Perceptually Guided Synthesis and Compression of Motion Capture Data

by

Amirhossein Firouzmanesh

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

Abstract

Using motion capture data is an efficient way to generate and transmit 3D character animation. We explore the possibility of incorporating human perceptual factors in compression and synthesis of motion capture data in order to achieve a higher performance in different aspects including reducing the bandwidth requirement, decreasing processing time and transmission delay in different scenarios. First we show that by incorporating perceptual factors in wavelet-based compression, the processing time could be significantly reduced without noticeable degradation in the reconstruction quality. 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.

We also propose a compression method based on using motion primitives. Using incremental encoding plus a database of motion primitives for each key point, our method achieves a higher or competitive compression rate with less online overhead. Trade-off between visual quality and bandwidth usage can be tuned by varying a single threshold value. A user study was performed to measure the sensitivity of human subjects to reconstruction errors in key rotation angles. While achieving real-time performance, our technique outperforms other methods in our experiments by achieving a compression ratio exceeding 50:1 on regular sequences without noticeable degradation in rendered qualities.

Finally we propose a high efficiency, fast, scalable method for compressing motion capture clips taking advantage of a 1-D variation of the SPIHT algorithm. SPIHT provides near-optimal reconstruction error using the allocated bits. A 1-D variation of the original SPIHT is proposed that works with individual channels of motion capture data. Instead of time-consuming optimization process we combine 1-D SPIHT with a bit rate allocation mechanism based on perceptual factors to distribute the available bandwidth between the channels of data based on their importance in perceptual reconstruction quality of the motion. Our studies show that the proposed method is capable of compressing data at a rate of 40:1 to 60:1 with close to perfect reconstruction quality, which is generally better than the current methods in the literature.

Preface

Chapter 3 of this thesis has been published as A. Firouzmanesh, I. Cheng, A. Basu,“Perceptually guided fast compression of 3-d motion capture data,” *IEEE Transactions on Multimedia*, vol. 13,issue 4, 829-834. I was responsible for proposing and implementing the suggested method, designing and performing user studies and analysing the results as well as the manuscript composition. I. Cheng and A. Basu were the supervisory authors and were involved with concept formation and manuscript composition.

Chapter 4 of this thesis has been published as A. Firouzmanesh, M. Lindgren, T. Drummond, I. Cheng, A. Basu,“ Efficient compression of rhythmic motion using spatial segmentation and temporal blending,” Proceeding of *IEEE Multimedia and Expo Workshops (ICMEW)*, 2013. I was responsible for proposing the suggested method and analysing the results as well as the manuscript composition. M. Lindgren and T. Drummond were responsible for implementing the proposed method as well as literature review. I. Cheng and A. Basu were the supervisory authors and were involved with concept formation and manuscript composition.

Chapter 5 of this thesis has been published as A. Firouzmanesh, I. Cheng, A. Basu,“ Perceptually Motivated Real-Time Compression of Motion Data Enhanced by Incremental Encoding and Parameter Tuning,” Proceedings of *Eurographics 2013-Short Papers*. I was responsible for proposing and implementing the suggested method, designing and performing user studies and analysing the results as well as the manuscript composition. I. Cheng and A. Basu were the supervisory authors and were involved with concept formation and manuscript composition.

Acknowledgements

I would like to thank NSERC - Canada, AITF - Alberta, and the Department of Computing Science - University of Alberta, for their financial support throughout my PhD education.

Also, I wish to thank CMU Graphics Lab and Mr. Bruce Hahn who have provided freely accessible motion capture data and open source libraries.

I would like to thank my supervisors Dr. Anup Basu and Dr. Irene Cheng for their guidance and support throughout this research. Many thanks to my PhD defense committee members especially Dr. Frédéric Dufaux and Dr. Pierre Boulanger for their time and their valuable comments. I also would like to thank Dr. Hossein Azari, Dr. Rui Shen, Mr. Nathaniel Rossol as well as other members of the Multimedia Research Center for their constructive feedbacks.

Finally, I wish to express my deep appreciation to my dear wife Zahra for her support and patience during my research. I am thankful to my father and mother for their support and encouragement.

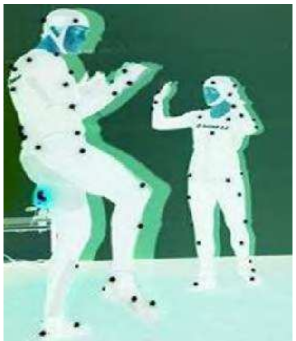
Table of Contents

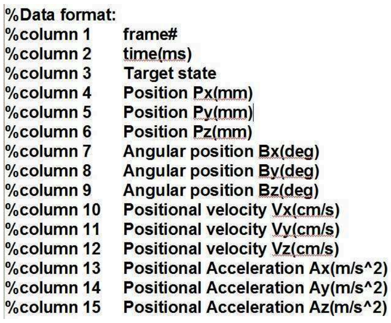
List of Tables

List of Figures and Illustrations

CHAPTER ONE: INTRODUCTION

In this research we are going to study the effectiveness of using perceptual factors of animation to improve the visual quality and performance of methods for compression and synthesis of human motion capture data. Motion capture is the process of recording the movements of one or several actors using tracking equipment [PP10]. Motion can be recorded using optical, mechanical or magnetic devices by tracking the movements of key points (such as joints) on an object (Figure 1.1). Motion capture is useful in many areas 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.





```
%Data format:
%column 1     frame#
%column 2     time(ms)
%column 3     Target state
%column 4     Position Px(mm)
%column 5     Position Py(mm)
%column 6     Position Pz(mm)
%column 7     Angular position Bx(deg)
%column 8     Angular position By(deg)
%column 9     Angular position Bz(deg)
%column 10    Positional velocity Vx(cm/s)
%column 11    Positional velocity Vy(cm/s)
%column 12    Positional velocity Vz(cm/s)
%column 13    Positional Acceleration Ax(m/s^2)
%column 14    Positional Acceleration Ay(m/s^2)
%column 15    Positional Acceleration Az(m/s^2)
```

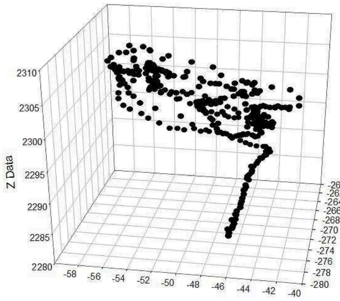


Figure 1.1: An example of (left) a motion sequence captured using markers (black dots) on key points, (right) a file format of the captured data, and (bottom) a 3D trajectory generated by one key point during motion capture.

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 DOFs for each marker, plus 3 translational DOF for the whole object.

1.1 Motivation

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.

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. For example, uncompressed motion of a single character with 60 degrees of freedom would require a 120 Kilobits per second bandwidth. 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.

In many simulation and training applications, a population of human-like characters needs to be present in the environment. Recent improvements in graphics hardware make it possible to render characters with realistic static appearance. These improvements raise the expectations of users regarding characters' motions. The appearance of human characters in simulation applications used for training military, rescue, and other hazardous-duty personnel are not yet very realistic. This limits the potential effectiveness of the simulation. For example, a fire-rescue worker is expected to respond very differently when presented with a character that looks and behaves similar to a frightened child than when he or she is presented with a mannequin designated with the word "victim." Characters with natural looking behavior tend to help the simulation users become more immersed in the virtual environment to the point where they behave as if in the real situation.

Rhythmic motion including dance and martial arts movements constitutes an important class of motion capture data with a wide range of applications. For example, motion capture is used in computer aided dance training systems. These systems work by displaying pre-recorded dance movements and ask users to mimic what they see [DLGY11]. The motion of the user is then captured (using a motion capture device such as Microsoft Kinect) and feedback is provided comparing the motion of the user with the recorded animation. In computer aided choreography, in order to design a dance motion for a new piece of music, instead of creating the entire dance sequence from scratch, the choreographer can combine recorded motion segments from previous dances [FXG12]. In the field of humanoid robots, scientists have tried to mimic the dance performance of a human character with a biped humanoid robot [SI08]. While there are several works in the literature that address compression of motion capture data in general, very few take advantage of the repetitive nature of rhythmic motions to achieve a higher compression rate.

1.2 Problem Definition, Focus, and Scope

In recent years, different approaches have been proposed to address the motion data compression problem. However, unlike video and image, perceptually motivated compression of motion capture data has remained untouched. In this thesis we investigate the possibility of improving the efficiency of the task of motion capture compression considering the properties of human visual system (HVS).

Motion primitives can be used to achieve higher compression rates especially for motions with rhythmic and/or repetitive natures. An important problem that needs to be addressed is natural appearance (high perceptual quality) of the reconstructed motion. It is desirable to consider the properties of HVS as well as the results of user studies when designing a compression method based on motion primitives.

Besides compression ratio and processing time in online applications, there is a need for low transmission delay, efficient bitrate allocation and scalability. We investigate the effectiveness of incorporating perceptual factors in a framework that addresses all of the abovementioned requirements.

1.3 Contributions

1) We propose and verify a method that incorporates perceptual factors of animation in a wavelet-based compression framework that can achieve comparable compression to the state of the art while reducing the processing time significantly [FCB11].

2) A compression method using motion primitives is proposed, which incorporates perceptual factors of animation along with the results of user studies that measure the sensitivity of human subjects to reconstruction errors [FCB13][FCB13b]. The proposed method can compress different types of motion at a very high rate, especially motions with rhythmic and/or repetitive nature.

3) We propose a high efficiency compression framework using a 1-D of set partitioning in hierarchical trees (SPIHT) [FCB14]. The proposed framework provides fast and accurate bandwidth allocation as well as efficient scalable data transmission. We incorporated four important perceptual factors of animation in the bandwidth allocation module to achieve very low bandwidth requirement while keeping the reconstruction quality close to perfect (as verified by user studies).

