

Perceptually Guided Fast Compression of 3D Motion Capture Data

A. Firouzmanesh, I. Cheng, and A. Basu

Abstract — *A time efficient compression technique, incorporating attention stimulating factors, for motion capture data is proposed. Compression ratios of 25:1 to 30:1 can be achieved with very little noticeable degradation in perceptual quality of animation. Experimental analysis shows that the proposed algorithm is much faster than comparable approaches using wavelets; thereby making our approach feasible for motion capture, transmission and real-time synthesis on mobile devices, where processing power and memory capacity are limited.*

Index Terms— **Motion Capture, Compression, Perception of Animation, Online 3D Environments.**

I. INTRODUCTION

USING motion capture data is an effective way to produce skeletal animations. In online applications, efficient compression of motion capture data can contribute to optimal use of available bandwidth while preserving the transmission of higher quality animation. In recent years, different approaches have been proposed to address the motion data compression problem. However, none of them directly discusses the possibility of achieving further reduction in data size with little noticeable perceptual degradation, considering the human visual system.

Even though perceptual factors in image and video have been widely studied [7], there is no existing qualitative metric [5] that considers different aspects of perception in animation. Thus, in this paper we propose a technique for perceptually guided compression of motion data, considering some of the most important factors, which can affect the perception of animation.

Motion capture refers to “the process of recording movement and translating that movement on to a digital model” [15]. Motion can be recorded using optical, mechanical or magnetic devices by tracking the movements of key points (such as joints) on an object (Figure 1). Motion capture is useful in many applications including military, entertainment, sports, medicine, computer vision and robotics. In computer animation, motion capture data can be used to create realistic 2D and 3D animated characters.

Motion data are usually sampled at frequencies between 60 to 240 Hz. A hierarchical biped structure (skeleton) is typically

used to coordinate the relative position and movement of each child key point with respect to its parent key point. The movement of a key point can be precisely described using 9 degrees of freedom (3 translations, 3 rotations, and 3 scaling factor) in the x-, y- and z- coordinates. We use the term channel to associate with one degree of freedom (DOF), tracing the trajectory of a key point during the entire motion sequence (animation). Given a defined number of key points, the size of each channel is dictated by the number of frames captured. Thus, if k markers are used to capture the movements of k key points in an n -frame sequence, the motion sequence can be encoded in $9k$ channels, with each channel recording n numbers defining the state of a key point at each of the n time slots. It is possible that the values of some channels remain zero during the entire animation. For body animation, very often movement of the whole object can be defined using at most 3 rotational DOF for each marker, plus 3 translational DOF for the whole object.

In online and interactive applications, motion capture data can consume a large amount of bandwidth especially when multiple dynamic characters are present in the scene. An important characteristic of realistic motion data is a considerable amount of spatial correlation between the different data channels and temporal correlation within each channel, which if processed and compressed effectively can reduce the data size considerably.

The primary motivation of this work is efficient data transmission for interactive online 3D environments (such as online games). To this end we propose a method which is capable of achieving an equivalent or better compression ratio compared to the state of art methods while consuming a much shorter compression and decompression time.

The rest of this paper is organized as follows. Section II introduces the basic concepts of motion capture in computer animation, reviews some of the important motion data compression approaches, and discusses their shortcomings for online transmission of interactive data. In the same section we also review the study results on human perception of animation. The proposed method is discussed in Section III followed by experimental results in Section IV. Finally concluding remarks and future work are outlined in Section V.

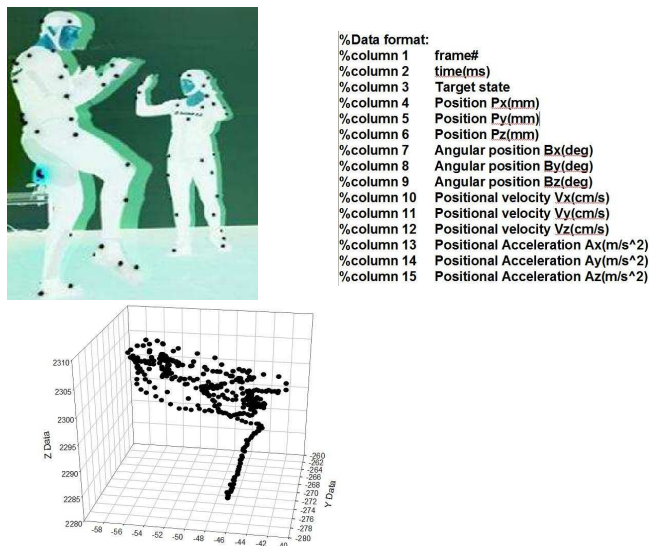


Fig. 1. An example of (left) a motion sequence captured using markers (black dots) on key points, (right) a file format of the captured data, and (bottom) a 3D trajectory generated by one key point during motion capture.

