DOI: 10.1111/cgf.12326
EUROGRAPHICS 2014 / B. Lévy and J. Kautz
(Guest Editors)
Volume 33 (2014), Number 2

Rate-distortion optimized compression of motion capture data

L. Váša, G. Brunnett

Technical University Chemnitz, Chemnitz, Germany

Abstract

Lossy compression of motion capture data can alleviate the problems of efficient storage and transmission by exploiting the redundancy and the superfluous precision of the data. When considering the acceptable amount of distortion, perceptual issues have to be taken into account. Current state of the art methods reduce the data rate required for high quality storage of motion capture data using various techniques. Most of them, however, do not use the common tools of general data compression, such as the method of Lagrange multipliers, and thus they obtain sub-optimal results, making it difficult to do a fair comparison of their performance.

In this paper, we present a general preprocessing step based on Lagrange multipliers, which allows to rigorously adjust the precision in each of the degrees of freedom of the input data according to the amount of influence the given degree of freedom has on the overall distortion. We then present a simple compression method based on Principal Component Analysis, which in combination with the proposed preprocessing achieves significantly better results than current state of the art methods. It allows optimization with respect to various distortion metrics, and we discuss the choice of the metric in two common but distinct scenarios, proposing a perceptually oriented comparison metric based on the relation of the problem at hand to the problem of compression of dynamic meshes.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism—Animation Image Processing and Computer Vision [I.4.2]: Compression (Coding)—Approximate methods;

1. Introduction

Motion capture (*mocap*) data are routinely used in a number of CG applications, such as computer gaming or in film industry. Current industrial motion capture systems are mature and allow precise acquisition of the high number of degrees of freedom (DOFs) associated with a humanoid figure motion. In current practice, data of 40-120 DOFs are usually used, which in combination with high sampling rates (commonly 120Hz) results in high storage and transmission costs. As the capabilities of motion capture systems improve, it is expected that these costs will even increase in the future.

Despite the recent improvements, the accuracy of motion capture systems remains limited, and for practical purposes, it is not even necessary to improve it beyond the limits of precision perceivable by a human observer, unless a high precision is required for tasks such as computerized ergonomic evaluation. A compression method that ignores the superfluous precision is essentially lossy, although lossless from the perception point of view. Additionally, a large subset of applications allow for a perceivable alteration of the data, as long as the perceptual quality and semantic meaning of the data is preserved. These observations can be exploited to alleviate the problem of efficient transmission and storage of mocap data by applying a lossy compression.

Current methods of lossy compression of mocap data already use the concepts sketched above. Various sophisticated approaches are used to express the data using a low number of bits. However, particularly the problem of preserving the perceptual quality of the data remains an open issue, mainly because of the lack of a perception correlated distortion metric. Several metrics have been proposed, focusing on different aspects of human perception, however a general consensus on a proper metric has not been reached yet. Additionally, researchers have proposed a number of compression techniques, some of which are mocap data specific, while others are more general (for example using a different entropy coding), and it is thus difficult to comprehend the relative efficiency of the approaches proposed so far.

In this paper, we propose a simple lossy compression method based on Principal Component Analysis (PCA). Our

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

method employs an approach that is simpler than some of the recently proposed ones; however, by including general compression tools, such as entropy coding of residuals and the method of Lagrange multipliers for rate-distortion optimization, we achieve results that are much better than the ones reported by the current state of the art methods. By publishing the implementation of the method, we hope to provide a solid baseline method for future research in this field.

The concepts used in our method allow for optimization against various error metrics. We present the results optimized for minimal distortion of joint positions (which is a non-perceptual metric used by many papers on mocap data compression) and on a perceptual metric derived from the Spatio-Temporal Edge Difference (STED) [VS11] perceptual metric, which has been recently proposed for the purposes of evaluating distortion in dynamic meshes.

We have compared the performance of our method against the state of the art approaches. Our approach provides distortions that are about 50% lower than the ones provided by the state of the art, at the same data rates.

The rest of the paper is organized as follows: In section 2 we give a brief overview of related research. In section 3 we describe our compression scheme based on PCA and Lagrange multipliers. Section 4 provides details on the two distortion metrics used, of which one is being newly proposed. Results and comparison against the state of the art are shown in section 5, and finally conclusions are drawn in section 6.

2. Related work

The problem of compression of motion capture data became a major focus point of researchers since roughly 2005, mainly because of fast evolution of mocap capture technology at that time, and its ever growing applications in games and film industry since.

One of the first papers on mocap data compression has been published by Arikan [Ari06]. The method is based on representing the data in the form of virtual markers, one for each joint. The method than approximates the trajectories of these markers by splines, which ensures that the decompressed data are going to be smooth, and thus perceptually more natural. Arikan compares his method, among others, against a baseline method based on PCA. The discussion, however, seems insufficient, since he does not discuss key issues, such as quantization of PCA basis and coefficients, focusing solely on the number of preserved degrees of freedom.

A PCA method has been proposed by Liu and McMillan [LM06]. Their approach is based on motion segmentation and interpolation of PCA coefficients using temporal splines.

The work by Chattopadhyay et al. [CBL07] identifies the need for optimization of parameters that influence the compression. In their indexing based algorithm, they resort to exhaustive search for finding the optimal configuration.

Wavelet based compression has been proposed by Beaudoin et al. [BPvdP07]. The key idea of their algorithm is to adjust the compression parameters to the relative importances of the separate DOFs. In their approach, the compression parameters are set using a stochastic process similar to simulated annealing. In contrast to that, we evaluate the relative importances of the DOFs directly.

A method based on dimensionality reduction using Principal Geodesics Analysis (PGA) has been presented by Tournier et al. [TWC*09]. His method works directly in the space of angles, and the space is described in terms of principal geodesics. The positions of end effectors are encoded directly, and the remaining DOFs are optimized using an IK-like solver in the space of principal geodesics.