1.4 Organization of the Thesis

The rest of this thesis is organized as follows: In chapter 2 we cover background and related work in different areas including studies on human visual perception, perceptual video compression, previous motion capture compression and synthesis methods, and the motion level of details. Chapter 3 covers the details of our first contribution and show that by incorporating perceptual factors in wavelet based compression we can significantly reduce the processing time without any noticeable degradation in perceptual quality. In chapter 4 we propose a general framework to achieve MoCap compression using motion primitives. In other words, we discuss the possibility of incorporating motion synthesis methods to achieve higher compression. In Chapter 5 we propose a method for MoCap compression using previously created databases of motion primitives and show the effectiveness of incorporating perceptual factors in the compression process. Chapter 6 proposes a more general-purpose high efficiency compression method suitable for multi-user online applications and discusses the details of our third contribution. Chapter 7 concludes this thesis and suggests some avenues for future research.

CHAPTER TWO: BACKGROUND AND RELATED WORK

In this chapter, we first review related studies on human visual system (HVS) and discuss how perceptual factors have been successfully incorporated in video coding methods. After that we review current studies on motion perception and measuring the perceptual quality of motion. This is followed by a review of current motion compression and synthesis methods as well as the closely related topic of the motion level of details.

2.1 Related Studies on Human Visual System

2.1.1 Visual Attention

There are two widely accepted models of human visual attention. Both models assume that visual attention is a two stage process. The first stage consists of analyzing the whole visual field in parallel, and the second stage is focusing on specific parts of the scene and processing information sequentially. The first model (spotlight model) [Jon80] defines three factors for visual attention, namely focus, margin and fringe (figure 2.1). The focus is the area around the center of attention where the scene is captured at a high resolution. Fringe is the area surrounding the focus where the data is captured at a low resolution. Margin is the outer boundary of the fringe. The second model (zoom-lens model) [EH72] extends the spotlight model by introducing the change in focus size property. The assumption is that attentional resources are fixed and so the processing time has an inverse relationship with the size of the focus. The minimum size of the focus area is believed to be 1° of visual angle while the maximum size is still unknown.

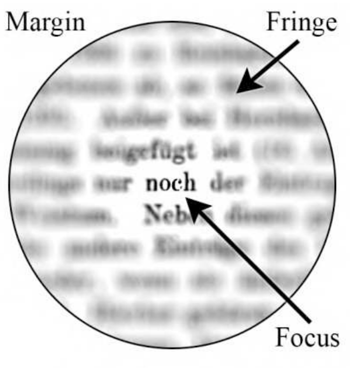


Figure 2.1: Focus, fringe and margin [WIKI2].

2.1.2 Fovea

The fovea is a part of the retina with the highest density of cones [II86] (Figure 2.2). About half the information carried by the optic nerve is from the fovea while the rest is from other parts of the retina. This area provides high resolution central vision that is necessary for activities such as reading and driving that require visual detail.

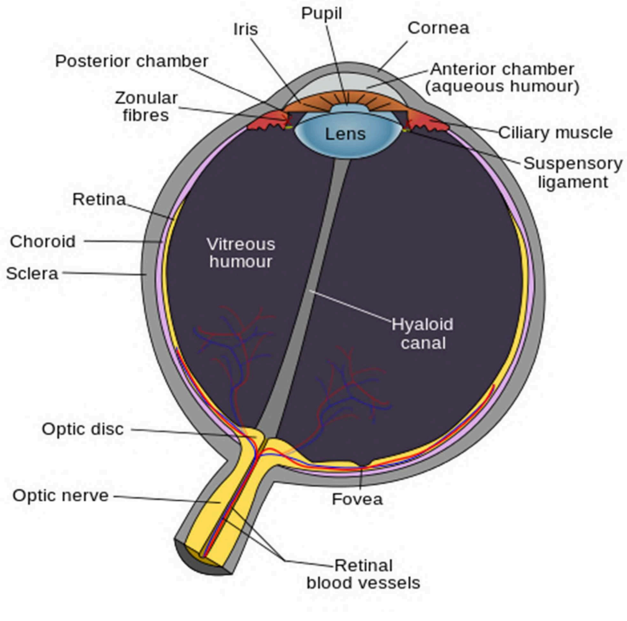


Figure 2.2: Schematic diagram of the human eye [WIKI1].

2.1.3 Eye Movements

While there are several types of movements defined for the eyes, the three most important ones in the field of perceptually adaptive graphics are smooth pursuit, saccades and vergence movements. Smooth Pursuit Eye Movements (SPEM) [SK08], a voluntary way of shifting gaze, enable us to follow a moving object. However, most people are not able to initiate SPEM in the absence of a moving object. The smooth pursuit ability is defined as the ability to smoothly pursue an object (without making saccades). For humans, this ability tends to be stronger in a horizontal than a vertical direction.

Saccades are fast non-smooth movements of both eyes in one direction [Fin09]. An important functionality of saccades is to move the fovea around the scene so that "interesting" pieces of the scene can be scanned at a high resolution. The speed of a saccade between two stops cannot be controlled consciously (the eyes move as fast as possible). The frequency of saccades is typically between three and four per second.

Vergence movements help in depth perception are the movements of the eyes such that the image of the object is formed on the corresponding locations on both retinas [PMM04].

2.1.4 Visual Saliency

Saliency of a part of an image is the extent to which that part stands out from its neighbors [Itti07]. Saliency detection helps human beings (as well as other organisms) to absorb the most important information from the scene in a short amount of time using their limited cognitive resources. Saliency is usually due to the contrast between the salient part and its neighbors. As an example in a scene consisting mostly of white dot a red dot would be a salient part. Generally, visual saliency detection is a bottom-up process. Attention can also be guided by top-down, user-driven factors. As it is difficult for humans to pay attention to more than one item at the same time, they need to continuously integrate and prioritize different attention influences.

2.1.5 Contrast Sensitivity

Contrast sensitivity measures the ability of a person to discern different luminance levels in a static scene [WIKI3]. Contrast sensitivity test is done by showing images of different contrast and spatial frequency (typically parallel bars) and asking the subject whether they can detect each image. The result of the test is a curve called contrast sensitivity function (CSF) describing the minimum contrast threshold for different spatial frequencies. The human CSF acts like a band-pass filter with maximum sensitivity at around 4 cycles per degree (CPD) and high-frequency cut-off of about 60 CPD (figure 2.3).

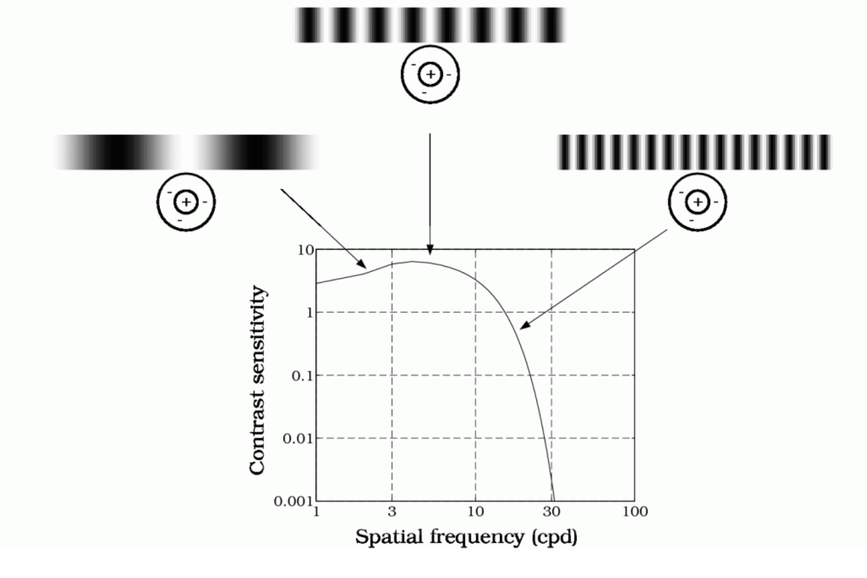


Figure 2.3: The human Contrast Sensitivity Function (CSF) [WIKI4].

2.1.6 Motion Perception

Human perception of motion is a topic of interest in neuroscience. In [Joh73] it is shown that under certain conditions the human visual system can interpret the trajectory of a series of moving points (point-light walker) as the movement pattern of an actor. This simple representation allows the scientists to study motion perception independent from other visual information. Besides recognizing the human walk, the HVS is able to extract more information such as the mood of the subject (*e.g.* happy or nervous) from a point-light system [Tro02].

2.2 Perceptual Visual Quality Metrics

While defining a metric for perceptual quality of motion in its early stages, perceptual visual quality metrics (PVQM) for image and video is a well-established topic. Several studies show that traditional signal fidelity metrics like MSE (mean square error) do not correlate very well with the perceived visual quality [LJ11]. The perceptual quality of an image/video can be found by performing a series of user studies. However, in many applications it is not feasible to perform user studies for every piece of data. What can be done instead is to define a model that efficiently predicts the quality score of the data in a standard test dataset. These datasets usually store the Mean Opinion Score (MOS) of a relatively large population for a set of images/videos.

The visual quality metrics found in the literature take the following facts about HVS into account:

(1) Some changes in a signal are not noticeable (the just noticeable difference or JND model).

(2) Different regions of a signal receive different levels of attention.

(3) HVS does not perceive all changes to the signal as degradation.

(4) Same magnitude of change in different channels results in different perceptual effect.

The building blocks or modules of PVQMs can be divided into four categories: Image/Video Decomposition, Features and artifacts detection, Just-noticeable distortion (JND) modeling and visual attention map generation. PVQMs might combine metrics that fall into one or more of these four categories.

