Motion Compression using Principal Geodesics Analysis

Maxime Tournier, Xiaomao Wu, Nicolas Courty, Élise Arnaud, Lionel Reveret

▶ To cite this version:

Motion Compression using Principal Geodesics Analysis

M. Tournier[1], X. Wu[1], N. Courty[2], E. Arnaud[1] and L. Revéret[1]

[1]Laboratoire Jean Kuntzmann, INRIA, France
[2]Université de Bretagne-Sud

Abstract

Due to the growing need for large quantities of human animation data in the entertainment industry, it has become a necessity to compress motion capture sequences in order to ease their storage and transmission. We present a novel, lossy compression method for human motion data that exploits both temporal and spatial coherence. Given one motion, we first approximate the poses manifold using Principal Geodesics Analysis (PGA) in the configuration space of the skeleton. We then search this approximate manifold for poses matching end-effectors constraints using an iterative minimization algorithm that allows for real-time, data-driven inverse kinematics. The compression is achieved by only storing the approximate manifold parametrization along with the end-effectors and root joint trajectories, also compressed, in the output data. We recover poses using the IK algorithm given the end-effectors trajectories. Our experimental results show that considerable compression rates can be obtained using our method, with few reconstruction and perceptual errors.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism–Animation—Vision and Scene Understanding [I.2.10]: Motion—:

1. Introduction

Motion capture has became a ubiquitous technique in any domain that requires high quality, accurate human motion data. With the outcome of massively multiplayer online games, the transmission of huge quantities of such data can become problematic. Furthermore, in order to provide the user with a wide range of animations, several motion capture sequences often have to be used at once: motion data is either picked from a motion database, or constructed using blending or learning on existing data. In that case, a compact and easily editable representation of motion data could drastically improve user experience while decreasing the storage needs.

Raw motion capture data are indeed very large: they consist in the aggregation of sampled markers trajectories or joints orientations across time. With a sampling rate of 120 Hz and about 40 markers for the skeleton, the data size rapidly grows large. However, human motion exhibits inherent redundancies that can be exploited for compression purposes:

- **Temporal coherence**, thanks to which the motion can be keyframed with few loss of information,
- **Joints motion correlations**, which allows the representation of the motion in a smaller subspace.

We propose a novel, lossy compression of human motion data that exploits both of them. We first approximate the poses manifolds using Principal Geodesics Analysis (PGA) in the configuration space of the skeleton. This model yields a reduced, data-driven pose parametrization that is then used in an Inverse Kinematics (IK) algorithm to recover poses given end-joints positions only. Thanks to this algorithm, we are able to reconstruct the motion given only end-joints trajectories and the root joint's positions and orientations. The key idea is thus to perform compression by only storing this compact model along with the end-joints and root-joints trajectories. As we constraint end-joints positions during the decompression, typical compression artifacts such as foot skating are automatically reduced. By modifying these end-joints trajectories, one can also easily edit motions compressed using our method. Our experimental results show that significant compression rates with few distortion can be obtained using this approach, while keeping the possi-

bility to easily and naturally edit synthesized motion. Our method is easy to implement and runs quite fast on modern machines.

The rest of the paper is organized as follows: the related work is reviewed in section 2. We present an overview of the compression technique in section 3, followed by a brief presentation of the non-linear tools and their use in our system. Section 4 is dedicated to experimental results in which compression performances are evaluated. We conclude in section 5 by a discussion of the proposed method and possible future work.

2. Background and Related Work

2.1. Motion Compression

Though recent works on motion capture data compression can be found, the problem of motion compression has been mainly focused on animated meshes compression so far: those high-dimensional data often present high spatial and temporal coherence that can be exploited to reduce the data size. [Len99] detects parts of the mesh with rigid motion to encode only the transformation and the residuals. Correlations that may exist in parts of the moving object have also been exploited through the use of Principal Component Analysis (PCA) [SSK05] to compress the mesh vertices.

Skeleton motion also exhibits those cross-bones correlations. These are mainly exploited in optimization frameworks as they allow for search space dimension reduction. [SHP04] apply PCA on a group of similar motions in order to synthesize motion in such a reduced space. [GMHP04] use a probabilistic latent variable space to perform inverse kinematics that preserves stylistic properties. [LM06] detect motion segments in which joints positions lie in a reduced linear subspace and use PCA to reduce dimensionality for compression purposes.

Motion capture data also exhibits temporal coherence which can also be exploited to acheive compression: [LM06] use spline keyframing to compress the PCA projections of markers in motion segments. [Ari06] uses splines to represent global markers trajectories. The control points for a whole motion database are then compressed using clustered PCA. In both cases, working with global marker positions requires an additional pass of optimization to keep the bone length constant along the synthesized motions. Other methods use rotational data: [BPP07] adapt standard wavelet compression on joint angles by automatically selecting the basis elements in a way that minimizes quadratic error. However, high compression ratios can result in strange reconstructed paths due to the use of Euler angles.