II. RELATED WORK

Current approaches to compression of motion data can achieve compression ratios of 25:1 to 35:1 without noticeable perceptual degradation [1], [2]. However each approach has some deficiencies in addressing the requirements of online 3D applications as discussed below.

A. PCA Based Approach

Arikan [1] proposed a method that can compress large databases of motion capture data, using a ratio from 30:1 to 35:1, with minimum amount of degradation in perceptual quality (Fig. 2). The input database consists of a file describing the skeleton structure of the animated character, plus a (typically large) number of motion clips. In their experiments they used two databases. The first one contained 620K frames sampled at 120Hz (1:30 hours long). The second one was the Carnegie Mellon motion capture database containing 2.9M frames sampled at 120Hz (6:30 hours long). The underlying skeleton was composed of 20 bones, coordinated with 56 DOFs.

Instead of working with joint orientation data, the authors use joint position data to avoid the problem that may arise from nonlinearity and hierarchical representation of joint orientation data. For each bone “the global position of 3 different and known points” in the bone’s local coordinate is calculated. These positions are called virtual markers.

A 3D cubic Bezier curve is then fit into the 3D trajectory of each virtual marker. The vectors containing Bezier curve parameter values generated from the virtual markers are clustered using Nystrom approximation. Principal Component Analysis (PCA) is performed on each cluster, and depending on the desired accuracy a number of larger rows of the resulting matrix are selected. The (x,y,z) positions of each virtual marker on the foot are considered as separate 1D

signals and compressed using Discrete Cosine Transform (DCT).

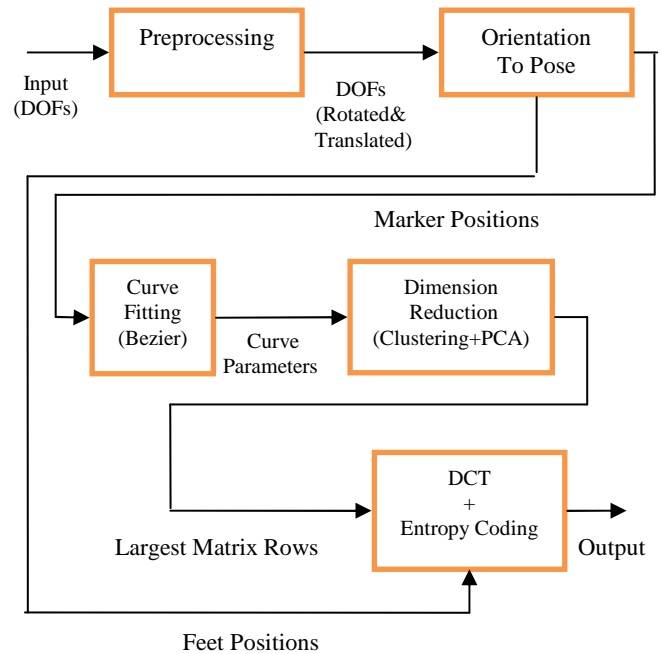


Fig. 2. PCA based approach.

While this approach achieves high compression ratios, it may not be suitable for some applications. First, for game and simulation engines that use joint angles representation, the overhead of converting marker positions to joint angles might be significant [2]. Second, PCA based approach achieves high compression ratios by exploiting correlation between DOFs in addition to temporal dependencies within each DOF. However, it might be desirable to achieve the same compression ratio while encoding each channel separately. For example, in some applications it is useful to combine different motion files to generate a new motion (*e.g.*, combining upper body of one motion with lower body of another one). Finally, this approach achieves high compression ratios only when applied to a large motion database, but for individual motion files with relatively small number of frames the compression ratio can be significantly lower [1]. This is because high compression is achieved when PCA is applied on large clusters.