Image or video can be decomposed into different color, spatial and temporal channels. It is known that the HVS has different sensitivity curves to different components of the signal. For example, HVS is more sensitive to changes in luminance than changes in chrominance [KB96]. Therefore, PVQMs can give a better estimate of the perceptual quality by giving more weight to the changes in channels based on the HVS response curve of the channel.

Several types of artifacts have significant effect on the perceptual quality of the image/video. Those include blockiness, blurring, edge damage and ringing. Traditional distortion measures such as MSE do not efficiently reflect the effect of these artifacts. Also some features of the signal have a more direct effect on the perceptual quality. For example, studies show that HVS has special cells for processing signal contrast and therefore extracts more information from contrast than the absolute signal [Kuf53]. For this reason contrast is used directly or indirectly in many PVQMs.

JND can be defined as the minimum amount of distortion below which the majority of the users (e.g. 75%) cannot tell the difference between the original and the distorted signal. A PVQM can incorporate JND by ignoring the changes that fall below the JND value. In the context of image/video coding, JND can be defined in subbands of the signal decomposition (e.g.

DCT or wavelet subbands) or at a pixel level. For example the DCT-subband luminance JND function could be expressed as:

$$JND_D(n,i,j) = T_{s,csf}(n,i,j) \prod_{\rho} \alpha_{\rho}(n,i,j) \tag{2.1}$$

where n is the block number, $T_{s,csf}(n,i,j)$ is the spatial CSF related threshold, and $\alpha_{\rho}(n,i,j)$ are elevation parameters taking different effects such as luminance adaptation, intra-band masking, inter-band masking, temporal masking, and chrominance masking into account [LJ11].

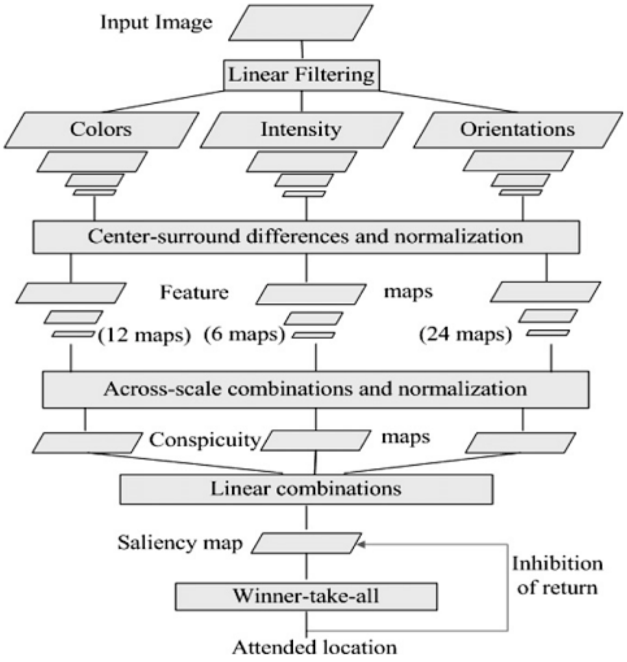


Figure 2.4: An example of visual attention map generation process [IKN98]. Copyright IEEE

Different parts of an image receive different levels of attention. This is because HVS detects parts of the image as important and using saccades directs the fovea to these areas for a more detailed analysis. Therefore, changes in the areas that grab the attention of the HVS have a stronger effect on the perceptual quality of the signal. A visual attention map assigns a score to each part of the image based on its predicted relative importance to the HVS. The visual attention map can be generated in a two-step process by first extracting and then combining a series of low-level features of the image (such as colors, intensity and orientation). As an example the visual attention map generation process used in [IKN98] is depicted in figure 2.4.

PVQMs first extract the features used in evaluation and then combine the features to report a single number representing the quality score of the signals. The features extracted might be from one or more of the modules discussed above. PVQMs can be model-based, meaning that features are created by applying a gain control function to each channel of the signal. The gain control function is derived from the HVS response curve of that channel. However, most recent PVQMs are signal-driven because model-based methods are computationally expensive, and also complete knowledge of HVS response curves especially in more complicated scenarios (where more than one or two stimuli are present) is not yet available.

2.3 Perceptually Guided Video Compression

Perceptual video compression methods try to incorporate the properties of HVS to achieve a higher compression ratio in a way that human observers do not notice any significant degradation in the quality of the video [LE12]. The three important aspects of perceptual video compression methods are as follows:

1) The HVS perceptual mechanisms are exploited to achieve higher compression;

2) The way those mechanism are incorporated in the coding process;

3) Validation method

The perceptual mechanisms that are usually incorporated in perceptual video compression methods include contrast sensitivity, masking, fovea and visual attention, all of which were discussed in section 2.2.

In terms of using perceptual mechanism, the perceptual compression techniques can be divided into four categories:

1) In some methods the region of interest (ROI) needs to be identified by specialists prior to encoding. Some medical video compression methods have used this approach [MH10].

2) ROI can also be detected based on user input while he/she is watching the video. The device used for identifying the ROI can be as simple as a mouse or more complicated devices such as eye-tracking equipment. Because of the nature of this approach (dependence on user feedback), it is not suitable for broadcasting applications.

3) Many perceptual compression techniques try to incorporate the computational methods of human visual attention to separate perceptually important from unimportant regions of the scene. While there are several computation models for attention, three models have been found to be effectively applicable to perceptual video compression:

> A) Bottom-up attention models (*e.g.* [Itt04]) combine several visual cues including color, intensity, orientation and motion in a way that resembles the response characteristics of low-level visual processing neurons in the brain.
>
> B) Top-down attention models try to identify semantically important parts of the scene. Examples of semantically important regions include the human face or skin area, which have been widely used in video conferencing applications, human upper body (sign language) and moving objects (more general application).

C) Researchers have tried to combine bottom-up and top-down models to prepare a combined attention model. This can be achieved by combining a saliency model with top-down features like face, skin or caption [CHN06, BMN*08, TRP07].

4) Using human visual sensitivity thresholds is a common approach in perceptual video coding. The concept is to allow distortion below the visually detectable levels. The distortion thresholds of different pixels in an image differ based on several factors, including correlation of pixels to their spatial and/or temporal neighbors in terms of luminance, color or semantics. Also the ROI detection methods might be incorporated to take into account different sensitivity in ROI vs. non-ROI. For example, [BSO11] take advantage of the fact that regions with complex texture have lower sensitivity to distortion and segments the frame into texture and non-texture regions. For the texture regions, they try to find and transmit the parameters of a model that can be used to synthesize the texture. The non-texture regions are encoded with the regular compression method. Tang et al. [TCYT06] use motion information along with texture structure by defining a visual distortion sensitivity index (VDSI). An object with attention-grabbing movement and high texture randomness can hide a large amount of distortion and therefore receives a very high VDSI. For stationary objects or objects with less attention grabbing movement the VDSI is calculated based on the degree of their texture randomness.

2.3.1 Implementation

There are two major ways to use perceptual models for the purpose of video encoding. The first approach is to preprocess the video in a way that different parts of the scene are distorted non-uniformly based on their perceptual importance. A common way to achieve this is by applying a spatial blurring filter with different blurring strengths in different regions [NH10]. The reason this approach leads to higher compression is that blurring removes the high frequency components and therefore less bits are needed to transmit highly blurred regions. Sheikh et al. [SEB03] try to simulate the foveation concept by applying a low-pass filter with cut-off frequency decreasing based on the distant from the fixation point. In [BSW93] the variable resolution (VR) transform is extended to address the problem of foveated video compression. Instead of the original video, the result of applying the extended VR transform is encoded and transmitted. The receiver decodes the data and applies inverse VR transform to retrieve the compressed video.

While the preprocessing approach is relatively simple and can be independent from the encoder/decoder, the coding gain is not as high as when perceptual models are embedded into the encoder. In embedded encoding, the non-uniform distortion allocation is achieved by changing the encoder parameters for different parts of the scene.

For DCT-based methods such as H.26x, these parameters include quantization parameter (QP), motion estimation (ME) search range and motion estimation accuracy. In H.264/AVC, each frame can be divided into several slides of MB and each slice can be encoded with a different QP. This feature has been used in [ACBD03], [ADCB07] and [CYC09] for the purpose of perceptual video compression.

In [LQI11] the bitrate allocation problem is formulated as a constrained optimization problem which tries to minimize the bitrate such that distortion time saliency of each MB is fixed. The result is different QPs for different MBs based on the perceptual importance of MBs.

Wang et al. [WZLG09] assign different H.264 coding parameters to ROI and non-ROI. For example, a large motion estimation search range is assigned to ROI and a small one to non-ROI. Also, not all the available prediction modes are considered for non-ROI.

Weibe and Basu [WB01] address the problem of the possibility of information loss with congestion in the asynchronous transfer mode (ATM) protocol. They propose an intelligent, fovea-driven priority assignment of image data that can reduce the negative impact of information loss over ATM networks.

Instead of reducing bitrate while minimizing perceptual degradation, perceptual rate control (RC) methods try to maximize the perceptual quality given the available bandwidth. This is usually achieved by minimizing a distortion measure that takes into account unequal importance of different frames, visual objects and/or MBs. [CCYJ08] gives different manually defined weights to the distortion of ROI and non-ROI MBs and uses a fuzzy controller to calculate the optimal QP for each MB. In [LPB01], a foveal signal-to-noise ratio is defined and used as the distortion metric in the rate-distortion (R-D) model. Perceptual RC can also be done by allocating more bits to more perceptually important regions. Daly et al. [DMR01] distribute the available bandwidth between the foreground and the background regions according to size, variance in pixel value and the PSNR of each region.