Temporal redundancy in the data, i.e. repeating motion, is explicitly exploited by some of the algorithms, such as the one proposed by Beaudoin et al. [BCvdPP08]. Recently, Lin et al. [LPLT11] has proposed an algorithm based on repeated motion analysis, PCA and representation of data by Catmull-Rom splines. Their method currently provides the best rate-distortion ratios, at the cost of a rather complex implementation.

Perceptual issues have been brought to attention again in the work of Firouzmanesh et al. [FCB11]. Their method is based on wavelet decomposition, and they have performed a series of experiments with users evaluating the quality of the decompressed meshes. Unfortunately, these experiments only involved skeletal motions without skin mesh (stick figures), i.e. no realistic conditions were used. The data from their experiments are moreover not publicly available, and thus they cannot be used to construct a perception correlated metric.

In absence of a perceptually correlated distortion metric, most researchers use some derivative of mean squared error applied onto joint positions, such as the KG error [KG04]. At the same time, since the pioneering work of Arikan, researchers agree that such measure does not capture well the amount of perceived distortion. Nevertheless, no perception correlated metric has been proposed so far, and the few subjective experiments performed so far were conducted with the purpose of evaluating one particular compression method.

Visible artifacts are among others often identified around the so-called "contacts", i.e. in areas where the figure touches some static object. In such areas, the data are usually not smooth, and since many of the compression methods essentially remove high frequencies, it is possible that the data become visibly distorted in such areas. One particularly well known problem of this kind is the foot skating, i.e. movement of feet that are in fact supposed to be in contact with the ground and thus static. Some compression algorithms attempt to address this issue explicitly by identifying such contacts and sending additional information. On the other hand, the problem of foot skating removal has been studied

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.

also from the point of view of motion retargeting and motion synthesis [ZRB10, AF02], and it has been addressed by specialised algorithms, such as [IAF06] or [PHO11]. In our proposed algorithm, we do not resolve these issues explicitly, and leave it to the user to choose a relevant foot-skate removal procedure.

The amount of distortion caused by compression of mocap data in fact manifests itself as distortion of a sequence of meshes which results from using the mocap data to animate a rigged model. From this point of view, the evaluation of distortion of mocap data is closely related to evaluation of distortion in animated (dynamic) meshes, which has been recently addressed by the STED algorithm by Váša and Skala [VS11].

There are also other links to dynamic mesh compression. Some mocap compression algorithms can be interpreted in terms of compressing trajectories of vertices, which corresponds to the Coddyac algorithm [VS07] proposed for compression of dynamic meshes [MZP06]. The problem of compressing basis of unevenly important basis trajectories (principal components of the space of trajectories) also appears in both fields [HYKL06]. This issue has been efficiently addressed by the COBRA algorithm [VS09], which can be in fact interpreted in terms of the method of Lagrange multipliers, which is used extensively in our proposal.

Most of the current state of the art algorithms use the general tools of data compression, such as quantization and entropy coding. Unfortunately, most of the time, the application of these methods seems to be incomplete, which makes it difficult to compare the results: sometimes, it is hard to asses, whether the improved performance stems from the actual improved compression strategy, or from using a more efficient entropy coder. Moreover, most of the state of the art approaches do not employ rate-distortion optimization, and the authors of the respective papers only report the results for one particular level of distortion and one particular data rate, instead of producing a full rate-distortion curve.

Our algorithm is easy to implement, yet its efficiency surpasses that of current state of the art. Our aim is to create a proper reference method for future efforts, against which future algorithms can be compared.

3. Method

The mocap data are usually represented by a *skeleton structure* and *frame data*. The skeleton structure defines the hierarchy of bones, their rest position and the degrees of freedom they have. The degrees of freedom are the values that change in each frame. These are usually joint rotations (up to three Euler angles are common) and joint translations, in the case when bone lengths change in time (usually they do not, in such case only the position of the root bone is specified in the form of joint translation). The frame data then specify all the degrees of freedom for each frame of the animation, which represents the largest part of the data representation.

Lossy compression encodes the input data into a small number of bits, from which an approximation of the input can be reconstructed. The reconstruction should be as close to the original as possible. Therefore, in order to construct an efficient compression algorithm, it is necessary to define an *error metric*, which evaluates the distortion of the reconstruction with respect to the input. Having such metric, it is possible to evaluate the performance of the compression algorithm in the form of a rate-distortion chart. Such chart then also allows to compare algorithms; however, the comparison always depends on the choice of the distortion metric.

The interpretation of the data at hand implies, that the separate degrees of freedom have a widely varying impact on the resulting overall distortion. A simple example may be an Euler angle of a bone that is located at the beginning of the kinematic chain. A local distortion of such an angle is reflected by distortion of positions of all joints that descend from that bone, which leads to a large global error (if the distortion metric in question measures discrepancies in joint positions). On the other hand, an angle at the end of the kinematic chain only affects one joint, and thus its distortion by the same amount causes a much lower global distortion. Our method takes this observation into account by using the Lagrange equalization as a first step. This equalization first analyses the influence of local distortion in each of the DOFs with respect to a particular target metric, and then it **scales** each DoF so that the global distortion is minimized. Note that, apart from different scaling selected by the equalization procedure, all the DOFs are treated equally in our algorithm, including the data associated with root of the kinematic chain.

Subsequently, we apply a sequence of mocap specific steps - we apply PCA to find a subspace of common poses in the input data, and then in a separate orthogonal PCA we decorrelate the DOFs in small chunks of frames. PCA has already been discussed as a baseline method for compression of mocap data in [Ari06], but only a very shallow discussion was given to the aspects of encoding using this tool. In our approach, we employ an advanced strategy for encoding of basis vectors, which has been proposed for the purpose of dynamic mesh compression. Finally, we encode the PCA coefficients using an entropy coder. The following subsections give details on each of these steps.

