SM (PCA).

Table 3 shows the average processing time for retrieval. MSM was the fastest (0.2 sec), but DSPM (PCA) also ran at a reasonable speed (2.9 sec). DSPM (PCA) was faster than SM (PCA), because DSPM (PCA) randomly samples data and SM (PCA) does not. The other methods were too slow for interactive motion retrieval.

5.2. Experiments with Larger Database

Because MSM and DSPM (PCA) outperformed other methods in terms of accuracy and computational time, we conducted another experiment with Protocol 2 using a larger

Table 3: *Computational time (sec).*

DTW	57.1
SM (PCA)	8.6
SM (GP-LVM)	189.7
MSM	0.2
DSPM (PCA)	2.9
DSPM (GP-LVM)	157.7

Table 4: *Computational time for larger database (sec).*

	Matlab R2008b	C++ (Visual Studio 2008)
MSM	1.7	0.0023
MSM + Jvh	7.1	0.0093
DSPM (PCA)	37.1	0.54
DSPM (PCA) + Jvh	43.0	0.76

database for these two methods. The database consisted of 479 motion clips (approximately 2.3 hours in total) collected from CMU Motion Capture Database (Subject #13, 14, 15, 79, 80, 81, 85, 87, 88, 89, 90, 125, 126, 139, 141, 143), and 3065 distinct behaviors were obtained after segmentation. We recruited five new subjects, and they performed 25 behaviors categorized as everyday, exercise and acrobatic behaviors. The subjects did not have any experience with animation design or techniques. The parameters of each method were the same as in the previous experiment.

In this experiment, we tested the joint velocity histogram combined with SM (PCA) and DSPM (PCA), as well as only SM (PCA) and DSPM (PCA). To combine these results, we assumed a Gaussian probability distribution for DSPM and the joint velocity histogram, p_{DSPM} and p_{Jvh}, and used their joint probability $p(\mathcal{Q},\mathcal{H})$ as a similarity between puppet and human motion $\mathcal{Q}$ and $\mathcal{H}$. The probabilities for DSPM and the joint velocity histogram are represented as

$$p_{DSPM}(\mathcal{Q},\mathcal{H}) = \exp(-\alpha \cdot D_{DSPM}(\mathcal{Q},\mathcal{H})^2), \quad (9)$$

$$p_{Jvh}(\mathcal{Q},\mathcal{H}) = \exp(1 - Sim(Jvh_{\mathcal{Q}}, Jvh_{\mathcal{H}})^2), \quad (10)$$

where α is a positive regularization factor. We empirically set $\alpha = 0.01$, because typically the range of D_{DSPM} is $[0,10]$. Therefore, the joint probability is represented as

$$p(\mathcal{Q},\mathcal{D}) = p_{DSPM}(\mathcal{Q},\mathcal{D}) \cdot p_{Jvh}(\mathcal{Q},\mathcal{D}). \quad (11)$$

Tables 5 shows the experimental results. As in the previous experiment, we recorded how many subjects thought that the target motion was ranked within the top 10 (without brackets) and 5 (with brackets) retrieval results. Particularly for acrobatic and exercise behaviors, DSPM (PCA) outperformed MSM. Adding the joint velocity histogram to the similarity metric further improved the results.

Table 4 shows the average processing time for retrieval with the large database. In this experiment, we used a C++ implementation (MS-Visual Studio 2008, no parallel computation) as well as the Matlab implementation. MSM was faster than DSPM (PCA) and DSPM (PCA) + Jvh. The joint velocity histogram requires additional computation time, but the benefit in performance outweighs the cost for most applications.

Table 5: *Results for larger database with Protocol 2. The numbers with and without brackets indicate how many subjects thought the target motion was ranked within top 10 and 5 retrieval results, respectively.*