B. Wavelet Based Approach

Using wavelet to compress the motion files can solve most of the problems in the PCA based approach discussed above. High compression ratios with low perceptual degradation can be achieved without exploiting the correlation between DOFs and without the overhead of converting joint angles to marker positions and vice versa. In wavelet based methods one dimensional wavelet transform is applied to each channel of motion data (Fig. 3). Depending on the desired compression ratio a percentage of the wavelet coefficients are kept and others are set to zero. This process is followed by quantization, run length encoding and entropy coding to produce the

compressed output stream.

Baudin et al. [2] proposed optimal wavelet coefficient selection for motion data compression. Their approach applies a distortion metric that takes into account the difference between the positions of each joint before and after compression, and gives higher weight to joints with more local influence (based on the length of the bone attached to the joint). Given a desired compression ratio, an optimal number of wavelet coefficients are selected for each channel, such that the distortion is minimized. Since the coefficient search space is very large, a discrete optimization algorithm is used to estimate the best solution. In order to reduce the artifacts related to foot skating, the difference between the actual position of a foot before and after coefficient selection is stored. For decompression, an inverse kinematic solver is used to correct the position of joints after moving a foot to its actual position. Their studies suggest that a compression ratio of 25:1 (for short motions) to 35:1 (for longer motions) can be achieved without any noticeable visual degradation.

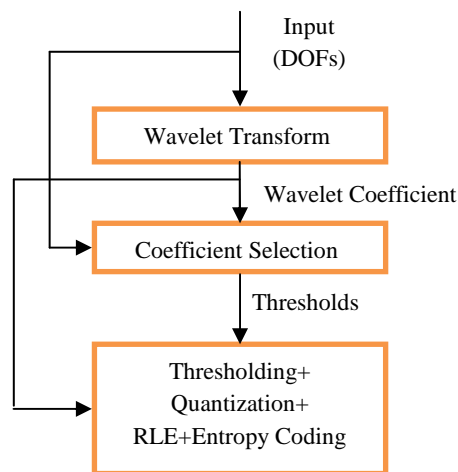


Fig. 3. Wavelet Based Approach.

While generating promising results, this approach has a number of shortcomings. First, the compression is very time consuming (133-327 ms per frame on a 2GHz AMD Athlon 64 bits) mainly because of the process of allocating an optimal number of coefficients to each channel. Also, the compression time per frame has a direct relation with the number of frames in the animation. Second, the error metric used for optimization is highly dependent on the joint hierarchy. In other words, giving more weight to the joints with high error rates does not necessarily yield good results.

C. MPEG-4 Bone-based Animation

In 2004 Bone-based Animation (BBA) was included in the MPEG-4 standard [16]. Motion is represented by variations in the DOF of joints in a skeleton (as described in Section I). For each frame, a binary string is used to record which DOF is modified, and only the values of the modified DOF are stored and transmitted.

Preda et al. [17] proposed two different methods for encoding and transmitting BBA. The first method is based on predictive coding followed by quantization and entropy coding. The second one consists of applying one-dimensional Discrete Cosine Transform (DCT) on each DOF in the skeleton, followed by quantization and entropy coding. They also proposed a frame reduction technique to achieve higher compression. Compression rates of the input BVH files vary from 5:1 to 45:1 depending on the method (DCT resulting in higher compression) and parameters, *e.g.*, quantization parameter, used.

D. Other Approaches

Onder et al. [4] presented two methods to reduce the number of key frames in a motion capture file. The first method is fitting a Hermite curve to each channel of motion data (rotation angles) using a dynamic programming approach. A Hermite curve can be reconstructed by specifying control points and tangent vectors. In the second method they use a curve simplification algorithm to represent each channel of data by a few keyframes. They applied their proposed methods on a short (22 frames) and a relatively long (1409 frames) motion. Both methods were capable of reducing the number of keyframes by 75-80% without any noticeable perceptual degradation. While the compression ratio is not as high as other approaches, their methods may be useful in editing motion data and motion understanding.