3.1. Lagrange multiplier

As mentioned above, distortion to each of the DOFs causes a different overall distortion, making some of the DOFs more important than others. Using the same precision for each of the DOFs is therefore suboptimal. Intuitively one should use a higher precision for more important DOFs, while a lower precision might suffice for DOFs that have a smaller impact

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.

on the overall error. The question at hand therefore is, how does one steer the precision in the separate DoFs, so that the performance is globally close to an optimum in the rate-distortion sense.

This issue appears in many data compression applications, and it usually boils down to following constrained optimization problem: In a space of parameters $q_1, q_2, ..., q_n$ (each influencing the precision in one of the DoFs), there are two functions defined by the compression algorithm: $R(q_1, q_2, ..., q_n)$ is the data rate for the given parameters, and $D(q_1, q_2, ..., q_n)$ is the resulting distortion. The task is to minimize $D(q_1, q_2, ..., q_n)$ under the condition that

$$R(q_1, q_2, ..., q_n) = r_t, \tag{1}$$

for some given target rate r_t. The method of Lagrange multiplier yields a condition for the optimum $grad(D) = \lambda grad(R)$, which can be rewritten as

$$\frac{\frac{\partial D}{\partial q_1}}{\frac{\partial R}{\partial q_1}} = \frac{\frac{\partial D}{\partial q_2}}{\frac{\partial R}{\partial q_2}} = ... = \frac{\frac{\partial D}{\partial q_n}}{\frac{\partial R}{\partial q_n}} = \lambda, \tag{2}$$

which is also known as the *principle of equal slopes* [PV10]. Note that the intuitive justification of this condition is, that if there was a different slope in a pair of parameters q_i and q_j, then it would be possible to reach a configuration that has the same rate and smaller distortion by increasing one of these parameters and decreasing the other.

In our scenario, we have a rather complex compression algorithm down the pipeline, however, with some simplifying assumptions, we can improve the performance considerably. Since the subsequent algorithm treats each of the DOFs equally, we can influence the amount of error introduced (together with the resulting data rate) by selecting a **weight** w_i for each of the DOFs. The values of the i-th DOF will multiplied by w_i before the compression, and this step is reverted by multiplying the values by w_i^{-1} after the decompression. The weights w_i now play the role of parameters q_i in the constrained optimization.

In order to evaluate the influence of the weights w_i on the data rate, we model the subsequent lossy compression of the input values as truncating the data in binary representation at a certain fixed position. Therefore, for example, multiplying the data by 2 adds one additional bit that does not get truncated, and therefore the data rate increases by one bit per value. In general, the increase is $log_2(w_i)$ bits per value. Since each DoF is represented by the same number of values, we can approximate the data rate by $R = R_0 + b\sum_{i=1}^{n} log_2 w_i$, where R_0 and b are some unknown constants. Note that this estimation is only important for estimating the gradient, which is:

$$grad(R) = b(1/w_1, 1/w_2, ..., 1/w_n). \tag{3}$$

In order to find the weights w_i, one also has to estimate gradient of D. The partial derivatives of course strongly depend on the used compression algorithm, and on the character of error measure being used. We therefore propose following numerical procedure for each DoF:

1. Add uniform noise of (small) standard deviation σ_s to the i-th DOF of the original data (this simulates decreasing the precision in the i-th DOF). We use 1/100 of the standard deviation of each DoF as σ_s.
2. Evaluate the distortion value D_i of the data with introduced noise, using the target error measure.

Note that for the estimation, noise is always introduced to a single DoF at the time. This allows us to estimate the partial derivatives $\partial D/\partial \sigma_i = D_i/\sigma_s$. Since the data are first multiplied by w_i, then a distortion, i.e. noise of some unknown (yet constant) standard deviation k is introduced, and finally, during the decompression, the data are divided by w_i, *together with the noise*, we can estimate the standard deviation of the resulting noise as $\sigma_i = k/w_i$, where k is some unknown constant. Therefore we get

$$\frac{\partial D}{\partial w_i} = \frac{\partial D}{\partial \sigma_i}\frac{\partial \sigma i}{\partial w_i} = -\frac{D_i}{\sigma_s}\frac{k}{w_i^2}. \tag{4}$$

This procedure yields an estimate of $grad(D) = -k/\sigma_s(D_1/w_1^2, D_2/w_2^2, ..., D_n/w_n^2)$. Together with equation (3) and the condition of optimality, putting all the constants into one, we get $(D_1/w_1^2, D_2/w_2^2, ..., D_n/w_n^2) = \lambda(b/w_1, b/w_2, ..., b/w_n)$, i.e.

$$w_1 = \frac{D_1}{\lambda}, w_2 = \frac{D_2}{\lambda}, ..., w_n = \frac{D_n}{\lambda}. \tag{5}$$

Intuitively, D_i represents how sensitive is the error measure to changes in i-th DOF. A higher sensitivity of a particular DOF naturally leads to a higher required w_i, which in turn reduces the amount of error introduced to that DOF. This equation allows us to select an optimized weight for each DOF, based only on a single parameter λ (the Lagrange multiplier), which plays the role of a global quantization parameter, i.e. it allows controlling the amount of loss and thus the size of the resulting file. In the following text, we will refer to this procedure as *Lagrangian equalization*. Adjusting the algorithm to a different error measure then results in changing step 3 in the estimation of $\partial D/\partial w_i$.

The presented procedure is based on multiple estimations that may not be accurate for all compression algorithms and/or error measures. However, we have achieved a considerable performance improvement (detailed in section 5) for two error measures that will be presented in section 4, using the compression algorithm described in the following section.

3.2. Pose space PCA

Having F frames and N degrees of freedom (each of which has been multiplied by the weight w_i determined by the Lagrangian equalization), the data can be represented by F vectors $\mathbf{v}_i, i = 1..F$ of length N, each vector representing the skeleton pose in one frame of the sequence. The character

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.