MPEG-4 enables object-based video coding, which can then be used for perceptual RC. Different quality levels can be assigned to different video objects based on their perceptual importance [CHN06].

In networks having users with different networking conditions and terminal types (e.g. mobile, PC, HDTV), scalable video coding can provide a bit stream that is adaptable to end-user conditions. A scalable video stream provides a base layer that is transmitted to all clients and several enhancement layers that can be either discarded or used to improve the quality of the base layer. Perceptual video coding can be embedded in scalable codecs. Wang et al. [WLB03] propose a wavelet based foveation scalable video coding. The bit stream is ordered in such a way that the information of the attended area of the scene is placed at the start. Therefore, if a bit stream is truncated because of network limitations, the information around the fixation point is still transmitted. Based on the scalable extension of H.264/AVC (SVC), [CZYZ08] assign different priority scores to ROI and non-ROI in enhancement layers so that if needed non-ROI data are discarded first. Based on a model called visual sensitivity profile, [ZCYZ08] define and use multi-layer saliency maps in spatial scalability layers of SVC.

2.3.2 Evaluation and Validation

In order to determine the suitability of a perceptual video coding method for a specific application the following criteria should be considered:

- **Coding gain:** At the same quality level, how much bit rate is saved in comparison with the original coding method?
- **Quality:** The quality of the compressed video can be assessed using subjective tests (user studies) or PVQMs as discussed in section 2.2.
- **Computation complexity:** If implementing a perceptual model has a significant processing overhead, it might not be suitable for real-time applications. Therefore, one area of research is to reduce the computation complexity of perceptual models [SEB03], [CN99], [SALZ06].

2.4 Perceptually Optimized Transmission of 3D Objects

Pan et al. [PCB05] propose a perceptual quality metric for 3-D objects considering the effects of resolution of texture and the geometry (number of vertices). The metric is derived by statistical analysis of the data collected from a 3-D quality evaluation user study. In the user study, five different visual objects each at six levels of wireframe resolution and three levels of texture resolution (for a total of 90 stimuli) were presented to the participants. The participants were asked to rate each stimuli from 1 (worst quality) to 5 (best quality) by comparing the stimuli to two reference stimuli (one with the highest geometry and texture resolution and the other with the lowest geometry and texture resolution).

Cheng and Basu [CB07] used this metric to propose a strategy for perceptually optimized transmission of 3-D objects. They suggest and examine the effectiveness of a few different strategies for packet transmission of both 3-D texture and mesh with respect to preserving 3-D perceptual quality under packet loss.

2.5 Perceptual Factors of Animation

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 [RPE*05]. The first category defines some constraints on the movements of joints and looks for violation of these constraints. Two examples of this are conservation of angular momentum in flight and violation of the friction cone [Ren06]. Another category, which will be discussed more below, defines some threshold on the percentage of errors in different aspects of animation (such as change in lengths, horizontal and vertical velocities, etc.) that can be obscured. The third category trains a classifier based on data labeled by humans as natural or unnatural motions.

Reitsma and Pollard [RP03] 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 the following: (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. [PHM03] studied the factors affecting perception of human movements. Results of their study suggest that when two motions have similar paths, the velocity is an important factor for differentiating them, especially for fast movements. For slower movements humans tend to pay more attention to the details of the movement.

Harrison et al. [HRvdP04] performed a series of experiments to measure human sensitivity to changes in length in animation considering the effects of expectation, task interference, increase versus 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 the following: (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. [RPE*05] 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 dynamic 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 the following: (1) The measures may not be valid for motions that are significantly different from the training data; (2) Some types of errors may not be reliably detected (for example, errors occurring within a very short duration of time).

McDonnell et al. [MNO07] performed a series of user studies to understand the effect of character pose update rate on perceptual quality of animation. The pose update rate can be different from the frame rate because a pose can be repeated at multiple frames. Lower number of poses per second (pps) can decrease the resource usage but might cause the animation to look un-smooth. Therefore, it would be helpful to find the minimum pps for which an animation will look smooth to the majority of observers. In their first experiment, they presented a large number of 2-second animations that differed in character type, character clothing and motion type as well as pose update rate, and asked the users whether or not each animation was smooth. For each combination of character type, clothing and motion, they found the pose update rate at which half the participant marked the animation as smooth (the 50% threshold). Their analysis of results shows that while the type and clothing of the character did not have a significant effect on the smoothness perception, the motion type was an important factor.

In the second experiment they examined the effects of three parameters of motion, namely linear velocity, motion complexity and cycle rate. Linear velocity refers to the velocity of the center of mass of the character. In their experiments they used a normal walk with arms by the side motion, and a complex walk, which is the same walk motion with additional activity in the arms, torso and head. In order to generate motion with different cycle rates, they modified the time distance between footsteps and matched the movements of arms, torso and head with the cycle of the legs. Their results show that motions with 1) higher linear velocity or 2) more complexity or 3) higher cyclical speed of limbs require more poses to appear smooth.

They also experimented how the users judge smoothness when multiple characters with different pose update rates are present in the scene. They presented animation in which one character was displayed in the front and several characters in the back. All the characters were performing a normal walk. The test conditions were 1) whether or the movement of the characters are synchronized, 2) the number of characters in the back and 3) the pose update rate of the front character (chosen from values above the minimum threshold). The test variable was the pose update rate of the background characters. For each animation the users were asked if all the characters in the scene appear smooth. Like the previous test, they found the 50% threshold for all different combinations of test conditions. The results show that a small crowd needed a higher pose update rate than medium or large size crowds, but there is no significant difference between medium and large size crowds in terms of smoothness perception. Also the pose update rate of the front character and whether or not the motion is synchronous did not have a significant effect on the minimum threshold.

2.6 Motion Capture Compression

Current approaches to compression of motion data can achieve compression ratios of 25:1 to 35:1 without noticeable perceptual degradation [Ari06, BPvdP07]. However, each approach has some deficiencies in addressing the requirements of online 3D applications as discussed below.

2.6.1 PCA Based Approach

Arikan [Ari06] 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 (figure 2.5). The input database consists of a file describing the skeleton structure of the animated character, plus a (typically large) number of motion clips. In his experiments he 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 author uses joint position data to avoid problems 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).

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 [BPP07]. Second, the 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 a relatively small number of frames the compression ratio can be significantly lower [A06]. This is because high compression is achieved when PCA is applied on large clusters.

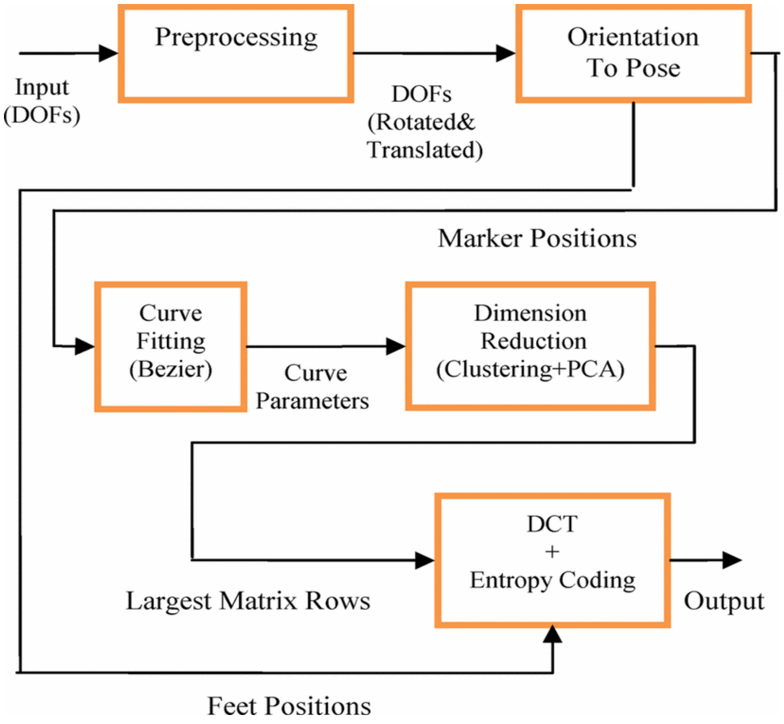


Figure 2.5: PCA Based Approach.

2.6.2 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 (figure 2.6). 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. [BPvdP07] 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.

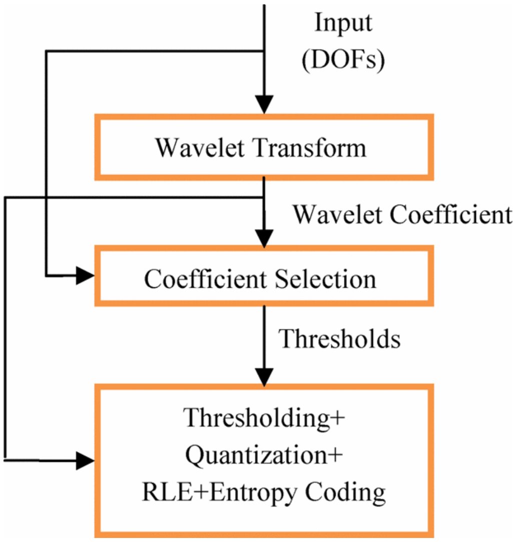


Figure 2.6: Wavelet Based Approach.

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.

2.6.3 MPEG-4 Bone-based Animation

In 2004, Bone-based Animation (BBA) was included in the MPEG-4 standard [ISO04]. In this method 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. [PJAP07] 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.

2.6.4 Other Approaches

Onder et al. [OGO*08] presented two methods to reduce the number of key frames in a motion capture file. The first method involves 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 key frames. 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 key frames 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. [TCA*09] 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 to182:1 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.