Tournier et al. [10] proposed a method based on Principal Geodesics Analysis (PGA) that exploits both temporal and spatial redundancy of motion data. PGA is an extension of PCA in non-Euclidian space (such as joint rotation angles). Joints in the skeleton are divided into three groups and each group is compressed in a different way. Root position and orientations are compressed using a lifting scheme. End-joints trajectory can be compressed using any standard linear compression methods, such as wavelets. PGA is performed on the rotation angles of inner joints. For decompression, a PGA-based inverse kinematics is applied that can recover the original position of inner joints using the root and outer joints positions. The authors reported a compression ratio of 18:1 to 182:1 for different types of animation. While this method achieves high compression for some motion data, it has a few drawbacks. First, the decompression is very time consuming because of the inverse kinematics. Also, the average decompression time per frame is directly related to the number of frames. Finally, the compression ratio varies significantly for different types of motion which is not desirable in online applications.

E. Measuring Perceptual Quality of Motion

The research on measuring the perceptual quality of animation is still in its early stage. A closely related problem is determining whether a motion is perceived as natural by human subjects. Approaches to quantifying natural human motion can be divided into three categories [9]. The first

category is defining some constraints on the movements of joints and looking for violation of these constraints. Two examples are conservation of angular momentum in flight, and “violation of the friction cone when the foot is in contact.” Another category, which will be discussed more below, is to define some threshold on the percentage of error in different aspects of animation (such as change in lengths, horizontal and vertical velocities, etc.) that can be obscured. The third category is to train a classifier based on data labeled by humans as natural or unnatural motions.

Reitsma and Pollard [8] performed experiments to measure the sensitivity of human subjects to violation of some of the laws of physics as a result of changes to the motion data. They systematically added errors to human jumping motion and measured the ability of subjects to detect these errors. Results of their studies suggest that: (1) Added acceleration is easier to detect than added deceleration; (2) Detection of low gravity is easier than high gravity; (3) Errors in horizontal velocities are easier to detect than errors in vertical velocities. They also proposed a formula to estimate the level of acceptable error, but mentioned that the specific results depend on many other factors including complexity of the geometric model.

Pollick et al. [6] studied the factors affecting perception of human movements. Results of their study suggest that for fast movements the judgment is mostly based on the velocity, while for slower, movements velocity along with another factor (not precisely defined) is used to distinguish between similar looking motions. Harrison et al. [3] performed a series of experiments to measure human sensitivity to changes in length in animation considering the effects of expectation, task interference, increase vs. decrease in length, duration of length change and division of attention. As a result of their studies, the authors suggest several guidelines for obscuring length changes in animation including: (1) A length change of up to 2.7% is almost unnoticeable by a human observer; (2) The change in length should not exceed 20% under any condition; (3) People are more sensitive to increase in length than to decrease; (4) Faster changes are more noticeable; (5) Sensitivity to length changes decreases during fast motions.

Ren et al. [9] investigated the possibility of developing a measure for quantifying the naturalness of human motion by training different classifiers (including a mixture of Gaussians, hidden Markov models, and switching linear dynamics systems) with a large database of motions. None of the classifiers were able to classify the test data with very low error rates. Also, they mention that this approach suffers from a few shortcomings including: (1) The measures may not be valid for motions that are significantly different from the training data; (2) Some types of error may not be reliably detected (for example, error occurring within a very short duration of time).

III. PROPOSED METHOD

The aim of this work is to develop a technique for lossy

compression of human motion capture data for online applications considering the following requirements: (1) Encoding and decoding should take a small portion of the processing resources (not just real time), since transmitting 3D animation online involves many other tasks with high resource demands besides the transmission of motion data; (2) It should be able to incorporate human perceptual factors of animation into the technique.

Among the different motion capture data compression methods discussed in the previous section, we find that wavelet coding provides the best technique considering the requirements mentioned above for several reasons. First, wavelet encoding proved to be one of the most efficient approaches for encoding multimedia data [13]. Second, even without considering spatial dependencies between motion channels, high compression ratios can be achieved. Finally, it is relatively straightforward to adjust the perceptual quality of motion. This is because each channel is encoded independently, and so if research studies show that one joint should be encoded more accurately, then the wavelet coefficients of the channels relating to that joint can be compressed with a lower ratio.