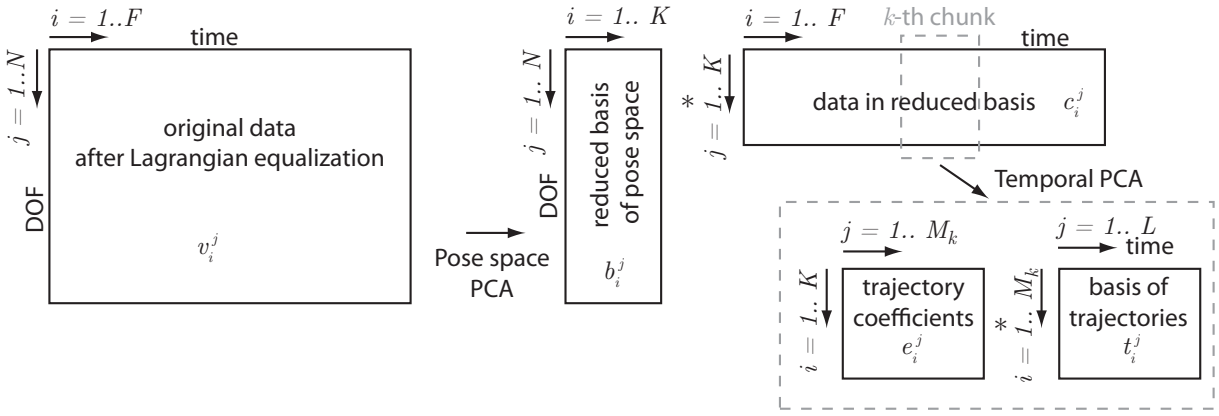

Figure 1: *Schematic of the compression steps. Black rectangles represent matrices, the asterisk character (*) represents matrix multiplication. The means vectors are omitted from the figure for clarity reasons.*

of the data implies that the vectors are not randomly and independently distributed in the N-dimensional space: some poses are likely, others unlikely, and also some poses are combinations of others. This suggests that there exists a linear subspace (of dimension lower than N) of the full space of poses that contains almost all the poses in the dataset. PCA gives a simple end efficient way to identify the basis of this space by eigenvalue decomposition of the autocorrelation matrix.

We use all the frames of the input sequence to form the autocorrelation matrix, and therefore the whole sequence must be known in advance to the encoding algorithm. On-the-fly encoding is therefore only possible using some buffer, which processes the input data in batches. Note that this is a property shared by most state of the art algorithms that focus on rate-distortion efficiency, and in many applications this poses no problem.

The PCA process yields a set of orthonormal basis vectors (in fact poses) $\mathbf{b}_i, i = 1..N$, such that each original vector $\mathbf{v}_i$ can be expressed as

$$\mathbf{v}_i = \overline{\mathbf{v}} + \sum_{j=1}^{N} \mathbf{b}_j c_i^j, \tag{6}$$

where $\overline{\mathbf{v}}$ stands for average of the input vectors $\mathbf{v}_i$. Apart from being globally independent, the coefficient vectors $\mathbf{c}_i$ can also be ordered in order of the variance of the data in the direction of the corresponding basis vector $\mathbf{b}_i$. Subsequently, one can omit the coefficients that correspond to basis vectors of low variance (the coefficients are going to be close to zero anyway), preserving only K coefficients for each frame. Having a rectangular matrix $B = (\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_K)^T$, one can compute the coefficients for each frame by

$$\mathbf{c}_i = B(\mathbf{v}_i - \overline{\mathbf{v}}). \tag{7}$$

This process thus reduces the dimension of the data describing each frame from N to K, at the cost of omitting the dimensions in which there is the lowest variation. The coefficient vectors $\mathbf{c}_i, i = 1..F$, each of length K, are then passed as input to the second step of the compression.

3.3. Temporal PCA

After the previous step, although projected onto a smaller subspace, the vectors $\mathbf{c}_i$ can still be interpreted as degrees of freedom in i-th frame. The character of the data implies, that in some parts of the sequences, there might be a strong correlation between some of the DOFs (angles correlate with each other), but this correlation may be different in different parts of the sequence (arm movement is strongly correlated to leg movement when walking, however they are not correlated at all for some other actions).

These correlations should be exploited by the compression algorithm, so that the common part of the information is not being transmitted multiple times with each of the DOFs. In order to address the changes in correlation throughout the sequence, we first divide the sequence into smaller chunks of length L (in our implementation L is a user defined parameter of default value 120 frames, which usually corresponds to one second of the data). The correlation is then going to be analysed and exploited in each chunk separately. Although a small C^0 discontinuity may appear at chunk transitions, we did not observe any such artifacts in our experiments.

PCA offers itself again as an efficient means of removing correlation, only this time we apply it orthogonally, i.e. samples now correspond to temporal development of DOFs in time within the given chunk. We are now working with vectors $\mathbf{c}^j = (c_1^j, c_2^j, ..., c_L^j), j = 1..K$. PCA finds a separate decorrelated basis for each chunk, and the DOFs are expressed in this new basis. In this process, the basis vectors

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.

$\mathbf{t}_i$ of length L can be interpreted as temporal 1-dimensional trajectories, and the temporal development of each DOF is expressed as a linear combination of these trajectories:

$$\mathbf{c}^j = \bar{\mathbf{c}} + \sum_{i=1}^{L} \mathbf{t}_i e_i^j, \tag{8}$$

where $\bar{\mathbf{c}}$ is the average of vectors $\mathbf{c}^j$ in the given chunk.

In order to describe the data, one has to store the combination coefficients e_i^j for each of the DOFs, and a given number of basis trajectories $\mathbf{t}_i, i = 1..M_k$ for each of the chunks. Note that the number of preserved basis vectors is chosen differently for each of the chunks, because the motion complexity can vary from chunk to chunk. In our implementation, the user defines the maximum error caused by this dimensionality reduction, and the largest possible number of neglected basis trajectories is selected so that this error is not surpassed (error caused by removing a basis trajectory is estimated as sum of magnitudes of coefficients associated with the given basis vector). Figure 1 shows a schematic of the algorithm.