2.7 Motion Capture Synthesis

2.7.1 Temporal scaling and alignment

Temporal alignment can be done prior to combining two motions (*e.g.* blending). A time alignment between two motions (which are sampled at the same rate) is a pair of functions $c_1(i)$ and $c_2(i)$ identifying which frame from each motion to pick at time step i. This is usually done by minimizing

$$\left\| X_{c_1(i)}^{(1)} - X_{c_1(i)}^{(2)} \right\| \tag{2.2}$$

for some norm. Kovar and Gleicher [KG03] use dynamic programming to compute such an alignment. Practically, two motions can be aligned by inserting or deleting frames from each one such that the resulting sequences align best.

2.7.2 Blending, transitions and filtering

Motion blending is to create a new motion (called a blend) by calculating the weighted average of two input motions. For better results the input motions are usually time aligned before blending. Very few motions could be blended if the representations of source motions include root positions. This is because even similar motions may vary in places they occur. To solve this problem, blending algorithms usually ignore root positions and keep velocity and angular velocity instead. The root positions of a blend can be found by computing the weighted average of root velocities [PP10]. If the rotation angles are represented by quaternions then the interpolation can be done using spherical linear interpolation (slerp). For two quaternions slerp can be computed as:

$$slerp(q_1, q_2, t) = \frac{\sin(1-t)\theta}{\sin(\theta)} q_1 + \frac{sin(t\theta)}{sin(\theta)} q_2 \tag{2.3}$$

where $\theta = \arccos(q_1 q_2)$ and t is the weight factor. It is also possible to blend more than two sequences, which might result in higher quality motions [BW95], [RCB98].

A transition is a special case of blending in which different activities are linked together in a natural looking (e.g. the slowing pace taken when moving from a run to a walk). Two important parameters of a transition are intervals over which blending will occur and the duration of these intervals. [KG03] propose a method to automatically select good transition points between two motions.

2.7.3 Motion editing

Gleicher [Gle97] proposes a space-time constraint method to interactively edit motions to meet new requirements while preserving the original quality as much as possible. Space-time constraint methods allow the user to define constraints over the entire motion and compute the best motion that meets these constraints using a solver. Users are able to make adjustments to the character with direct manipulation, for example repositioning a character's hand. The animation system tries to preserve the naturalness of the motion by considering the entire motion when making changes. While the output is not guaranteed to look natural, this method enables the motion author to manage constraints and update the process to obtain a natural looking motion. Using parametric warps, Witkin and Popovi [WP95] modified motions so that they pass through user specified keyframes. Similar methods are used by Shin et al. [SKG03] to edit the motion in a way that meets physical constraints (*e.g.* preserving angular momentum when the character is not in contact with the environment).

2.7.4 New motions by cut and paste

New motions can be produced by different body parts of the character in different motions. For example, the new motion might combine the upper body of one motion with the lower body of another one. Ikemoto and Forsyth [IF04] trained a classifier to detect natural looking motions. Based upon a set of rules, a randomized search is performed over the motion collection to find the transplants that are likely to be successful and enrich the existing motion collection. The transplants that are tagged natural by the classifier are added to the database. One limitation of this method is that the classifier might not be reliable when presented with motions that are very different from the training set.

2.7.5 Motion fill-in by nonparametric regression

Pullen and Bregler [PB02] verify the fact that DOFs of a motion are often correlated, so missing DOFs can be predicted from ones that are present. They propose a motion synthesis method that allows the animator to create a motion by sketching a small number of keyframes on a fraction of DOFs. After that the system searches a motion capture database to find best matching sets of motion and computes the missing DOFs of other joints and other frames using a non-parametric regression method. This process might result in multiple motions. Selection of the best motion is up to the animator.

2.7.6 Motion interpolation

Motion interpolation is the process of synthesizing motion by interpolating between or extrapolating from existing motion capture measurements. Some examples of these measurements are momentum, force, and torque. A controller can track measurements and reconstruct motion by controlling some body parameter when measurements are not available. These controllers can be used to adjust the motion when the character interacts with other characters or the environment [FP03], [PB00]. Additionally, these controllers can be helpful in smoothing errors during motion recording.

2.7.7 Simplified characters

Popovic and Witkin [PW99] propose a method for transforming motions trying to preserve its physical properties. They use spacetime constraint dynamics to maintain the naturalness of the original motion while giving the animator as much control as possible. They model muscles as proportional derivative controllers that attempt to drive a DOF to a set point. In the first step, a simplified character is constructed and the motion of the simplified model is fit to the motion capture data. Next, a physical spacetime optimization solution is obtained that include body's mass properties, pose and footprint constraints, muscles and the objective function. New motions can be generated by modifying the constraints, physical parameters of the model and other spacetime optimization parameters (such as footprint positions, gravity, limb geometry). Finally, the motion change of the simplified character is mapped back to the original motion.

2.7.8 Modified physics

Liu and Popovic [LP02] propose a method for fast prototyping of natural looking motion. The system produces realistic animation from a rough initial sketch provided by the animator. The system infers environmental constraints from the initial motion and separates the original motion sequence into several (constrained or unconstrained) stages. Next, a transition is established between constrained and unconstrained animation stages. After that the system generates physical constraints according to Newtonian laws and biomechanics knowledge. Finally, an objective function is constructed that favors smooth motions that are similar to the input motion and balanced when stationary. To produce the animation, the system uses a sequential quadratic programming method that finds the optimal solution satisfying environment, transition pose and the objective function constraints.

2.8 Motion Level of Detail

Ahn and Wohn [AW04] proposed a motion simplification method to minimize the simulation costs of generating crowd animation while maintaining the similarity between the original and simplified motion. They simplify the trajectory of each joint separately by dividing the trajectory into clusters of consecutive joints that satisfy an error threshold and replacing each frame with the key posture of the cluster. The key posture is the posture with minimum sum of error. Their method could simplify the motion by 32% to 98% depending on the complexity of the skeleton and acceptable error. They achieve better results for more complex skeletons (higher number of joints) and monotonous motion (like standing and looking around) as opposed to variable motion like dancing.

Savoye and Meyer [SM08] propose a 3-layer Level of Detail (LoD) for character animation. They define LoD at 1- The skeleton 2- The mesh and 3- The motion level. Skeleton LoD is achieved using a skeleton labeling algorithm and multi-resolution skeleton graph. They define five categories of joints, namely end effector joints, root joints, major joints, minor joints, and extreme joints and propose an algorithm (skeleton labeling) to assign each joint in the skeleton to one of the above categories. For each joint, a metric called "Joint Energy" is defined as well which indicates the importance of the joint in a specific motion. The tree structure of the skeleton is converted into a multi-resolution graph. Each joint in the graph not belonging to root, end-effector or major joints could be enabled or disabled (decimated). The criteria for enabling or disabling joints is a threshold on joint energy (joints with energy below the threshold are disabled).

The motion LoD is provided by defining a similarity metric between consecutive frames of the motion. Based on the distance of the character from the camera, a threshold is defined so that the neighboring frames with similarity scores above the threshold are merged.

CHAPTER THREE: PERCEPTUALLY GUIDED FAST COMPRESSION OF 3-D MOTION CAPTURE DATA

3.1 Introduction

In this chapter we propose 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 [FCB11].

Among the different motion capture data compression methods discussed in chapter two, we found 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 [GW08]. 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.

3.2 Proposed Method

The concept 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 [BPvdP07]) because: (1) Either the compression or decompression would require high processing resources, which is not suitable for the intended application; (2) As discussed in [BPvdP07], 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).

We tried 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 [BPvdP07] "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 applied the Interactivity-Stimulus-Attention Model (ISAM) [LTG*07], 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 ($c_i, 1 \le i \le N$) 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 ($1 \le k \le M$) 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:

$$v_i = \frac{\sum_{p=m_i*W}^{m_i*W+W-1} \left|c_{i,p+1} - c_{i,p}\right|}{W} \tag{3.1}$$

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.

$$A_i = l_i v_i \tag{3.2}$$

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

3.3 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 [CMU] in our user studies. LZW compression [LD04] 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 56x4x(number of frames) bytes.

We performed a user study using two short motion sequences from the CMU [CMU] 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 users participated in this user study. All the users were studying computing science at the University of Alberta. Users were able to compare the original and compressed motions side by side (Figure 3.1). Compressed motions were shuffled in random order before presenting to the users. The users were asked to rank the quality of each compressed motion from 1 to 5 (5 being the best quality) with a step of 0.5 (*i.e.,* score 4.5 was acceptable). 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.

Table 3.1 reports the user study results: The average quality score, standard deviation, and confidence interval ($\alpha = 0.05$) of the compressed motions using the proposed method, with compression ratios from 20:1 to 40:1 for QP=9 and window size of 256 frames. 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.

Table 3.2 compares the quality scores of the proposed method with standard wavelet compression using 1-tailed T-test. For compression ratios over 30:1 our method shows a statistically significant improvement over the standard wavelet compression.

Table 3.1: Results of user study on two of CMU database motion files.

Motion File	Compression Ratio	Average Quality Score	Standard Deviation	Confidence Interval
2_2	20:1	4.705	0.437	0.238
2_2	25:1	4.545	0.450	0.245
2_2	30:1	4.159	0.525	0.285
2_2	35:1	3.886	0.444	0.241
2_2	40:1	3.727	0.445	0.242
49_14	20:1	4.885	0.288	0.199
49_14	25:1	4.615	0.487	0.337
49_14	30:1	4.115	0.211	0.146
49_14	35:1	3.923	0.385	0.267
49_14	40:1	3.404	0.455	0.315

Table 3.2: Comparing the proposed method and the standard wavelet compression. The third and fourth columns show average quality scores.