The basic idea is to select different numbers of coefficients for different channels of data based on the importance of each channel on the perceived quality of motion. Also, based on global properties of the object (such as distance from camera, number of objects present in the scene, degree of attention, etc.) the total number of coefficients to be transmitted could be multiplied by a factor (between 0 and 1). We did not use a global optimization method (such as that in [2]) because: (1) Either the compression or decompression would require high processing resources, which is not suitable for the intended application; (2) As discussed in [2], global optimization approaches might not provide good results when the distortion metric is not “dependent enough on the joint hierarchy.” This would be the case if the distortion metric gives higher weights to the error of some joints (*e.g.*, joints in contact with the environment).

In this work we try to optimize the coefficient selection algorithm considering two important factors. The first factor is the length of the bone connected to a joint. Larger bones have greater effect on the perceptual quality of an animation than smaller bones. It should be noted that we are considering the effect of a joint on the whole hierarchy because: (1) We are concerned with the perceptual quality of the coordinated animation, and not the individual position of each joint; (2) Finding the individual effect of each joint on its hierarchy is not straightforward and may become time consuming. For example, as mentioned in [2] “In a boxing animation sequence where the actor holds his arms close to his body, a small rotation of the torso can have less influence on the positional distortion than a rotation at the elbow or the shoulder.”

The evaluation of animation quality involves multiple factors, which include not only the movement itself but also other environmental factors such as the viewer’s region of

interest, texture masking and scene illumination. How environmental factors affect human perceptual quality is a large research topic worth separate discussion and is not in the scope of this manuscript. In our experiments, we apply regular lighting in an empty scene mapping simple plain colors on the animated character, and assume the viewer focuses on the animated character. This experimental setting provides a high likelihood for majority of the viewers to discover any artifact in the animation.

In order to locate the high attention regions, we apply the Interactivity-Stimulus-Attention Model (ISAM) [19], which explains how interactivity stimulates immersion of cognitive resources. ISAM was validated by involving 700 participants in their user studies. We hence assume that the amount of attention focused on a region is proportional to the amount of activity associated with that region. Similarly, in a motion sequence where the character moves its left hand more often but its right hand remains almost still, we assume the viewer pays more attention to the area around the left hand. In order to compare the amount of activity for each DOF, we measure the sum of its variation in a set of consecutive frames (see Step 4 of the proposed algorithm below).

The proposed method can be described by the following steps. Each channel of data ($c_i, 1 \leq i \leq N$) represents one DOF (usually rotational) of a joint of the skeleton. α is an input parameter to the algorithm indicating the percentage of the wavelet coefficients to be kept for the whole animation. The other inputs are QP (quantization parameter) in bits, and W (window size) which contains the number of frames that are compressed together in a single slice. The output of the algorithm is the compressed wavelet coefficients of the motion file. The processing steps are:

1. Divide the animation into M slices, having W frames in each slice (with the exception in the last slice, which can contain less than W frames).
2. For each slice m_k ($1 \leq k \leq M$) perform Steps 3 to 9:
3. Since the animated characters can be of different sizes, we use the normalized values (or relative measurement) to provide a comparable basis. Thus, for each channel we obtain the normalized bone length (l_i) by dividing the bone length attached to the corresponding joint by the maximum bone length in the skeleton.
4. Find the variation of each channel (c_i) using the following formula:

$$v_i = \frac{\sum_{p=m_i * W}^{m_i * W + W - 1} |c_{i,p+1} - c_{i,p}|}{W} \quad (1)$$

5. Combine the effects of relative bone length and variation to obtain a weighted quantity (A_i)

representing the relative importance (perceptual impact) of each channel.

$$A_i = l_i v_i \quad (2)$$

6. For each channel compute $K_i = \max(\frac{\alpha A_i N W}{\sum_{j=1}^N A_j}, C)$, the

largest (in absolute value) wavelet coefficient, and set other coefficients to zero (where C is a constant representing the minimum number of wavelet coefficients kept for each channel).

7. Quantize the remaining wavelet coefficients linearly into QP bits.
8. Use run-length to encode the wavelet coefficients.
9. Use an entropy coding method to compress the data.

IV. EXPERIMENTAL RESULTS

We implemented our compression algorithm using the C programming language. Daubechies D4 wavelet was used for encoding data. We used the Carnegie Mellon University ASF/ASM viewer [11] in our user studies. LZW compression [12] was used as the entropy coding method. The input data were in “asf” format, which described the skeleton structure; and in “amc” format, which defined the movements. In order to provide a format independent estimation of compression ratio, the motion data is converted into arrays with IEEE 32 bit (4 bytes) floats precision, where every 4 bytes represent one DOF in a frame. The skeleton hierarchy used was composed of 20 bones, controlled by a total of 56 DOFs (the number of DOFs does not change from frame to frame), and thus the input size was $56 \times 4 \times (\text{number of frames})$ bytes.