3.4. Encoding of bases

After the transformations, it is necessary to store the coefficients $\mathbf{e}_i$ and basis vectors $\mathbf{t}_i$ for each of the chunks, and the basis vectors $\mathbf{b}_i$, which represent the global basis of the space of poses. It is of course not necessary to store these numbers in full precision, thus quantization takes place. The coefficients $\mathbf{e}_i$ are quantized uniformly and encoded using an entropy coder [MSB*03], while a more elaborate quantization strategy should be taken for the basis vectors. The character of vectors $\mathbf{t}_i$ is very similar to the character of basis trajectories used in dynamic mesh compression, and thus they are well suited for compression using the COBRA algorithm [VS09].

The key ideas of the COBRA algorithm can be again expressed in terms of Lagrange multipliers - the basis vectors are not equally important, and thus their quantization should be adjusted according to their importance. Moreover, the basis vectors can be interpreted as temporal (1D) trajectories, and therefore temporal predictors can be used to encode them efficiently. In our implementation, we have added a temporal subsampling step, which allows achieving a smooth reconstruction even from very low bitrates. The procedure is very simple, we preserve only every s-th sample in each basis vector, where s is a user selected parameter, which can be chosen rather high for 120-Hz datasets. In the reconstruction, the missing samples are simply interpolated from the four surrounding known samples (border values are repeated where necessary), using a cubic polynomial interpolant.

For details on the COBRA algorithm we refer the reader to the original paper. COBRA can be also applied to the basis vectors $\mathbf{b}_i$, however, this time the temporal interpretation is no longer possible and thus this step is omitted.

3.5. Algorithm configuration

There are several parameters of the algorithm, such as

- K, the number of preserved basis poses,
- P, the parameter influencing the number of preserved basis trajectories in each chunk, expressed as a fraction of data variance to be maximally removed by dimensionality reduction,
- Q_l and Q_g, the quantization parameters for local and global PCA bases,
- λ the Lagrange multiplier which plays the role of quantization parameter for the PCA coefficients,
- s the stride in subsampling of local bases.

The overall performance depends on the interplay of these parameters. If the optimal compression performance is required, then one can use a procedure similar to the one presented in [PV10] in order to find the optimal configuration. Optimizing the parameters allows reaching virtually any desired accuracy and bitrate, and thus we use it in our experiments to produce results at bitrates comparable to the ones achieved by competing algorithms.

The procedure requires running the compression algorithm several times in order to optimize the configuration, leading to asymmetric processing times, where compression is significantly slower than decompression. Such asymmetry is acceptable in scenarios where content is being prepared for final delivery, similarly to video coding. On the other hand, if the data are to be stored for further processing, a default configuration may be used, which may lead to slightly suboptimal results in much shorter time. We have selected the form of parameters in our implementation so that they are as data-independent as possible. The default configuration is part of our reference implementation of the algorithm.

4. Distortion evaluation

Lossy compression algorithms are always evaluated with respect to a particular error measure, and ideally the algorithm should be configurable to provide optimal results with respect to an error metric of choice. This adjustability is in our algorithm provided by the Lagrangian equalization step, which estimates the influence of each DOF on the overall distortion.

Similarly to image, video and audio compression, in compression of mocap data it is possible to identify two main scenarios where lossy encoding is used: First, for storage of *raw* data intended for further processing, and second, for storage of *final* product, intended for viewing by the final consumer. These two scenarios have specific requirements on the error measure used. In the *raw* storage, users usually prefer encoding that has some guaranteed precision, usually related to the accuracy of the input device. Perceptual issues are usually not considered, because processing steps further down the pipeline might alter the perceivability of artifacts.

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.

Although high accuracy is usually required, some precision loss is usually allowed. Simple mathematical error measures, such as Mean Error (ME), usually serve well in this scenario.

The second scenario, in which *final* product is being encoded for delivery to the consumer, has different requirements. The required precision is usually not specified in terms of maximum spatial discrepancy, but in terms of perceivability: any distortion that is not perceivable by a human observer is usually allowed, and sometimes also some perceivable, but not perceptually disturbing changes are allowed as well. It is well known in the compression community, that measuring the amount of perceived error is a complex task which involves subjective experiments and either modelling of human visual system, or attempting to mimic the results of perception by a black box metric. At this point, there is still no consensus on how to measure perceived error caused by mocap data compression; however, it has been conclusively demonstrated that metrics such as ME are not suited for such task. In the following, we discuss the possibilities for both *raw* and *final* scenarios.

4.1. Mean Error

ME is a metric used by many current mocap data compression algorithms. It builds on the notion of representing the mocap data by a skeleton structure consisting of joints, which are connected with bones. It is straightforward to convert the **original** and **distorted** mocap data into a set of joint positions, $\mathbf{p}_i^j, i = 1..F, j = 1..J$ and $\widehat{\mathbf{p}}_i^j, i = 1..F, j = 1..J$ respectively, where J is the number of joints in the skeleton. ME is then simply the mean distance between the original and distorted positions:

$$ME = \frac{1}{FJ} \sum_{i=1}^{F} \sum_{j=1}^{J} \sqrt{(\mathbf{p}_i^j - \widehat{\mathbf{p}}_i^j)^T (\mathbf{p}_i^j - \widehat{\mathbf{p}}_i^j)} \qquad (9)$$