	MSM	MSM +Jvh	DSPM (PCA)	DSPM (PCA)+Jvh
right punch	0 (0)	1 (0)	2 (2)	3 (3)
squats	1 (1)	1 (0)	4 (2)	4 (3)
wash window 1	5 (5)	5 (5)	5 (5)	5 (5)
wash window 2	5 (5)	5 (5)	5 (5)	5 (5)
side twists	3 (2)	2 (2)	4 (4)	5 (4)
climb 2 steps	4 (4)	4 (3)	4 (2)	4 (2)
point by right hand	5 (2)	5 (3)	5 (4)	5 (4)
sit on high stool	3 (3)	3 (3)	2 (1)	3 (1)
sit and swing legs	5 (3)	4 (3)	5 (4)	5 (4)
hand on chin	0 (0)	1 (0)	4 (3)	5 (3)
drink soda	4 (1)	4 (1)	3 (3)	4 (3)
hand stand kicks	3 (3)	3 (2)	5 (5)	5 (5)
backflip	3 (2)	4 (1)	4 (4)	4 (4)
cartwheel	3 (2)	3 (1)	4 (4)	5 (4)
spins	0 (0)	0 (0)	0 (0)	1 (0)
handstands	4 (3)	3 (1)	3 (3)	4 (3)
somersault	4 (2)	4 (1)	3 (2)	3 (2)
jump kick	2 (1)	2 (0)	0 (0)	2 (2)
front hand flip	4 (4)	4 (2)	5 (5)	5 (5)
rug pull fall	5 (2)	4 (3)	5 (5)	5 (4)
Russian dance	1 (0)	1 (0)	5 (5)	5 (4)
breast stroke	0 (0)	0 (0)	5 (5)	5 (5)
free style	0 (0)	0 (0)	5 (5)	5 (5)
back stroke	1 (0)	1 (0)	5 (5)	5 (5)
fly style	1 (1)	1 (0)	5 (5)	5 (5)
Average	2.2 (1.4)	2.6 (1.4)	3.9 (3.5)	4.3 (3.6)

6. Discussion and Conclusion

In this paper, we developed a puppet interface system for retrieval of motion capture data. We designed the hardware for an instrumented puppet that uses rotational sensors at the joints and an orientation sensor to measure body orientation. We introduced a novel motion retrieval algorithm called the dual subspace projection method that outperforms conventional pose-based retrieval methods. Our system is easy to use and allows the users to find desired behaviors intuitively. The computational efficiency of our method enables the user to interactively retrieve desired motions from databases. We validated the system through experiments with ten novice subjects.

Puppet interface: All users were satisfied with the puppet interface for its intuitiveness and usability. They also felt that it had enough DOFs to express the desired behaviors.

All users appeared to be accustomed to the interface within five minutes. One user said that it was good to show puppet motion in realtime during the performance. Some users thought that our system was useful for searching motions that are difficult to describe with keywords. Some subjects pointed out that manipulating the puppet was quite easy for behaviors that required one or two limbs, but became difficult for complicated behaviors requiring three or four limbs simultaneously. In our experiment, two subjects avoided this issue by attaching the puppet to their body with a strap. Another possibility is to introduce a *layered* retrieval system, in which the users would specify poses part-by-part while the system gradually refines the retrieval results. An active puppet interface [YSC*11], where each joint can be actuated by a motor, might be useful to play back motion the users performed previously. As the users develop more experience with manipulating the puppet, this problem might also be reduced. The system requires only a good enough match to select the desired motion over similar motions in the database, and it does not require a complete performance with motion of all limbs. As the users gain familiarity with the system, they may discover which elements of the performance matter most to a good match.

We also found that most subjects were satisfied with the puppet's DOFs to express motions. Some subjects wanted a neck joint, but later noticed that it was not necessary for most behaviors. During the experiments, some users adjusted the tightness of the bolts to fix joint angles. This capability helped them perform complicated behaviors. For example, to express "hand on chin" motion, some subjects first adjusted the pose of the leg joints and tightened the bolts, and then started to perform an action of two arms (Figure 6). The subjects also could retrieve acrobatic behaviors such as "backflip" and "cartwheel," which they could not perform with their own bodies. These observations demonstrate the usability of our puppet design.

System design and cost: As we discussed in Section 2, there have been a few solutions that allow the specification of articulated motion easily and cheaply. Our interface hardware costs less than 100 USD without the orientation sensor. This price is quite cheap compared to motion capture or other special measurement systems. In our development, we used an IMU-based orientation sensor (InertiaCube3 by InterSense Inc.). Recent orientation sensors are commercially available at a reasonable price (around 100 USD) and have very high accuracy. These sensors will be useful for future development of puppet interfaces. We also want to develop a wireless puppet hardware so that the users can specify acrobatic behaviors more easily.