Any lossy motion capture compression method, being orientations-based or positions-based, introduces errors that are likely to introduce various perceptual artifacts. The most striking is probably *foot skating*, which greatly penalizes

the visual quality of synthesized motions. This artifact can be corrected using inverse kinematics(IK) techniques. However, using *style-based* IK [GMHP04] is often needed in order to correct the motion while preserving its visual identity.

2.2. Motion Metric

As with any lossy compression system, a central problem with motion capture data compression is the error metric used to evaluate the quality of the results. The problem in our case is that the metric should take perceptual features into account, which is a difficult task. While it is commonly accepted that the standard L^2 norm over markers positions is a weak indicator of the perceptual closeness of two animations, few works propose an alternative, efficient metric. [RP03] propose a study of user sensitivity to errors considering only ballistic motions. [RPE*05] try to evaluate the natural aspect of an animation. To do so, 3 classes of metrics are distinguished:

- Heuristic rules, that penalize the score of an animation when violated (*e.g.* physical laws)
- Perceptual metrics that highlight artifacts noticed by users (*e.g.* foot skating)
- Classifiers-based metrics trained on large data sets

The first two usually fail to quantify the natural aspect, or the style of an animation, but are good at detecting precise artifacts. The latter is based on the assumption that a human will perceive a motion as natural if it has already been seen a lot of times. On the contrary, an unusual motion will be perceived as unnatural. Such metrics often detect stylistic closeness successfully, but are highly dependent on the data set used for the training: they will fail for a natural motion that is not in the data set. Moreover, local physical anomalies or artifacts are often not detected. As a matter of fact, finding an accurate and robust metric for human motion perception remains, to the best of our knowledge, an open problem.

2.3. Non-linear Analysis

As mentioned earlier, two natural ways of compressing motion data are to exploit both temporal coherence and correlations in the motion of parts of the skeleton. To achieve this, one typically uses multiresolution and dimension reduction techniques. While well-known theoretical frameworks for these are available in the case of data lying in a linear space (such as wavelets, PCA), their extension to non-linear spaces (for instance, the space of rotations $SO(3)$) is not trivial and is a recent field of research.

[LS01] propose a *multiresolution* scheme for orientation data that allows editing, blending and stitching of motion clips. A potential application to compression is mentioned, though not developed. [RDS*05] generalize this scheme to symmetric Riemannian manifolds using exponential and logarithmic maps. The interpolating scheme used

can be seen as a special case of the so-called *lifting scheme* [Swe98]. The *lifting scheme* is an alternative way of defining wavelets and is presented in section 3.5. [RDS*05] propose an application to the compression of airplane headings with promising results.

The *dimension reduction* problem is often solved using descriptive statistical tools. Those tools typically yield a space that is more suitable for expressing the data: smaller dimension, orthogonal axis, most notably. The extension of known linear statistical tools to the non-linear case is not eased by the fact that many elementary results in the former case do not hold when dealing with more general spaces. For instance, the problem of finding the mean value of data lying on a sphere can no longer be expressed through probabilistic expected value, but has to resort to the minimization of geodesic distances [BF01]. Averaging rotations falls into this class of problem [Moa02].

Pennec [Pen06] gives basic tools for probabilities and statistics in the general framework of Riemannian manifolds. Fletcher [FLJ03], [FLPJ04] proposes a generalization of PCA to certain non-linear manifolds named Principal Geodesic Analysis (PGA), which consists in finding *geodesics* that maximize projected variance. He also presents an approximation of the analysis that boils down to a standard PCA in the tangent space at the mean of the data. It is presented in more details in section 3.3. An algorithm performing *exact* PGA for rotations is presented in [SCLS07], which shows that the number of principal geodesics needed for an exact reconstruction is not *a priori* bounded.

3. Proposed Method

3.1. Motivations - Overview

In this section we give an overview of our motion capture data compression method. Most approaches to human motion compression exploit global markers positions to achieve compression. While this has some advantages, such as speed and the use of well-known frameworks, the biggest drawback is that the constant bone-length of the skeleton cannot easily be guaranteed, which can introduce undesired limbs deformations. A post-processing pass is needed for this constraint to be enforced. Yet, this additional process can itself introduce artifacts. We want to address this problem by working on orientations rather than positions. However, because of the hierarchical nature of the skeleton, even slight errors in reconstructed orientations can lead to significant positions errors for end-joints. The most notable artifact of this kind is probably foot skating, which greatly penalizes the perceptual quality of synthesized animations. We intend to work around this by building a pose model from the animation clip: this model will allow us to synthesize poses that match given end-joints constraints, while staying close to the input data.