We performed a user study using two short motion sequences from the CMU [11] motion capture database. The first one labeled 2_2 is a simple walking motion and the second one labeled 49_14 is a dancing sequence. Both motions were captured at the rate of 120 frames per second. The motion sequences were compressed using three methods: 1-The proposed method with both bone lengths and variation in the degrees of freedom being taken into account; 2-The proposed method but only bone lengths being taken into account; 3-Standard wavelet compression (setting the same percentage of coefficients to zero for all the channels). For our method, motions were compressed using different parameters, e.g., quantization parameter (QP) and window size. A few sample videos of the original and compressed motion can be found at <http://cs.ualberta.ca/~firouzma/mocap>. Fifteen computer science students from the University of Alberta participated in this user study and were able to compare side by side the original and compressed motions. The latter were shuffled in random order before being presented to the users, who were asked to rank their quality from 1 to 5 in 0.5 steps, 5 being the best quality.

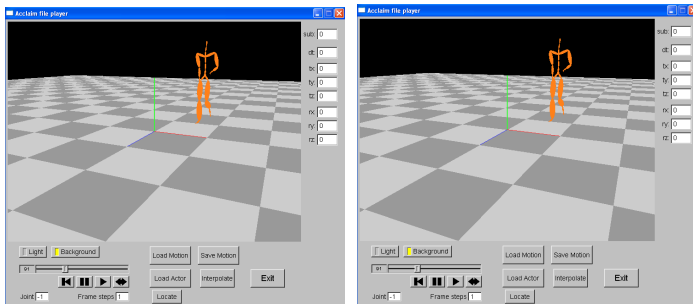


Fig. 4. Motion capture viewer environment.

Before starting the evaluation the users were introduced to the concept of quality degradation due to motion data compression. A total of 76 (38 for each motion) compressed motions were presented to the users in a random order. The compressed motions were generated by applying the two cases (with and without bone length) of our proposed method, and standard wavelet compression; using different compression ratios (1:10 to 1:70), different QPs (7 to 11) and different window sizes (64 to 256). Eight (four for each sequence) compressed motion files appeared twice in the test set. The users were allowed to see each motion as many times as they wanted but the total time of the test was limited to 40 minutes. Before starting the test a few compressed motions with a wide range of compression ratios (from 10:1 to 70:1) were shown to the users to help familiarize them with the difference in qualities.

Fig. 5 reports the user study results: The mean opinion score (MOS) for compression ratios from 20:1 to 40:1 for QP=9 and window size of 256 frames using the proposed method. The error bar represents confidence interval ($\alpha = 0.05$). Observe that for a compression ratio lower than 25:1 our method gets an average score of around 4.5 which indicates very little visible difference (and no difference to some viewers) between the perceptual qualities of the original and compressed motions. Also, for compression ratio 35:1 the average quality score is still close to 4.

Fig. 6 compares the quality scores of the proposed method with standard wavelet compression. For compression ratios over 30:1 our method shows a statistically significant improvement over the standard wavelet compression (the P-Value of one-tailed t-test comparing MOS of proposed method and standard wavelet compression with the same compression ratio was less than 0.05 for all the experiments).

In order to verify our choice of variation in rotation degrees as an attention factor we compare the quality of compressed motion using the proposed method (both bone lengths and variation in rotation) and the case where only bone lengths are considered (Fig. 7). Here again the results show a significant improvement when both factors are considered.