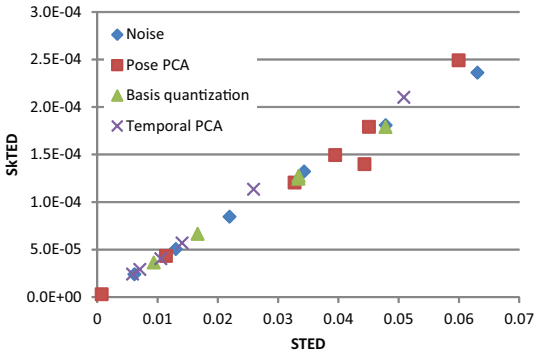

Figure 2: *Correlation of the STED and SkTED on the dancing man dataset.*

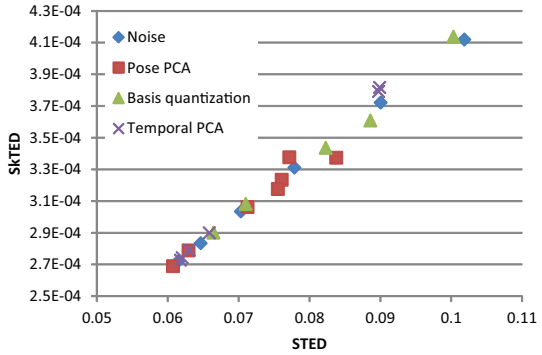

Figure 3: *Correlation of the STED and SkTED on the walking dataset.*

4.2. Perceptual

The inadequacy of ME as measure of perceived distortion has been already identified in the seminal work on mocap data compression by Arikan [Ari06]. However, apart from stating that ME correlates with perception poorly, only few attempts were made to propose a better measure. The question of proper way to measure distortion in the *final* scenario is still open, and new measures are expected in the future.

Subjective experiments have been recently performed in the context of mocap data compression by Firouzmanesh et al. [FCB11]. In these experiments, animations of stick figures were shown to users, and the users evaluated the amount of perceived distortion with respect to a known original motion. In our view, the fundamental problem of these experiments is, that the users were evaluating the results in an artificial scenario, which does not occur in practice, where the mocap data are always used for *skinning animation of a surface mesh*. The users should therefore evaluate the distortion of the final animated mesh instead of a stick figure.

Recently a perceptual metric STED (Spatio-temporal Edge Difference [VS11]) has been proposed in the context of compression of dynamic meshes, i.e. sequences of static meshes, which represent a movement of a surface in time. This metric provides a much higher correlation with human perception, reaching Pearson correlation of more than 0.9. In a typical scenario, mocap dataset is applied to animate a mesh, using some set of animation weights, which can be for example computed by the bone glow algorithm [WL08]. The result is a sequence of meshes of shared connectivity, which exactly matches the input of the STED measure, thus it seems that this metric could provide the first step towards perceptual evaluation of mocap data distortion.

The STED metric consists of two terms, one focusing on spatial distortion, and the other on temporal distortion. The key concept behind the spatial term of the metric is that humans perceive local, relative changes of vertex positions

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.

rather than the dislocation from original coordinates. Therefore, the spatial term is computed from changes in edge lengths rather than changes in absolute vertex positions.

The second term of the metric focuses on the so-called *temporal artifacts*, which may appear in dynamic meshes and cannot be detected by considering each frame separately. Typically, smooth dislocation of all vertices is hard to perceive, however, if the dislocation appears in different directions in subsequent frames, the users can clearly perceive the "shaking" of the surface. This effect is particularly well visible in parts of the mesh that are static, or move slowly. In order to capture these artifacts, it is therefore necessary to take the vertex speed into account as well. The temporal term of STED is computed in a way similar to the spatial term, only this time considering virtual, temporal edges, which connect each vertex position in two subsequent frames (in a sense, these edges represent the vector speed of vertices). The difference of length of these edges is computed and normalized using the inverse original speed of the vertex. The sum of these edge length changes is used as the temporal term. The overall error is computed as hypotenuse of the spatial and temporal term. For more details on STED, see the original paper [VS11].

The STED error measure can be applied directly, however, there are some mocap data specific considerations, which may simplify the computations considerably. First, the result depends on the mesh that is used for skinning, however, one might want a metric that is mesh independent. Also, the spatial term is probably going to be quite small in relation to the temporal term, given that the animation weights are smooth and the only source of distortion is the compression of mocap data. We therefore propose a simplified version of the STED error, denoted SkTED (Skeletal Temporal Edge Difference) which is only based on the temporal term of the STED error, and which is computed *on the joint positions only*, thus it is mesh independent.

The SkTED measure is evaluated as follows: having the total number of joints J and total number of frames F, we convert the **original** and **distorted** mocap data into joint positions $\mathbf{p}_i^f, i = 1..J, f = 1..F$, and $\widehat{\mathbf{p}}_i^j, i = 1..J, j = 1..F$ respectively. Next, the temporal edge lengths are computed:

$$tel_i^f = \sqrt{(\mathbf{p}_i^f - \mathbf{p}_i^{f+1})^T(\mathbf{p}_i^f - \mathbf{p}_i^{f+1}) + d^2}, \qquad (10)$$

where d is a constant representing the temporal distance between frames, which is used to avoid division by zero later in the algorithm for the case of static vertices. Similarly, $\widehat{tel}_i^j$ is evaluated for the distorted joint positions. Vertex speed s_v^f, at a given frame f, is computed as average of tel_v^n, where n is in a given range of frames around the frame f. From these values, *temporal edge difference* is computed as

$$ted_i^f = \frac{\|tel_i^f - \widehat{tel}_i^f\|}{s_i^f}. \qquad (11)$$

Finally, the SkTED is computed as average of ted_i^f over all frames and all vertices.

To justify the proposed simplification, we have performed a set of experiments, which demonstrate the correlation of the results of SkTED with the temporal part of STED applied on a mesh skinned according to the compressed mocap data. We have tested several kinds of distortion, including dimensionality reduction, quantization of basis vectors and quantization of coefficients, each at varying strength. We have evaluated both STED (on the resulting dynamic mesh) and SkTED (on the input mocap data) for each kind of distortion, and we have evaluated the correlation of the two.

We have experimented with different motions, using a linear blend skinning method, and the results shown in figures 2 and 3 show that there is indeed a strong correlation between temporal part of STED and the SkTED. The correlation is very close to linear, as documented by the Pearson correlation coefficient of >0.99 over all the experiments we have performed. Another big advantage of SkTED over STED is, that it is evaluated much faster - a speedup of about $200\times$ has been achieved with respect to STED applied on a skinned animation of 58k vertices.