A pose is defined as a vector of rotations that describe the orientations of the skeleton's joints. It is therefore an element of $SO(3)^n$, where $n \in \mathbb{N}$ is the number of joints in the skeleton. Given a motion composed of $m \in \mathbb{N}$ poses, we use the *Principal Geodesics Analysis* to build a descriptive model of those pose data, keeping only the leading principal geodesics. This model is then used in an inverse kinematics system to synthesize poses that both match end-joints constraints *and* are close to the input data. Given this pose model, we only have to store the compressed end-joints trajectories as well as the root joint's positions and orientations (also compressed) in order to recover the motion using IK. The compression/decompression pipeline is presented on figure 1.
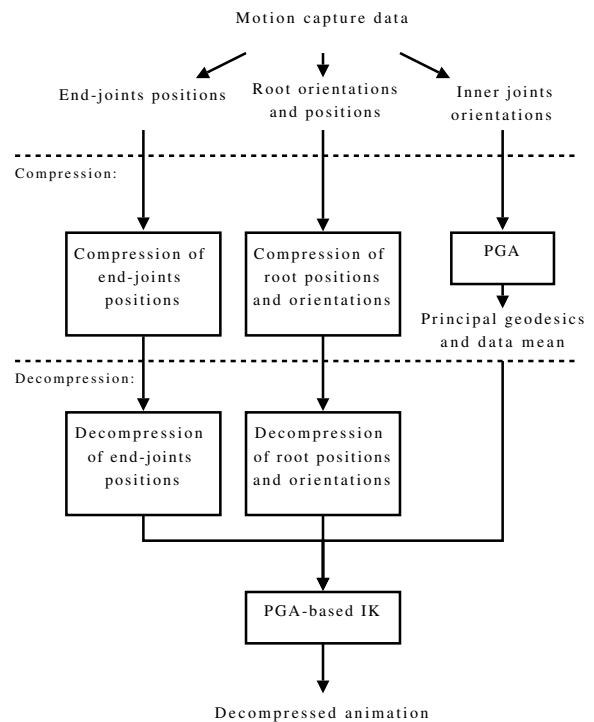

Figure 1: *Flow diagram for the compression pipeline*

We now briefly present the non-linear tools employed throughout this paper, as well as their use in our algorithm.

3.2. Lie Groups - Exponential Map

The space for three-dimensional rotations is a particular case of a Lie group. A Lie group is a group which is also a differentiable manifold, and for which the inverse and the group operations are differentiable. Such manifolds have been extensively studied and used in robotics to describe configuration spaces of articulated bodies [MSZ94]. We give a quick definition of the *exponential map* used throughout this paper. For a complete theoretical treatment, see [JJD00]. Let G be

a Lie group. The *exponential* map is a mapping from the tangent space of G at the identity (that is, the Lie algebra of G, $\mathfrak{g}$) to the group itself G. For every tangent vector of the Lie algebra $v \in \mathfrak{g}$, one can define a left-invariant vector field v^L by left-translation of v. Let $\gamma_v(t)$ be the unique maximal integral curve of such a vector field, the exponential map is then defined by $\exp(v) = \gamma_v(1)$. It is the unique one-parameter subgroup of G with initial tangent vector v. It is a diffeomorphism in a neighborhood of $0 \in \mathfrak{g}$, and the inverse mapping is called the *logarithm*. In the case of a Lie group endowed with a compatible Riemannian structure, such as $SO(3)$), the Riemannian and Lie exponential coincide. This means that we can compute geodesic curves (*i.e.* locally length-minimizing curves) as well as measuring lengths of such curves using the Lie exponential. The length of the shortest geodesic curve(s) between two points is called the *geodesic distance*.

For matrix Lie groups (*i.e.* subgroups of $GL_n(\mathbb{R})$), the exponential is defined by the usual exponential power series: $\exp(M) = \sum_{i \geqslant 0} \frac{M^i}{i!}$. For rotations, the sum of this series is known as the *Rodrigues'* formula [Gra98]. One can define an exponential mapping at any point $g \in G$ of the group simply by translating (in the group's operation sense) the standard exponential map: $exp_g(v) = g.exp(v)$.

In the case of data lying in an abstract manifold such as $SO(3)$, one can no longer define the mean as a weighted sum of elements. Instead, the *intrinsic* mean is defined as a point that minimizes the geodesic distance with respect to all the points considered. It can be computed by an optimization algorithm (see [FLJ03] or [Pen06]) which uses the exponential map and that usually converges in a few iterations.

3.3. Principal Geodesics Analysis

The Principal Geodesics Analysis is an extension of the Principal Component Analysis introduced by Fletcher in [FLJ03], [FLPJ04]. Its goal is to describe variability in Lie groups that can be given a Riemannian structure compatible with the algebraic one. Such groups include $(\mathbb{R}^n, +), (\mathbb{R}, \times), (SO(3), \circ)$ as well as any direct product between them. The idea behind PGA is to project data onto *geodesics* in a way that maximizes the projected variance. In the linear case, the geodesics are simply lines between two points, and the PGA then boils down to standard PCA. However, the projection onto a geodesic curve cannot be defined analytically in the general case, and therefore involves a minimization algorithm. In order to avoid this, Fletcher proposes to approximate the projection onto geodesics by a linear projection in the tangent space at the intrinsic mean of the data, using the exponential mapping at that point. Under this approximation, the principal geodesics' directions can be computed by a standard PCA of the linearized data. The fact that the linearization is done at the data mean conveniently guarantees its coordinate-invariance, but also ensures that the induced distortion is minimized.