Motion retrieval algorithm: In our experiments, DSPM (PCA) outperformed other methods in terms of accuracy, and the computational cost was acceptable for interactive retrieval. The joint velocity histogram also improved the accuracy further. MSM had the second highest accuracy and achieved the quickest retrieval. This accuracy is surprising to us, because MSM does not consider pose similarity but only considers a similarity of joint movements. The accuracy could perhaps be further improved if we can effectively combine MSM and methods that consider pose similarities. Interestingly, GP-LVM was not so effective as SM and DSPM. We believe that this occurred because GP-LVM generates a very tightly fitting latent space and that it does not allow for large differences between human and puppet motion. In contrast, PCA generates a more approximate latent space. and worked well in our retrieval task. We also confirmed that the DSPM approach increased accuracy for both PCA and GP-LVM.

Limitations: Some subjects pointed out a few drawbacks of our interface. One is that our interface was not good for grasping actions because of the lack of wrist joints. It is possible to install additional sensors at wrists and ankles. However, we wonder whether or not the users can actually manipulate these joints, because motion of these joints are subtle compared to other joints. We would need further subject tests to investigate this issue.

Another concern from the subjects was that that they could not control the translation of the puppet. Our system has only an orientation sensor and cannot measure translation. In particular, our system cannot differentiate between "lift up legs" and "sit on rear" because the difference is mainly in the body translation. While designing this system, we discussed introducing a camera (vision sensor) or a magnetic sensor to measure the absolute position of the puppet. However, the former would have a problem with occlusion by hands, and the latter is sensitive to metal in the environment. If we can use some position sensing system, it should be possible to increase accuracy, particularly for differentiating actions such as walking and running in which translation plays an important role. We plan to investigate this possibility with state-of-the-art sensing technologies and further subject tests. Our current system is designed to find behaviors from a database and therefore, the system cannot identify finer details of similar behaviors (e.g. various styles of a single behavior). Further improvements for both the hardware and the retrieval algorithm are necessary for such tasks.

Acknowledgements

We would like to thank Kei Ninomiya for the development of the preliminary system and support for the experiments, Moshe Mahler for his help in modeling and rendering, and Justin Macey for his help in recording the motion capture data. This work was supported in part by the JST Precursory Research for Embryonic Science and Technology (PRESTO) program, the MIC Strategic Information and

Communications R&D Promotion Programme (SCOPE), Japan, and NSF Grant CNS-0931999 and CCF-0811450.

References

[AF02] ARIKAN O., FORSYTH D. A.: Interactive motion generation from examples. *ACM Transactions on Graphics 21*, 3 (2002), 483–490. 2

[AFO03] ARIKAN O., FORSYTH D. A., O' BRIEN J. F.: Motion synthesis from annotations. *ACM Transactions on Graphics 22*, 3 (2003), 402–408. 2

[Ani] ANIMATE ME, INC.: Animeeple. http://www.animeeple.com. 1

[BSP*04] BARBIČ J., SAFONOVA A., PAN J., FALOUTSOS C., HODGINS J. K., POLLARD N. S.: Segmenting motion capture data into distinct behaviors. In *Proc. Graphics Interface* (2004), pp. 185–194. 4, 6

[BW95] BRUDERLIN A., WILLIAMS L.: Motion signal processing. In *Proc. ACM SIGGRAPH 95* (1995), pp. 97–104. 2

[CH05] CHAI J., HODGINS J. K.: Performance animation from low-dimensional control signals. *ACM Transactions on Graphics 24*, 3 (2005), 686–696. 2

[CMU] CMU GRAPHICS LAB MOTION CAPTURE DATABASE: http://mocap.cs.cmu.edu. 1, 6

[DGL09] DENG Z., GU Q., LI Q.: Perceptually consistent example-based human motion retrieval. In *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2009), pp. 191–198. 3

[EPO95] ESPOSITO C., PALEY W. B., ONG J.: Of mice and monkeys: a specialized input device for virtual body animation. In *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics* (1995), pp. 109–114. 2

[FGDJ08] FENG T.-C., GUNAWARDANE P., DAVIS J., JIANG B.: Motion capture data retrieval using an artist's doll. In *Proc. International Conference on Pattern Recognition* (2008). 2

[FYM05] FUJIMAKI R., YAIRI T., MACHIDA K.: An approach to spacecraft anomaly detection problem using kernel feature space. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (2005), pp. 401–410. 5