Motion File	Compression Ratio	Proposed Method	Standard Wavelet	P-value
2_2	30:1	4.159	3.773	0.053
2_2	35:1	3.886	3.182	0.016
2_2	40:1	3.727	2.364	1.07E-06
49_14	30:1	4.115	3.692	0.005
49_14	35:1	3.923	2.962	0.004
49_14	40:1	3.404	2.308	0.008

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 (Table 3.3), using a 1-tailed T-test. Here again the results show a significant improvement when both factors are considered.

The compression times of three motion files with different number of frames are shown in Table 3.4. 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 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 μs computed based on relatively long motion files, which demonstrates that the algorithm has very efficient processing time.

Table 3.3: Verifying the efficiency of selected perceptual factors. The third and fourth columns are the average quality scores.

Motion File	Compression Ratio	Proposed Method	Bone Lengths Only	P-value
2_2	30:1	4.159	3.568	0.010
2_2	35:1	3.886	3.182	0.016
2_2	40:1	3.727	3.227	0.022
49_14	30:1	4.115	3.538	0.005
49_14	35:1	3.923	3.308	0.004
49_14	40:1	3.404	2.769	0.008

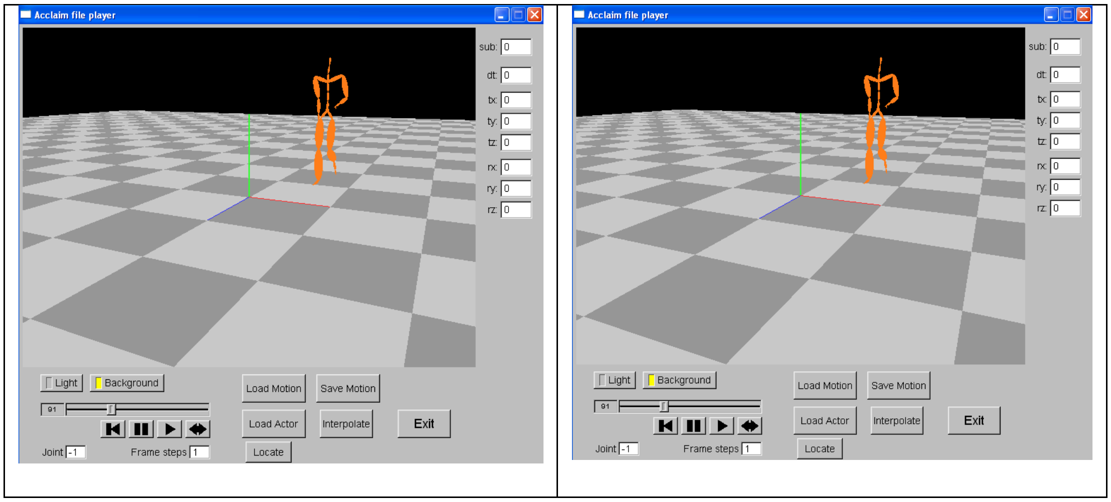


Figure 3.1: Motion Capture Viewer Environment.

Table 3.4: 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

3.4 Discussion

It is reported by the adaptive wavelet compression method [BPvdP07] 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 Table 3.1 for the same compression ratio. The compression time reported in that paper is between 133 and 483 milliseconds per frame on a 2 GHz AMD Athlon 64 bit which is approximately 10^4 times greater than our method. In Arikan's paper [Ari06], 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.

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.

CHAPTER FOUR: EFFICIENT COMPRESSION OF RHYTHMIC MOTION USING SPATIAL SEGMENTATION AND TEMPORAL BLENDING

4.1 Introduction

Human motion data is used in a wide variety of applications including interactive games, virtual environments and computer aided training systems, such as dance and rehabilitation. Efficient compression of motion data can save a significant amount of bandwidth, especially in the presence of other competing media data. While several motion capture compression methods have been discussed in the literature [Ari06,BPvdP07,FCB11,FCB13,GPD09], very few take full advantage of high spatial and temporal correlation in rhythmic motion data to achieve more effective compression.

Motion capture is a process to measure the configuration (spatial) of a human body in a given time frame (temporal) [FAI*05]. Motion sequences can be recorded using optical, mechanical or magnetic devices by tracking the movements of key points (such as joints) on an object. Human motion data show a considerable amount of spatial and temporal correlation, especially in rhythmic procedural movements, which can be exploited for effective compression.

Using a collection of featured motion primitives can contribute to higher efficiency. Gu et al. [GPD09] first proposed dictionary based compression of motion data. They compressed the movement of each marker separately using a database of primitives. An average compression ratio of 25:1 was reported. Firouzmanesh et al. [FCB13] improved Gu's technique [GPD09] based on efficient adaptation of bandwidth and human perceptual factors. A compression ratio of 50:1 was reported in [FCB13] without noticeable visual degradation.

Rhythmic movements can be synthesized by using a series of example motions associated with input music. Shiratori and Ikeuchi [SI08] focus on three key features of input music: rhythm, speed and mood, to generate representative dance sequences. For each of these features, they have developed corresponding analysis and synthesis methods. Based on observation of human dance, the authors hypothesize that the correspondence between movement and rhythm is captured by key poses – brief pauses in movement synchronize the motion rhythm with the musical rhythm. The motion analysis step is based on the speed of the performer's hands, feet and center of mass. Key poses are extracted according to two criteria: First, dancers clearly stop their movements during a key pose. Second, dancers clearly recompose their body parts between consecutive key poses.

To explore the music-driven approach, Sauer and Yang [SY09] create a system which generates dance animation based on the attributes of a piece of music. They use script files to identify the movements and timing involved in the performance. By fine-tuning the script, the user has easy control of the animated characters and can create dance animations that are appealing to the viewers. As the first step beat and dynamics are extracted from the input music file. Next, music information is used in conjunction with user-defined motion routines. Five components come together to synthesize the dance. (1) Primitive Movements: twenty four movement primitives are implemented which can form combinations to create complex and interesting motions. These include heel clicks, hops, stomps, leg lifts, knee bends, and leg raises. (2) Script File: A text file to list primitives and routines that the user wants performed in the desired animation. (3) Mappings: In generating the animation, an algorithm maps the primitives defined in the script file to the beats determined through beat detection. (4) Constraints: Constraints keep track of foot positions and synchronize body components' movement order, producing realistic animation. (5) Routines: A series of movement primitives are combined to create a more complex set of motions called a routine.

Currently, Sauer and Yang's system is limited to producing Celtic dance. In Celtic dance, only the dancer's legs move; their arms remain on the sides and the torso stays upright. Dancers do not interact or dance collaboratively, and usually execute their routines individually while standing in a line. Although Sauer and Yang's system could be expanded to handle some other dance primitives, there are design constrains. For example, the system is not able to incorporate paired dancing, where synchronization is required between dancers' movements, in addition to individual dancer's independent movements. Incorporating arm, torso, and full body movements further complicates the system. Currently, their dance generation algorithm can only process foot and leg movements typical of Celtic dance; the system only needs to ensure that each leg is performing one move at a time, and track which foot is in front of the other when initializing motions based on the current "leading foot." Other forms of dance involve more complicated movements, such as side-stepping in addition to forward and backward steps, movement of the dancer around the floor rather than staying in place, arm movements, full body and torso movements, and so on.

Motivated by the music-driven approach [SY09] and its deficiency, in this work we propose a framework using spatial and temporal blending to achieve highly efficient compression of rhythmic motion. The rest of this paper is organized as follows: Section 4.2 presents our proposed method. Section 4.3 reports experimental results. Section 4.4 discusses limitations and future work, and concluding comments are given in Section 4.5.

4.2 Proposed Method

To the best of our knowledge, constructing a spatial temporal hierarchy in a time series to perform motion segmentation and blending, to achieve high compression rate and rendered quality is novel. Figure 4.1 illustrates the processing pipeline of the proposed approach.

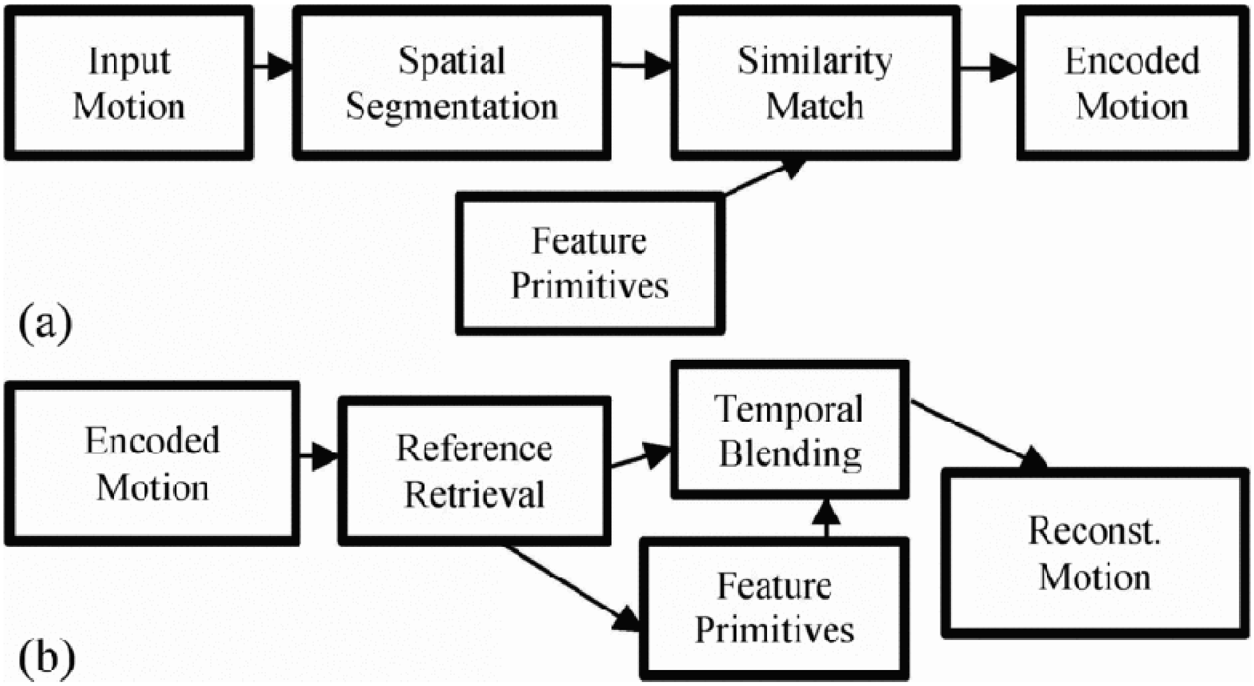


Figure 4.1: Processing pipeline of the proposed method: (a) encoding and (b) reconstruction.