We employed the PGA framework to describe the variability of the *inner* joints orientations during a motion. However, since the *exact* PGA computation in $SO(3)^n$ remains (to our best knowledge) an open problem, we used the approximation proposed by Fletcher to compute the principal geodesics' directions. These geodesics can be looked upon as the *eigenposes* of the skeleton during the sequence. We did not include the root joint's orientation in the analysis: indeed, it is only poorly correlated with the pose of the skeleton, and using it in the PGA can alter the resulting principal geodesics. As mentioned earlier, the pose of the skeleton is represented by a vector of the direct product $SO(3)^n$, where n is the number of joints of the skeleton. Applying the *approximate PGA* to the poses data from a motion with $m \in \mathbb{N}$ frames yields:

- The intrinsic mean of the data, $\mu \in SO(3)^n$
- $k \in \mathbb{N}$ tangent directions $(v_j)_{1 \leqslant j \leqslant k}$, where each $v_j \in \mathfrak{so}(3)^n \simeq \mathbb{R}^{3n}$ uniquely defines a geodesic of $SO(3)^n$
- A set of coordinates $T = (t_{i,j})$ where $1 \leqslant i \leqslant m$ and $1 \leqslant j \leqslant k$, where the ith row is the approximate projection of the ith pose over the k geodesics

The ith pose can then be recovered partly using the k leading geodesics with:

$$p_i = \mu . \prod_{j=1}^{j=k} e^{t_{i,j} . v_j}$$

This parametrizes the approximate poses manifold using the *canonical coordinates of the second kind* (ccsk, see [MSZ94]). One can think of the reconstruction formula as a weighted composition of the k first eigenposes. Note that here the exponential over the direct product $\mathfrak{so}(3)^n$ is used. Examples of principal geodesics extracted from motion capture data can be seen on figure 2.

As shown on figure 3, the number of principal geodesics needed to represent 99% of the input data variance is generally inferior to 20. For motions with stronger correlations, such as walking motions, 10 geodesics are in most cases enough to express 95% of the input variance.

By only considering the $k \leqslant n$ first modes of the PGA, we obtain a new reduced parametrization of a motion in terms of geodesics coordinates. We show next how such a reduced pose parametrization can be used to perform inverse kinematics.

3.4. PGA-based Inverse Kinematics

The reduced parametrization with geodesics coordinates allows us to define a function $f : \mathbb{R}^k \to \mathbb{R}^{3d}$ that maps a set of geodesics coordinates $x \in \mathbb{R}^k$ to the global space positions of $d \in \mathbb{N}$ end-effectors: $y \in \mathbb{R}^{3d}$. This function is the composition of the reduced pose parametrization by the ccsk $h : \mathbb{R}^k \to SO(3)^n$ and the classical direct kinematics function, which maps a skeleton pose to the global position of
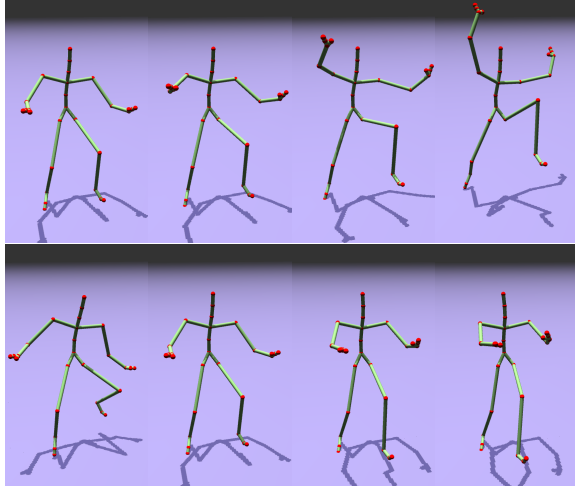
Figure 2: *Two examples of principal geodesics extracted from one break dance motion using PGA. Here, the corresponding poses are given for 4 different geodesic coordinates. The top left picture shows the mean pose (with a zero geodesic coefficient).*

Figure 3: *Histograms showing the numbers of geodesics needed to achieve given variance reconstruction, using approximate PGA. Top: 99% of variance for the whole CMU database. Bottom: 95% of variance for only walking motions.*