[GMHP04] GROCHOW K., MARTIN S. L., HERTZMANN A., POPOVIĆ Z.: Style-based inverse kinematics. *ACM Transactions on Graphics 23*, 3 (2004), 522–531. 5

[HK09] HO E. S. L., KOMURA T.: Indexing and retrieving motions of characters in close contact. *IEEE Transactions on Visualization and Computer Graphics 15*, 3 (2009), 481–491. 3

[IWZL09] ISHIGAKI S., WHITE T., ZORDAN V. B., LIU C. K.: Performance-based control interface for character animation. *ACM Transactions Graphics 28*, 3 (2009), 1–8. 2, 5

[JWB*99] JOHNSON M. P., WILSON A., BLUMBERG B., KLINE C., BOBICK A.: Sympathetic interfaces: Using a plush toy to direct synthetic characters. In *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems* (1999), pp. 152–158. 2

[Kai67] KAILATH T.: The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology COM-15* (1967), 52–60. 6

[KG04] KOVAR L., GLEICHER M.: Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics 23*, 3 (2004), 559–568. 2

[KGP02] KOVAR L., GLEICHER M., PIGHIN F.: Motion graphs. *ACM Transactions on Graphics 21*, 3 (2002), 473–482. 2

[KHSW95] KNEP B., HAYES C., SAYRE R., WILLIAMS T.: Dinosaur input device. In *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems* (1995), pp. 304–309. 2

[Law04] LAWRENCE N. D.: Gaussian process latent variable models for visualisation of high dimensional data. In *Proc. Neural Information Processing Systems* (2004), p. 2004. 5

[LCR*02] LEE J., CHAI J., REITSMA P. S. A., HODGINS J. K., POLLARD N. S.: Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics 21*, 3 (2002), 491–500. 2

[LGY*06] LI Q. L., GENG W. D., YU T., SHEN X. J., LAU N., YU G.: MotionMaster: authoring and choreographing kung-fu motions by sketch drawings. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), pp. 233–241. 2

[MCN*11] MAZALEK A., CHANDRASEKHARAN S., NITSCHE M., WELSH T., CLIFTON P., QUITMEYER A., PEER F., KIRSCHNER F., ATHREYA D.: I'm in the game: embodied puppet interface improves avatar control. In *Proc. International Conference on Tangible, Embedded, and Embodied Interaction* (2011), pp. 129–136. 4

[MP07] MCCANN J., POLLARD N. S.: Responsive characters from motion fragments. *ACM Transactions on Graphics 26*, 3 (2007). 2

[MR06] MÜLLER M., RÖDER T.: Motion templates for automatic classification and retrieval of motion capture data. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), pp. 137–146. 2

[MRC05] MÜLLER M., RÖDER T., CLAUSEN M.: Efficient content-based retrieval of motion capture data. *ACM Transactions on Graphics 24*, 3 (2005), 677–685. 2

[RSH*05] REN L., SHAKHNAROVICH G., HODGINS J. K., PFISTER H., VIOLA P.: Learning silhouette features for control of human motion. *ACM Transactions on Graphics 24*, 4 (2005), 1303–1331. 2

[SH08] SLYPER R. Y., HODGINS J. K.: Action capture with accelerometers. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2008). 2

[Smi] SMITH MICRO SOFTWARE: Poser. http://poser.smithmicro.com. 1

[TBvdP04] THORNE M., BURKE D., VAN DE PANNE M.: Motion doodles: An interface for sketching character animation. *ACM Transactions on Graphics 23*, 3 (2004), 424–431. 2

[YAH10] YAMANE K., ARIKI Y., HODGINS J.: Animating non-humanoid characters with human motion data. In *Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (2010). 5

[YFM98] YAMAGUCHI O., FUKUI K., MAEDA K.: Face recognition using temporal image sequence. In *Proc. IEEE International Conference on Automatic Face and Gesture Recognition* (1998), pp. 318–323. 3

[YP03] YIN K., PAI D. K.: FootSee: an interactive animation system. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), pp. 329–338. 2

[YSC*11] YOSHIZAKI W., SUGIURA Y., CHIOU A. C., HASHIMOTO S., INAMI M., IGARASHI T., AKAZAWA Y., KAWACHI K., KAGAMI S., MOCHIMARU M.: An actuated physical puppet as an input device for controlling a digital manikin. In *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems* (2011), pp. 637–646. 2, 9