At this point, we do not have any direct experiment results that would confirm the correlation of SkTED with perception, such experiments are beyond the scope of this work. On the other hand, the good correlation of the STED measure has been confirmed by a set of subjective experiments, and apart form the experiments with stick figures very little work has been done in this direction. From this point of view, SkTED may serve as an important first step in a search for a proper error measure, since even though it is evaluated on the skeleton structure only, its validity is related to experiments with full 3D surfaces that were performed by the authors of [VS11].

5. Results

We have tested our method on selected clips from the Carnegie Mellon motion database, and we have compared them with the current state of the art algorithms proposed in [LPLT11] and [TWC*09]. We have used both ME and SkTED error metrics to demonstrate the results.

The results in table 1 show that our algorithm provides an improvement of up to 86% with respect to the state of the art, in terms of reduction of distortion at the same file size. These results demonstrate, that the previous claims that PCA is not suited well for compression of mocap data were not justified. Table 2 shows the compression and decompression times for some of the datasets, demonstrating a processing speed of up to 3000 fps for encoding and 21000 fps for decoding. These times were measured on a desktop machine with Intel Core i7-920 CPU @ 2.67GHz.

The figures 4 and 5 show the rate-distortion diagrams

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.

Table 1: *Comparison against the state of the art, all the file sizes are in kB.*

	[LPLT11]		proposed		
dataset	size	ME	size	ME	impr.
15_04	76.9	1.86	63.2	0.70	62.3%
17_08	26.7	1.58	25.4	0.70	52.3%
17_10	16.5	2.57	17.2	0.99	61.5%
85_12	23.1	5.02	24.2	1.87	62.8%

	[LPLT11]		proposed		
dataset	size	SkTED	size	SkTED	impr.
15_04	76.9	0.38	74.2	0.14	62.6%
17_08	26.7	0.45	26.0	0.27	39.9%
17_10	16.5	1.01	16.9	0.74	26.6%
85_12	23.1	1.40	24.1	0.75	46.2%

	[TWC*09]		proposed		
dataset	size	ME	size	ME	impr.
15_04	127.8	2.12	124.4	0.29	86.4%
17_08	13.4	1.29	13.6	1.92	-49.0%
17_10	16.9	2.16	17.2	0.99	54.2%
85_12	14.3	6.26	14.8	3.71	40.7%

Table 2: *Preprocessing, encoding and decoding times*

	frame count	prep. [s]	enc. [s]	dec. [s]	enc. [fps]	dec. [fps]
15_04	22549	15.4	8.64	1.05	2609	21445
17_08	6179	4.34	2.14	0.35	2886	17878
17_10	2783	1.92	0.98	0.18	2820	15529
85_12	4499	3.15	1.51	0.28	2982	16247

in terms of ME and SkTED errors. The figure shows that the Lagrangian equalization (LE) is an essential step which boosts the performance of our simple coding scheme. Also the pose PCA (PPCA) and temporal PCA (TPCA) steps bring a substantial improvement of performance and thus cannot be omitted. The competing approaches are represented by single points, since they cannot be easily steered to different data rates and distortions (their parameters are fixed). Note that we are not showing the results of the [Ari06] algorithm, since they lie off the charts and are already largely outperformed by the state of the art. Finally, an example of the difference between decompressed skeletons at roughly comparable data rates is shown in figure 6.

We provide a reference implementation of our compression algorithm which can be used for benchmarking of future proposals. The implementation allows changing the error metric easily, so that the results can be optimized to existing metrics, as well as to metrics that are going to be proposed in the future. The implementation can be downloaded at http://www.tu-chemnitz.de/informatik/GDV/.

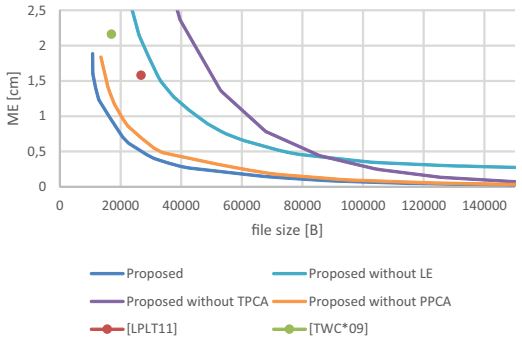

Figure 4: *Rate-distortion chart for the 17_10 dataset.*

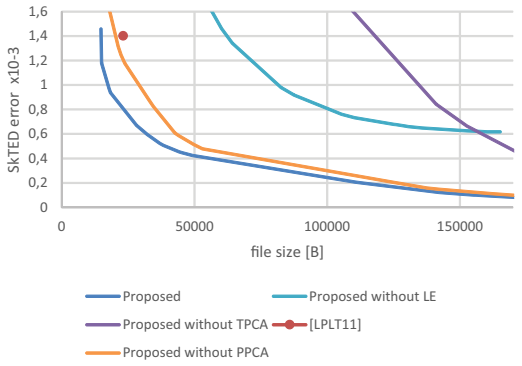

Figure 5: *Rate-distortion chart for the 85_12 dataset.*

6. Conclusions and future work

Our contribution to the field is threefold:

- We propose an automated tool for perceptual evaluation of distortion introduced to mocap data SkTED, based on the observed connection with dynamic meshes. Although the measure only works with the skeletal structure, its validity is supported by user studies with surface meshes.
- We propose a Lagrange equalization method, which allows adjusting the amount of error introduced to each of the DOFs.
- We propose a simple, fast and efficient algorithm based on PCA, which improves the performance of motion capture data compression well beyond the current state of the art.

Our algorithm is efficient and it can be configured to provide any desired accuracy.

At current stage, our algorithm outperforms the state of the art algorithms, which are based on different general methods. In most of the competing methods, it is possible to include Lagrangian optimization using a procedure similar to the one described here, possibly obtaining an improvement of the results. Whether or not including Lagrangian

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.