the d end-effectors, $g : SO(3)^n \rightarrow \mathbb{R}^{3d}$. The derivative of g at the pose $x \in SO(3)^n$ simply maps instant rotation vectors for each joint to the linear velocities of the end-effectors.
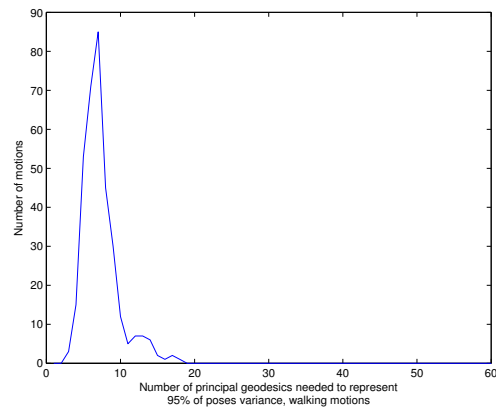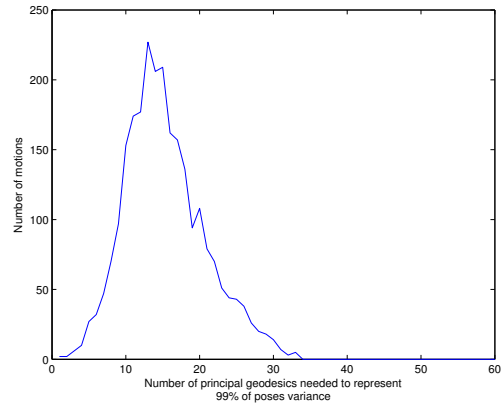
Since $h(x) = \mu . \prod_{j=1}^{j=k} e^{x_j . v_j}$ is a product of differentiable functions (the exponentials) in a Lie group, h is therefore differentiable. Each partial derivative of h with respect to x_j can be easily computed due to the ccsk parametrization using the adjoint Ad map over $SO(3)$ (see for instance [MSZ94], p.116). We are eventually able to compute the whole Jacobian matrix J_f of the function f using chain rule. We then use this Jacobian in a least square optimization method, such as the well-known Levenberg-Marquardt algorithm, in order to find the geodesic coordinates $x_j^\star$ that best match the given end-effectors constraints $y_0 \in \mathbb{R}^{3d}$:

$$x^\star = \underset{x \in \mathbb{R}^k}{\operatorname{argmin}}(\|f(x) - y_0\|^2)$$

The benefits of using this method are threefold:

- The optimization is done in a **much smaller space** than traditional IK (usually 30 degrees of freedom): this not only speeds up the process, but also better constraints the IK problem.
- The geodesics being principal poses modes, **the optimization naturally exploits correlations** between joints to reach the objectives, resulting in a more natural pose.
- The geodesics formulation allows a quick computation of the Jacobian J_f used in the optimization, thus eliminating the need for numerical differentiation.

The main drawback is that the geodesics yield a limited

reachable space: our IK works better in a neighborhood of the input data. However, since we are only interested in recovering poses belonging to the input data, this is not really a problem. Of course, the more geodesics we use, the larger the reachable space. In our experiments, 10-12 modes are generally sufficient to perform IK on 4 or 5 IK handles at once. Apart from compression, this IK system can also be used independently in real-time as seen on figure 4 for interactive motion editing.

In order to recover an animation sequence using the PGA-based IK, one needs the following data:

- The inner orientations mean and the k leading principal geodesics
- The end-joints trajectories across time
- The root joint's orientation and position across time

To recover each frame, we first express the end-joints trajectories in the root joint's frame, then perform IK as presented
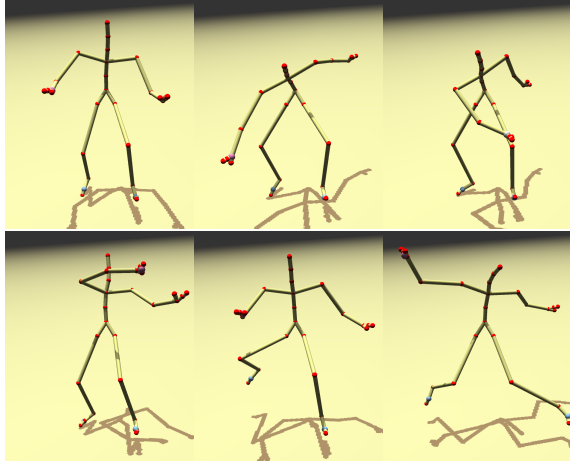
Figure 4: *The PGA-based IK system used in real-time. The input motion is a* break dance *sequence from the CMU database. There are three IK handles in this example that can be manipulated by the user: one on each foot and one on the right hand. Here the optimization is done using 10 geodesics.*

earlier. Doing so already results in a good compression of the data since we only have to store 7 trajectories (2 for the root and 5 for the end-joints) instead of more than 30 found in the original motion. Of course, since those trajectories present a high degree of temporal coherence, we will exploit it to further compress the motion data.

3.5. Multiscale Representation for Orientation Data

In order to compress the root's orientation, we use the multiscale representation for manifold data introduced in [RDS*05]. As a particular case of the so-called *lifting scheme* [Swe98], it can be summed up as follows. Let D be the set of data we want to represent in multiscale:

- D is first **partitioned** into two parts, A and B
- Data in A are then used to **predict** the data in B, using some prediction operator
- The **differences** between the *predictions B_A* and the actual B data form the *details*

The process is then iterated over the set A until there are no data remaining. By doing so, one creates a collection of decreasing-size levels of details. This "pyramid" is the multiscale representation of the original data. For this representation to be useful in compression, one generally wishes to partition the data so that the prediction step is as accurate as possible. In that case, the details needed to correct the prediction are small and can hence be omitted with few errors.