The feature primitive database contains the basic moves (primitives) of the different rhythmic sequences that need to be compressed. Depending on the application the size of the database varies. The database could be created from a set of sample motions. The primitives are generated by segmenting the sample motions either manually or using some automatic method [SI08]. The compression is done by encoding the segmented motion by replacing each segment with a reference to the best matching primitive in the database. Decoding is done by blending the referenced primitives in a way that the output is perceptually appealing without noticeable artifacts.

4.2.1 Spatial Segmentation

Using a hierarchical biped (bones) structure, the motion state of an object can be defined by the position of the root marker in the world coordinates, plus the relative orientation of each marker in its parent's coordinate system. Joints are commonly used as marker positions but other key positions can also be used. The orientation data is converted to the quaternion space. The input motion data at state T can thus be represented using a set of motion signals $m(i,T) = \left(p(i,T), q_1(i,T), \dots, q_j(i,T)\right)$ where $p(i,T) \in R^3$ represents the root position and $q_1(i,T), \dots, q_j(i,T)$ are quaternions representing the orientations of the other markers. This way the animation states, composed of feature poses at given time frames, are hierarchically integrated into a time series of motion segments.

4.2.2 Temporal Blending

Different motion segments end at very different poses. In order for the reconstructed animation to look natural, it is necessary that subsequent motion segments maintain smooth transition from the previous. Our approach synchronizes key positions and ensures smooth transition between motion segments for a variety of body movements. We achieve this by using spherical linear interpolation (SLERP) of quaternions [PP10] in conjunction with distance mapping [Slo07]. The root position is adjusted by repositioning the root between transitions in the time series, accounting for changes in its X and Z coordinates, and recalculating based on the root's velocity from one motion to the next.

All bones branch down in some configuration from the root, which does not have a parent. Root joint positions are blended by computing the weighted average of root adjusted positions:

$$p_{blend} = \omega_1 p_1 + \omega_2 p_2 + \cdots + \omega_n p_n \tag{4.1}$$

where ω_i is the weight of each motion, and p_i is the position adjusted by velocity. In this process, joint orientations of each position are represented as non-singular unit quaternions. Slerp [PP10] is computed as:

$$slerp(q_1, q_2, t) = \frac{\sin(1-t)\theta}{\sin(\theta)} q_1 + \frac{sin(t\theta)}{sin(\theta)} q_2 \tag{4.2}$$

where $\theta = arccos(q_1 q_2)$ and *t* is the interpolation weight.

In order to achieve smooth transition, the blending needs to start and end at a point where the animation motions to be combined are most similar. We apply distance mapping to achieve this. A distance map is implemented as a 2-d array of distances between the frames in the two animation sequences. Distances between each pair of frames are calculated as:

$$D\big(F_{dst}(i), F_{src}(j)\big) = \sum_{k=1}^{n} \left\| p_{k,j} - p_{k,i} \right\|^2 \tag{4.3}$$

where $\mathrm{p}_{\mathrm{k,j}}$ and $\mathrm{p}_{\mathrm{k,i}}$ are the position of the k^{th} joint in frames *i* and *j*. An example of a distance map is shown in figure 4.2. The frames with minimum distance are chosen as the start and end points for blending. In this example, (a) is chosen because the black diagonal indicates zero distance between the two frames (most similar). In the current implementation, our distance calculation does not account for the weighting of bones. We expect that better results can be achieved if different weights are assigned to different key positions considering factors like bone length/volume and bone density.

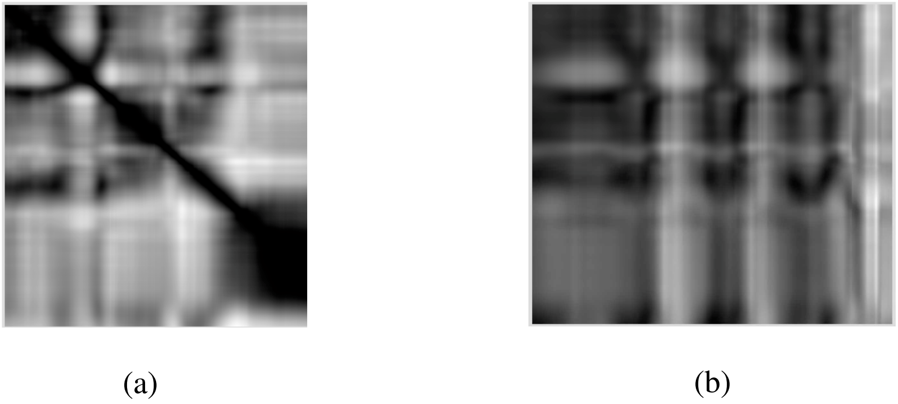


Figure 4.2: Examples of distance maps. The distance is represented by pixel intensity. A black pixel means zero distance. (a) A distance map comparing two identical animations. Notice how the diagonal is completely black. (b) A distance map comparing two different animations.

Our time performance is better than the distance mapping approach in [Slo07] because in order to maintain smooth transition between two animations, we only need to perform blending over a small number of frames; currently we use about one quarter of the frames in each segment.

4.3 Experimental Settings and Results

We used the input motion capture files in the CMU database [CMU] in AMC format. The biped structure has 30 joints with a total of 56 Degrees of Freedom (DOFs) (53 rotational and 3 translational). The motions were sampled at 120 frames per second (fps). We calculate the compression ratio assuming each DOF in each frame is stored as a 4 byte floating point number. The compression ratio depends on the average number of frames in each segment. It will be more efficient if the primitive database already exists on the client side. For typical rhythmic motion the average segment size is between 15 and 30 frames. Each segment is replaced with a reference to the database (4 bytes). Suppose a given animation consists of 20 primitives and each primitive is 20 frames on average. So the raw database takes 20*20*56*4=89 KBs. If we apply wavelet compression [BPvdP07] and compress the data at the rate of 10:1 (which is at a very low error level) the database would be about 9KBs. Table 4.1 presents the estimated compression ratios for animations of different durations (from 3000 to 7500 frames). Note that our compression ratio improves as the number of frames increases.

Table 4.1: Compression ratios of typical rhythmic motions using our proposed method.

No. of frames	3000	4500	6000	7500
Compression Ratio (est.)	67:1	89:1	134:1	168:1

Current compression rates reported in the literature are between 25:1 and 75:1 for the same type of data. This shows that our framework can make more effective use of the available bandwidth.

In order to examine the visual quality of our proposed blending method, we designed three experiments (see Figure 4.3, the video recordings are in the supplementary material). The first video demonstrates a simple combination of animation files, emulating what a choreographer might put together when designing a dance. The animated figure takes a few steps forward which transits into a twirl motion. During the transition, one can subtly see the limbs sliding into position in preparation for the twirl motion. After the second twirl, the figure's feet change position. This interpolation successfully corrects the artifact adversely created by twirling large bones in the animation; where the figure's weight shifts to its right leg as the left leg comes forward. In the subsequent transition the figure spins in place counter-clockwise as a result of a drastic change in orientation between motions. The interpolation successfully generates a smooth turning of the figure's face.

The second video tests interpolation between drastic changes in arm positions. The first motion ends with the figure's left arm raised, and the second motion begins with the figure's arms lowered. The interpolation between motions successfully moves the arms between positions in a natural way. Later, interpolation occurs between a motion ending with arms down, and a motion beginning with arms up. This transition is again smooth and successful.

The third video shows a figure covering a large distance during its dance. Interpolations successfully maintain the figure's momentum as it travels.

4.4 Discussion and Future Work

In the current implementation, we use Euclidean distance when calculating the distance between pairs of corresponding positions in two frames. No weighting criteria are used in [Slo07] either. However, based on our experimental observation, we believe that minor bones, such as fingers, should carry less importance than larger bones, such as those representing the upper and lower legs and arms. Thus, in future work we will take into account the length/volume, density, or weight of each bone when calculating the distance between two segments. However, choosing the appropriate weight based on each bone’s physical property is tedious. We note that minor bones tend to occur in the extremities, and thus it is worth considering a propagation strategy to automatically assign weights from the root to the leaves in the bones hierarchy, assuming that a small motion in a large bone will likely create noticeable movements in its progressively smaller child bones, but movements in small bones will not affect their larger parent bones. Although we use rhythmic dance sequences for illustration in this work, our technique can be applied to motion capture data in general, and is most effective for sequences of long duration.