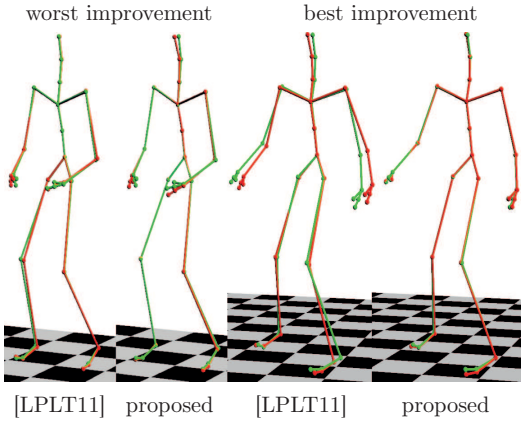

Figure 6: *Comparison of compression results. The original data are shown in green, decompressed in red, on two frames from the 17_08 sequence, the one where the proposed algorithm provided the worst (left) and best (right) improvement with respect to [LPLT11].*

optimization into these methods improves their performance over our proposed approach is a question for future research.

Another interesting question is the practical validation of our SkTED error measure by a subjective experiment. Such experiment should include results of various mocap data processing methods, and it probably should also take into account varying meshes that can be used with the same mocap data. We plan to design such experiment in the future.

Acknowledgements

Motion data from the CMU Graphics Lab Motion Capture Database were used in this paper. The authors would like to thank I-Chen Lin for providing the results of the Repeated Motion Analysis method.

References

[AF02] ARIKAN O., FORSYTH D. A.: Interactive motion generation from examples. *ACM Trans. Graph. 21*, 3 (July 2002), 483–490. 3

[Ari06] ARIKAN O.: Compression of motion capture databases. *ACM Trans. Graph. 25*, 3 (2006), 890–897. 2, 3, 7, 9

[BCvdPP08] BEAUDOIN P., COROS S., VAN DE PANNE M., POULIN P.: Motion-motif graphs. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2008), SCA '08, Eurographics Association, pp. 117–126. 2

[BPvdP07] BEAUDOIN P., POULIN P., VAN DE PANNE M.: Adapting wavelet compression to human motion capture clips. In *Proceedings of Graphics Interface 2007* (New York, NY, USA, 2007), GI '07, ACM, pp. 313–318. 2

[CBL07] CHATTOPADHYAY S., BHANDARKAR S. M., LI K.: Human motion capture data compression by model-based indexing: A power aware approach. *IEEE Transactions on Visualization and Computer Graphics 13*, 1 (Jan. 2007), 5–14. 2

[FCB11] FIROUZMANESH A., CHENG I., BASU A.: Perceptually guided fast compression of 3-d motion capture data. *IEEE Transactions on Multimedia 13*, 4 (2011), 829–834. 2, 7

[HYKL06] HEU J., YANG J.-H., KIM C.-S., LEE S.-U.: Effective quantisation scheme for principal components of 3-d mesh sequences. *IEE Electronics Letters 42*, 14 (July 2006), 799–800. 3

[IAF06] IKEMOTO L., ARIKAN O., FORSYTH D.: Knowing when to put your foot down. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2006), I3D '06, ACM, pp. 49–53. 3

[KG04] KARNI Z., GOTSMAN C.: Compression of soft-body animation sequences. In *Computers & Graphics* (2004), vol. 28, pp. 25–34. 2

[LM06] LIU G., MCMILLAN L.: Segment-based human motion compression. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2006), SCA '06, pp. 127–135. 2

[LPLT11] LIN I.-C., PENG J.-Y., LIN C.-C., TSAI M.-H.: Adaptive motion data representation with repeated motion analysis. *IEEE Transactions on Visualization and Computer Graphics 17*, 4 (2011), 527–538. 2, 8, 9, 10

[MSB*03] MARPE D., SCHWARZ H., BLÄDTTERMANN G., HEISING G., WIEG T.: Context-based adaptive binary arithmetic coding in the h.264/avc video compression standard. *IEEE Trans. Circuits Syst. Video Techn. 13* (2003), 620–636. 6

[MZP06] MAMOU K., ZAHARIA T. B., PRÊTEUX F. J.: A skinning approach for dynamic 3d mesh compression. *Journal of Visualization and Computer Animation 17*, 3-4 (2006), 337–346. 3

[PHO11] PRAŽÁK M., HOYET L., O'SULLIVAN C.: Perceptual evaluation of footskate cleanup. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2011), SCA '11, ACM, pp. 287–294. 3

[PV10] PETŘÍK O., VÁŠA L.: Finding optimal parameter configuration for a dynamic triangle mesh compressor. In *Proceedings of AMDO 2010* (2010), pp. 31–42. 4, 6

[TWC*09] TOURNIER M., WU X., COURTY N., ARNAUD E., REVÉRET L.: Motion compression using principal geodesics analysis. *Comput. Graph. Forum 28*, 2 (2009), 355–364. 2, 8, 9

[VS07] VÁŠA L., SKALA V.: Coddyac: Connectivity driven dynamic mesh compression. In *3DTV Conference Proceedings* (2007). 3

[VS09] VÁŠA L., SKALA V.: Cobra: Compression of the basis for pca represented animations. *Comput. Graph. Forum 28*, 6 (2009), 1529–1540. 3, 6

[VS11] VÁŠA L., SKALA V.: A perception correlated comparison method for dynamic meshes. *IEEE Trans. on Visualization and Computer Graphics 17*, 2 (Feb. 2011), 220–230. 2, 3, 7, 8

[WL08] WAREHAM R., LASENBY J.: Bone glow: An improved method for the assignment of weights for mesh deformation. In *Proceedings of AMDO 2008* (Berlin, Heidelberg, 2008), AMDO '08, Springer-Verlag, pp. 63–71. 7

[ZRB10] ZHANG L., RUSDORF S., BRUNNETT G.: Data-Driven On-Line Generation of Interactive Gait Motion. In *Proceedings of AMDO 2010* (07 2010), Springer Berlin / Heidelberg, pp. 250–259. 3

© 2014 The Author(s)
Computer Graphics Forum © 2014 The Eurographics Association and John Wiley & Sons Ltd.