In the case of time-dependent orientation data, D can be represented by a collection of rotations $d = (d_i)_{1 \leqslant i \leqslant m}$,

where m is the number of samples. In order to exploit temporal coherence, we simply sub-sample the data by a factor 2 to partition the data. The prediction step is done using the tangent spline interpolation described in [RDS*05]. To interpolate between two rotations r_0 and r_1, we express r_{-1}, r_0, r_1, r_2 in $T_{r_0}(\mathrm{SO}(3))$ using the logarithm map. We then interpolate those tangent vectors *in the tangent space* using splines at $t = \frac{1}{2}$, and finally go back to $\mathrm{SO}(3)$ with the exponential map. The difference between the predicted value $\widetilde{r}_{\frac{1}{2}}$ and the original data r_{orig} is stored as $d = \log(\widetilde{r}_{\frac{1}{2}}^{-1}.r_{\mathrm{orig}})$. The original data can eventually be recovered using $r_{\mathrm{orig}} = \widetilde{r}_{\frac{1}{2}}.\exp(d)$.

We could have used simple SLERP [Sho85] prediction between r_0 and r_1 (as in [LS01]). This corresponds to simple linear approximation in the tangent space at one of the two points. However, this lead to a piecewise SLERP reconstructed signal when omitting levels-of-details, which presents discontinuities of the first derivatives that penalize the visual quality of the result. Instead, the use of tangent spline interpolation results in a smooth reconstructed signal even in the case of missing data.

3.6. Putting Everything Together

After the principal geodesics have been extracted from the input motion using approximate PGA, global end-joints trajectories can be compressed using any linear compression method. The root orientation is eventually compressed using the multiscale representation presented in section 3.5. For the sake of consistency, our implementation uses the presented multiscale scheme for both orientations and positions, but any suitable temporal coherence-based compression technique could work. The decompression phase consists in decompressing the global trajectories as well as the global root orientation, then expressing the end-joints positions in the root joint's frame, and eventually performing PGA-based IK to recover poses.

Let us now give an estimation of the data size needed to store an animation using our technique. Each of the k geodesics kept after the PGA is a vector of $\mathbb{R}^{3n}$, which is roughly the size of one motion frame. The mean value of the inner joints can also be stored as a vector of $\mathbb{R}^{3n}$ using the exponential map. The data needed for the PGA reconstruction can hence be stored in a $s_{\mathrm{PGA}} = (k+1) \times 3n$ matrix. The global root orientations and positions as well as the 5 end-joints' positions can easily be compressed by a factor $8 = 2^3$ by omitting 3 levels of details: each time we get rid of one level, we divide the data size by two. All those trajectories together can be encoded in a $s_{\mathrm{traj}} = 3(2+5) \times \frac{m}{8}$ matrix. Given an initial animation with size:

$$s_{\mathrm{orig}} = m \times 3(n+1) = O(m \times n)$$

whereas the compressed version using our algorithm, keeping k geodesics, will have the size:

$$s_{\mathrm{compressed}} = s_{\mathrm{PGA}} + s_{\mathrm{traj}} = O(m+n)$$

Examples of compression ratios obtained using our technique can be found in section 4.

As it involves an optimization process for decompressing each frame, our method is subject to the classic pitfalls of those algorithms: degenerate Jacobian, local extrema, smoothness of the reconstructed animation. In our experiments, we found that the first happen almost only when the constraints are out of the reachable space, and such cases do not happen when reconstructing the original signal. The smoothness of the solution can be enforced by adding a penalty term in the optimization so that the solution for the current frame decompression is searched in a neighborhood of the previous one:

$$x^\star = \underset{x \in \mathbb{R}^k}{\mathrm{argmin}}(\|f(x) - y_0\|^2 + \lambda \|x - x_{\mathrm{previous}}\|^2)$$

where λ is a given user threshold. This inevitably leads to drifting problems due to the existence of local extrema. In our experiments however, setting $\lambda = 1$ always gave correct results.

Before presenting the results obtained with our method, let us explicit what we feel are the main benefits of an approximate PGA over a PCA of the standard exponential maps for poses parametrization, since the two techniques might seem similar. First of all, using Fletcher's approximation at the intrinsic mean leads to a coordinate-invariant and distortion-minimized linearization of the data. This property improves the quality of the statistical analysis since it only depends on the input motion, and not on its *parametrization* (*e.g.* the choice of the reference pose). Secondly, the pose parametrization along the principal geodesics is easily differentiated, which is highly desirable for any pose-related optimization, since no finite differences approximations need to be computed.

4. Experimental Results

We present here the compression rates of our algorithm on selected motions from the Carnegie Mellon University's Graphic Lab motion capture database available at http://mocap.cs.cmu.edu. We chose motions with different characteristics of length, diversity, and dynamics as shown on table 1.