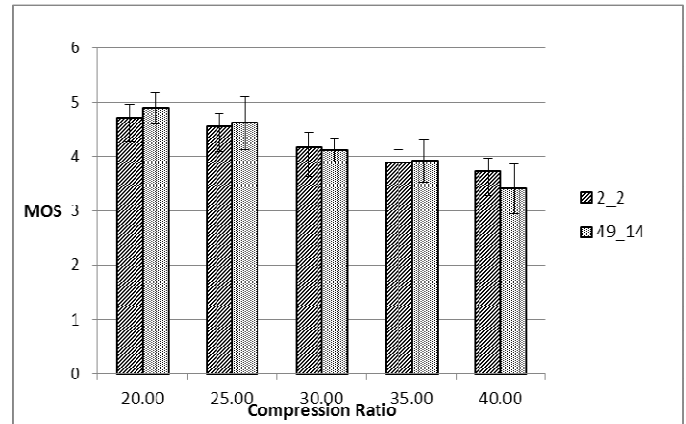


Fig. 5. Results of user study on two of CMU database motion files

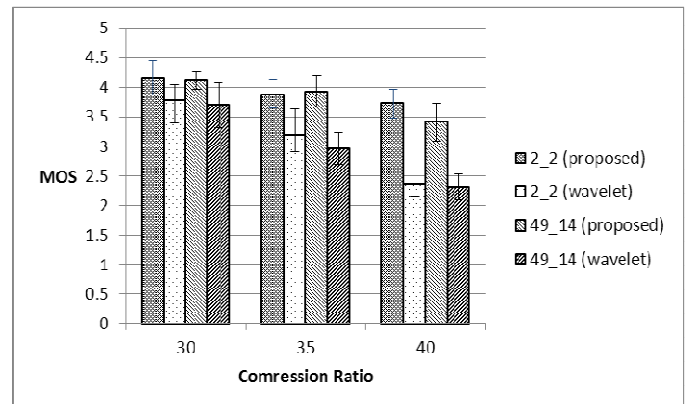


Fig. 6. Comparing the proposed method and the standard wavelet compression.

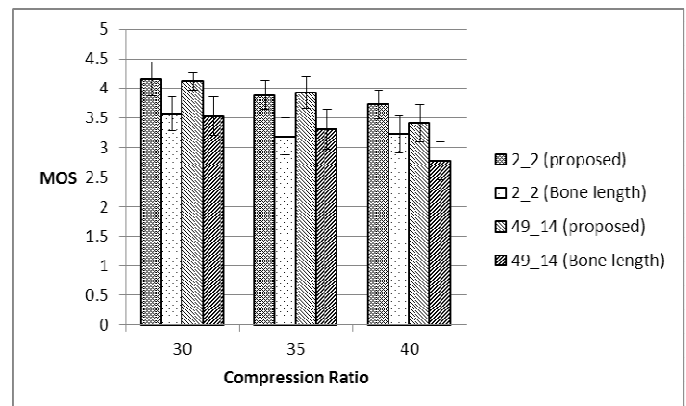


Fig. 7. Verifying the efficiency of selected perceptual factors.

The compression times of three motion files with different number of frames are shown in Table I. The compression times for shorter clips are not reported because animations in real time applications are much longer. Also, the imprecision in the timer can distort the results for short clips. The program was executed on a desktop computer with an Intel Core 2 Duo 2.66 GHz processor and 2 GB of memory. We can see that the average compression time per frame is about 25 μ s for relatively long motion files, which demonstrates that the algorithm has very efficient processing time.

TABLE I
Compression times (times are in μs)

Motion File	No. of frames	Total Time	Per Frame
14_9	3287	79687	24.2
1_4	4298	107500	25.01
25_9	5147	131250	25.5

It is reported by the adaptive wavelet compression method [2] that, for short sequences, a compression ratio of 25:1 is achieved without any noticeable degradation. This can be comparable to the average quality score of around 4.5 reported in Fig. 5 for the same compression ratio. The compression time reported in that paper is between 133 and 483 *ms* per frame on a 2 GHz AMD Athlon 64 bit which is approximately 10^4 times greater than our method. In Arikan's paper [1], a compression ratio of 30:1 is reported for compressing the whole CMU database, which contains 6:30 hours of animation at 120 frames per second (2.9M frames). However in this calculation the input files are not in 32 bit floats, but in a text based format which needs about twice as much storage. Based on the reported figures in that paper, the compression ratio is equivalent to less than 20:1 if the input is converted to 32 bit floats. Also, note that the statistics is based on compressing a very large motion database containing motion clips of various contents and lengths. It is not expected to obtain the same results when a single regular length motion clip is the target. In their case, a decompression time of 1.2 *ms* per frame is reported on a Pentium 4 @ 3.4 GHz which is 40 times greater than the decompression time of our algorithm.