ID	Subject/Trial	Description	#Frames
1	09/06	Running, short	141
2	17/08	Walking, slow	6179
3	15/04	Various, dancing, boxing	22948
4	85/12	Breakdance	4499
5	17/10	Boxing	2783

Table 1: *Motion capture clips used in our experiments*

As stated in section 2.2, no really robust and efficient metric is available to assess the quality of the reconstructed animations. However, for the sake of results comparisons, we used a distortion rate as the one defined in [KG04] to evaluate the quality of the reconstruction. This distortion rate is defined as:

$$d = 100 \frac{\|A - \tilde{A}\|}{\|A - E(A)\|}$$

where A is the $m \times 3n$ matrix containing absolute markers' positions at each frame for the original motion, $\tilde{A}$ is the same matrix for the decompressed animation, and each row of $E(A)$ contains the mean markers' positions with respect to time.

Table 2 shows the obtained compression ratios and distortion rates for different combinations of geodesics numbers and trajectories levels of details. Table 3 shows the results obtained by [LM06], who holds the best compression rates at the time of writing. Note that we always used 5 end-joints in our tests, but more could be used if a higher quality is required. As expected, our method works best when the spatial and temporal coherence is high: a rather slow walking motion can be compressed 185 times with few reconstruction errors, whereas a highly dynamic breakdance sequence is only compressed 118 times for about the same distortion rate. This table shows that our technique allows substantial compression rates improvement over existing techniques, with very limited distortion. The accompanying video additionally shows that most of time, the differences between compressed and original animation are hard to find out, unless the two motions are displayed at the same position.

ID	1	2	3	4	5
#Geodesics	6	12	17	15	12
#LOD (root)	4/9	8/14	12/16	9/14	8/13
#LOD (end-joints)	4/9	8/14	12/16	9/14	9/13
Compression ratio	1:18	1:182	1:69	1:97	1:61
Distortion rate (%)	0.36	0.049	1.55	0.56	0.49
Decompression time (msec/frame)	7.88	16.2	30.6	20.42	15.97

Table 2: *Compression rates, distortion errors and decompression times for the selected motions using our technique. Different combinations of geodesics numbers and trajectories level of details are presented*

When too few geodesics are used in the IK, the reachable pose space gets too small and the synthesized poses sometimes fail to match all the given constraints. On the contrary, once enough geodesics have been selected, further increases on that number only result in a slower optimization time. In our experiments, 10 to 15 geodesics are sufficient in most cases to yield a large enough pose space. For long motions

Sequence	Compression ratio	Distortion rate	Decompr. time (msec/frame)
Jumping, bending, squats	1:55.2	5.1	0.7
Long breakdance sequence	1:18.4	7.1	0.7
Walk, stretches, punches, drinking	1:61.7	5.1	0.7
Walk, stretches, punches, kicking	1:56.0	5.4	0.7

Table 3: *Compression rates presented in [LM06] using PCA on motion segments. We achieve substantial compression rates improvement with fewer distortion, though our decompression pass takes longer.*

in which clear distinct motion behaviors occur, a segment-based approach similar to [LM06] may be used. This could improve the accuracy of each different pose model, resulting in more natural poses for each behaviors, and possibly better conditioning the Jacobian matrix used in the optimization. The transition between different models would have to remain smooth however, which is not an easy task. As expected, when the compression for the end-joints trajectories is set too high, artifacts start to appear as the feet contacts on the ground are smoothed too much. In the same way, too high compression over the root joints' position and orientations causes the skeleton to slide, as hung in the air. The acceptable compression ratio for those trajectories highly depends on the dynamics found in the animations. In any case, quantization may be used to compress the trajectories reconstruction errors while controlling additional overhead.

Since we are performing a statistical analysis to represent the motion's poses, strong outlier poses will be more difficult, if not impossible, to reconstruct correctly. In practice, such a situation could arise if the character holds a very specific pose for a very short amount of time with respect to the duration of the motion. We did not encounter such cases in our experiments. The compression time depends on the length of the animation, as it only involves the intrinsic mean of pose data calculation, and a PCA of the linearized rotations in the tangent space at that point. In practice, it is very inferior to the decompression time, during which an optimization is performed for each frame to reconstruct poses given end-joints constraints. Our implementation was realized in C++ on a Dell 390 workstation, with dual 2.6 Ghz CPU and 4 GB memory. Due to the nature of the calculations involved in the optimization pass, we believe that a GPU implementation might be engineered by sampling the research space.

5. Conclusion and Future Works

We present a novel method for human motion capture data compression exploiting both temporal and spatial coherence to achieve high compression ratios with few perceptual distortion. Our experiments show that the use of a compact pose model allows to successfully recover poses given only end-joints positions. As the end-joints and root joint's trajectories present high temporal coherence, they can also be compressed in order to further improve compression rates. A particularly appealing aspect of our technique is that the pose model may also be used for editing compressed motions by employing the very same algorithm.

Though the inverse kinematics algorithm we presented is able to run in real-time on a modern machine, the decompression times are still longer than for other motion capture data compression techniques. However, our implementation could still be improved. The compression technique used for end-joints trajectories could also be enhanced to better reconstruct sharp features, such as foot contacts. The use of a suitable wavelet compression could lead to better results. We also did not exploit the linear correlations present in the end-joint's positions: applying a compression technique similar to the one presented in [LM06] to these markers could even improve compression performances, either allowing to further reduce data size, or to increase the number of constraint joints in the IK. If more quality is required, quantization could also be employed to improve the reconstruction of joint's trajectories by compressing the errors with controllable size overhead.