V. CONCLUSION AND FUTURE WORK

We proposed a fast encoding and decoding technique for lossy compression of motion capture data, taking human perception into consideration. Experimental results show that our algorithm, in general, is much faster than other comparable methods. Initial studies suggest that compression ratios of at least 25:1 are achievable with little impact on perceptual quality. Since our method is faster while preserving an equivalent or better perceptual quality, the compressed motion data is more robust to constrained bandwidth, which is especially important in a mobile environment. In the current model, we introduce two attention factors: bone lengths and variation in rotation. However, there can be many other factors affecting the quality of animations, including the distance from camera, horizontal and vertical velocity of the object and the size of the limbs. Also, the efficiency of correcting contact positions using inverse kinematics [14] could be further studied. We believe our work presented in this paper will inspire more interest in the research community, resulting in more attention factors to be discovered, and the mathematical model to be extended and refined.

REFERENCES

- [1] O. Arikan, "Compression of motion capture databases," ACM Transactions on Graphics (TOG), Volume 25 Issue 3, July 2006, pp. 890-897.

- [2] P. Beaudoin, P. Poulin, M. Panne, "Adapting wavelet compression to human motion capture clips," in Proc. Graphics Interface Conference, Montreal, QC, May 2007, pp. 313-318.
- [3] J. Harrison, R. A. Rensink, and M. Panne "Obscuring length changes during animated motion," ACM Transactions on Graphics (TOG), VOL 23; PART 3, 2004, pp. 567-571.
- [4] O. Onder et al., "Keyframereduction techniques for motion capture data," in proc.3DTV-CON'08, Istanbul, Turkey, May 2008, pp.293-296.
- [5] Y. Pan, I. Cheng, and A. Basu, "Quality metric for approximating subjective evaluation of 3D objects," IEEE Transactions on Multimedia, Vol. 7, Issue 2, April 2005, pp.269-279.
- [6] F. E. Pollick, J. G. Hale, and P. McAleer, "Visual perception of humanoid movement," In Proc.Third International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, 2003, pp. 107-114.
- [7] M. G. Ramos, and S. S. Hemami, "Perceptual quantization for wavelet-based image coding," in Proc. IEEE Int. Conf. on Image Processing, Vancouver, B.C., September 2000, pp.645-648.
- [8] P. Reitsma, and N. Pollard, "Perceptual metrics for character animation: Sensitivity to errors in ballistic motion," in proc.ACM SIGGRAPH'03, San Diego, CA, July 2003, pp. 537-542.
- [9] L. Ren, "A Data-Driven Approach to Quantifying Natural Human Motion," ACM Transactions on Graphics (TOG), VOL 24; NUMB 3, 2005, pp. 1090-1097.
- [10] M. Tournier et al., "Motion Compression using Principal Geodesics Analysis", Computer Graphics Forum, Vol. 28, Issue 2, April 2009, pp. 355-364.
- [11] CMU Graphics Lab, "CMU Graphics Lab motion capture database," Available: <http://mocap.cs.cmu.edu>.
- [12] Z. Li, and M. Drew, Fundamental of Multimedia, Pearson Education Inc., First Edition, 2004, ch. 7.
- [13] R. Gonzalez, and R. Woods, Digital Image Processing, Pearson Education Inc., Third Edition, 2008, pp. 604-614.
- [14] L. Ikemoto, O. Arikan, and D. Forsyth, "Knowing when to put your foot down", in Proc. Symposium on Interactive 3D Graphics and Games, Redwood City, California, 2006, pp. 49-53.
- [15] Wikipedia, "Motion capture," available: http://en.wikipedia.org/wiki/Motion_capture.
- [16] ISO/IEC JTC1/SC29/WG11, "Animation Framework eXtension (AFX)", 2004, Published by ISO.
- [17] M. Preda, B. Jovanova, I. Arsova, and F. Preteux, "Optimized MPEG-4 animation encoder for motion capture data," in Proc. Web3D'07, Perugia, Italy, April 2007, pp. 181-190.
- [18] P.B. Lowry, N. Twyman, J. Gaskin, B. Hammer, A. Bailey, and T. Roberts, "Proposing the Interactivity-Stimulus-Attention Model (ISAM) to explain and predict enjoyment, immersion, and adoption of purely hedonic systems," in Proc. SIGHCI, Montréal, QC., December 2007, pp. 72-76.