Finally, our data-driven, PGA-based IK algorithm is very promising as it allows to perform *data-driven* inverse kinematics without resorting to computationally expensive learning, such as the one presented in [GMHP04]. Thus, it could find application in multiple areas such as recovering poses and motion from videos. The run-time optimization also seems to be faster, which could make it suitable for real-time applications.

Acknowledgements

Figure 5: *3 animations from the CMU database compressed using our technique*

References

[Ari06] ARIKAN O.: Compression of motion capture databases. *ACM Trans. Graph. 25*, 3 (2006), 890–897.

[BF01] BUSS S. R., FILLMORE J. P.: Spherical averages and applications to spherical splines and interpolation. *ACM Transactions on Graphics 20*, 2 (2001), 95.

[BPP07] BEAUDOIN P., POULIN P., PANNE M. V. D.: Adapting wavelet compression to human motion capture clips. In *GI '07: Proceedings of Graphics Interface 2007* (New York, NY, USA, 2007), ACM, pp. 313–318.

[FLJ03] FLETCHER P. T., LU C., JOSHI S. C.: Statistics of shape via principal geodesic analysis on lie groups. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2003 Proceedings CVPR-03* (2003), pp. I–95.

[FLPJ04] FLETCHER P. T., LU C., PIZER S. M., JOSHI S. C.: Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging 23*, 8 (2004), 995.

[GMHP04] GROCHOW K., MARTIN S. L., HERTZMANN A., POPOVIC Z.: Style-based inverse kinematics. In *ACM SIGGRAPH 2004 Papers on - SIGGRAPH 04 SIGGRAPH 04* (New York, NY, USA, 2004), ACM Press, p. 522.

[Gra98] GRASSIA F. S.: Practical parameterization of rotations using the exponential map. *journal of graphics tools 3*, 3 (1998), 29–48.

[JJD00] J. J. DUISTERMAAT J. A. C. K.: *Lie Groups*. Springer, 2000.

[KG04] KARNI Z., GOTSMAN C.: Compression of soft-body animation sequences, 2004.

[Len99] LENGYEL J. E.: Compression of time-dependent geometry. In *I3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics* (New York, NY, USA, 1999), ACM, pp. 89–95.

[LM06] LIU G., MCMILLAN L.: Segment-based human motion compression. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2006), Eurographics Association, pp. 127–135.

[LS01] LEE J., SHIN S. Y.: A coordinate-invariant approach to multiresolution motion analysis. *Graphical Models 63*, 2 (2001), 87.

[Moa02] MOAKHER M.: Means and averaging in the group of rotations. *SIAM J. Matrix Anal. Appl. 24*, 1 (2002), 1–16.

[MSZ94] MURRAY R. M., SASTRY S. S., ZEXIANG L.: *A Math-*

ematical Introduction to Robotic Manipulation. CRC Press, Inc., Boca Raton, FL, USA, 1994.

[Pen06]   PENNEC X.: Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision 25*, 1 (2006), 127.

[RDS*05]   RAHMAN I. U., DRORI I., STODDEN V. C., DONOHO D. L., SCHRODER P.: Multiscale representations for manifold-valued data. *Multiscale Modeling & Simulation 4*, 4 (2005), 1201–1232.

[RP03]   REITSMA P. S. A., POLLARD N. S.: Perceptual metrics for character animation. In *ACM SIGGRAPH 2003 Papers on - SIGGRAPH 03 SIGGRAPH 03* (New York, NY, USA, 2003), ACM Press, p. 537.

[RPE*05]   REN L., PATRICK A., EFROS A. A., HODGINS J. K., REHG J. M.: A data-driven approach to quantifying natural human motion. In *ACM SIGGRAPH 2005 Papers on - SIGGRAPH 05 SIGGRAPH 05* (New York, NY, USA, 2005), ACM Press, p. 1090.

[SCLS07]   SAID S., COURTY N., LEBIHAN N., SANGWINE S.: Exact principal geodesic analysis for data on so(3). In *Proc. of the 15th European Signal Processing Conference (EUSIPCO 2007)* (Poznan, Poland, September 2007).

[Sho85]   SHOEMAKE K.: Animating rotation with quaternion curves. In *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1985), ACM, pp. 245–254.

[SHP04]   SAFONOVA A., HODGINS J. K., POLLARD N. S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. In *ACM SIGGRAPH 2004 Papers on - SIGGRAPH 04 SIGGRAPH 04* (New York, NY, USA, 2004), ACM Press, p. 514.

[SSK05]   SATTLER M., SARLETTE R., KLEIN R.: Simple and efficient compression of animation sequences. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation - SCA 05 SCA 05* (New York, NY, USA, 2005), ACM Press, p. 209.

[Swe98]   SWELDENS W.: The lifting scheme: a construction of second generation wavelets. *SIAM J. Math. Anal. 29*, 2 (1998), 511–546.