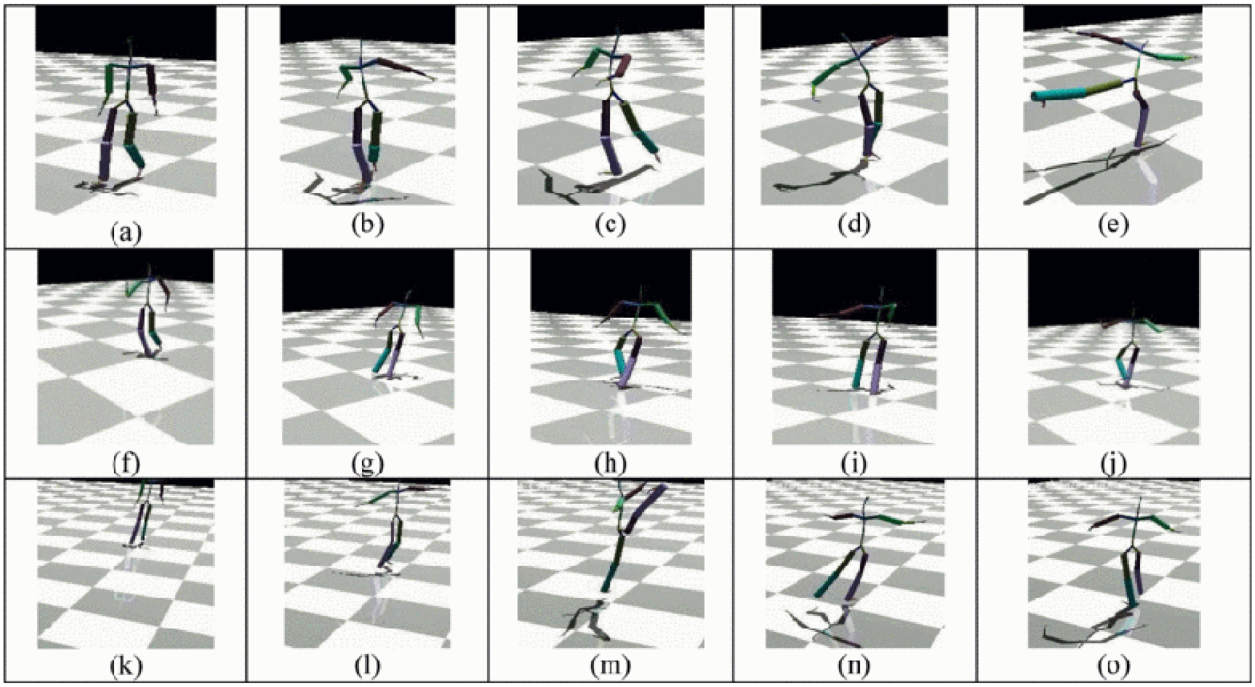


Figure 4.3: (a)-(e) Screenshots the first video. The figure takes a few steps forward (a) and transits into a twirl motion (b), (c); After the second twirl (d), the figure's feet change position .(f)-(i) Screenshots of the second video; The first motion (f) ends with the figure's left arm raised, and the second motion (g) begins with the figure's arms lowered. Later, interpolation occurs between a motion ending with arms down (h), and a motion beginning with arms up (i). (k)-(o) Screenshots of the third video.

CHAPTER FIVE: PERCEPTUALLY MOTIVATED REAL-TIME COMPRESSION OF MOTION DATA ENHANCED BY INCREMENTAL ENCODING AND PARAMETER TUNING

5.1 Introduction

Using human motion capture data has become very common in a variety of applications including interactive games, virtual environments and movie production. In online applications the bandwidth that needs to be dedicated to motion capture data can be considerable especially when multiple dynamic characters are present in the scene. Like images and videos, motion capture data show a significant amount of spatial and temporal correlation that can be exploited to effectively reduce the data size. In this research we focus on lossy compression of rhythmic human motion capture data taking human perception into consideration.

While there are several works in the literature that address compression of motion capture data in general (as discussed in the next section), very few take advantage of the repetitive nature of rhythmic motions to achieve a higher compression rate. Gu et al. [GPD09] address this problem by creating a database of primitives for each motion and replacing each segment of the motion with its closest match in the database. However, their method has several drawbacks: First, since a separate database is created and transmitted for each motion file, the compression cannot be done in real time, or with a low delay. Also, their approach ignores the error between the closest primitive and the original segment, which may result in a significant loss of visual quality.

Limitations of the human visual system have been used to significantly compress images and video, in which the human observer notices little or no degradation in the quality of the compressed data. However, measuring the perceptual quality of motion is still in its early research stage (as discussed in Section 2.4). For this reason we performed a user study to measure the sensitivity of human observers to the errors in the joint (key point) rotations generated from the compression algorithms.

In this chapter we propose a method for fast compression of rhythmic motion taking perceptual factors into consideration. Our main contributions include:

a) Incorporating perceptual factors in animation: Perceptual factors have been extensively used to achieve higher compression in images and videos. However, measuring perceptual quality of animation is still in its early stage and there are very few studies in the literature (like [FCB11]) that try to exploit perceptual factors for higher compression. In this work, we propose a method to measure the sensitivity of human subjects to errors in the rotation angles at the joints of the models and effectively integrate the finding in our compression process.

b) Using an existing database of primitives to achieve higher compression: While [GPD09] first proposed creating and using a primitive database for compression, in their method a specific data-base is created and transmitted for each motion capture file. Instead, we use a pre-existing primitive database for faster compression to achieve real time performance.

c) Easy bandwidth and quality adaptation: The proposed method allows bandwidth adaptation by adjusting the acceptable error in joint rotations. Using wavelet-based compression methodology, this adjustment process is translated to selecting a percentage of wavelet coefficients in a way that the error falls below a predefined threshold.

5.2 Background and related work

While Arikan's method [Ari06] (as discuss in chapter 2) can significantly reduce the size of large motion capture databases treated as a single sequence, the compression ratio is not as high when it is applied to individual motion clips of regular length. This is because when PCA is applied on smaller clusters the compression ratio is not as high.

Lie et al. [LM06] applied PCA and key frame reduction using splines on subsequences of the motion to achieve higher compression ratio. The motion is segmented appropriately so that each subsequence can be reconstructed with high precision by just using a small portion of the PCA matrix rows. Their method could compress a motion file containing the global position of markers at a rate between 18:1 and 61:1 depending on the complexity of the motion.

The main drawback of the Baudin et al.'s wavelet based approach [BPvdP07] is that compression is computational expensive because of the global optimization process. In chapter 3 we proposed a wavelet based approach with the goal of achieving the same compression ratio with the same perceived quality without consuming time on global optimization. While the results are promising, the effect of other parameters including distance from camera, velocity of the object and the size of the limbs still needs to be studied. Also, the high compression ratio is achieved only on relatively large segment size (256 frames) creating long transmission delay.

Gu et al. [GPD09] compressed the movement of each marker separately using a database of primitives. The trajectory of each marker is partitioned into smaller segments. The chopping points correspond to sharp changes in the motion. Each segment is normalized to a fixed number of frames. The K-Means algorithm is applied to the motion segments and the center of each cluster is considered a primitive. Compression is done by replacing each motion segment by its closest primitive. Since each database has a fixed size and for each motion a database needs to be transmitted, the compression efficiency decreases for smaller motion sequences due to the higher number of databases.

Tournier et al. [TCA*09] uses Principal Geodesics Analysis (PGA) along with inverse kinematics to exploit both temporal and spatial redundancies of the motion. PGA is the extension of PCA in non-Euclidean space. While the reported compression ratio is higher than many other methods the decompression process is very time consuming because of the inverse kinematics. Furthermore, perceptual tolerance has not been adequately explored in related work.

5.3 Proposed Method

Using a motion primitive database facilitates the exploitation of temporal redundancies in repetitive and/or rhythmic motion. Figure 5.1 presents overview of the proposed method. While databases have been used in related work [GPD09], there are a number of important issues that need to be resolved before previous approaches can be applied to real-time dynamic motions: (1) In an interactive real-time setting, the spontaneous motions initiated by the user cannot be predicted beforehand. So a complete pre-defined database is not available and primitives have to be created on-the-fly. Thus, (2) in order to minimize the online processing time, we divide the compression process into offline and online phases. In the offline phase a reference primitive database is created based on a set of training motion data. A reference database copy is stored at the rendering site. (3) In the online phase, the motion is compressed segment by segment. For each segment the reference to the closest primitive in the database is transmitted. If the difference between the segment and its closest database primitive is above a certain visual threshold, we employ incremental encoding to transmit the compressed difference. Details on visual threshold are given in Section 5.3.5.

5.3.1 Input Data

Motion capture data is produced by tracking the position of a series of key points at different parts of a subject's body. Using a hierarchical biped structure the motion can be represented by the position of the root marker in the world coordinates, plus the orientation of each marker in its parent's coordinate system. The input motion data consists of a set of motion signals $m(i) = \left(p(i), q_1(i), \dots, q_j(i)\right)$ where $p(i) \in R^3$ represents the root position and $q_1(i), \dots, q_j(i)$ are quaternions representing the orientations of the markers.

5.3.2 Motion Segmentation

The goal of the motion segmentation part is to divide the movements of each joint into clips of approximately similar length. There are several methods in the literature for this purpose. One of the most commonly used approach is based on the second derivative or angular acceleration of the motion (for example see [Bin00], [Zha01], [BSP*04]). Gu et al. [GPD09] define a metric to measure the smoothness of motion change and chop the motion at points where the metric is above a predefined threshold. Support Vector Machine classifiers are employed by [LFA*10] to decompose motions into a set of primitives in a way that each motion can be reconstructed by a linear combination of primitives. Barbic et al. [BSP04] use PCA to segment motion into distinct activities. Their methods are based on the assumption that intrinsic dimensionality of motions representing one activity is much lower than that of combinations of two or more different activities. The chopping points are defined where there is a sudden change in the intrinsic PCA dimensionality of the motion.

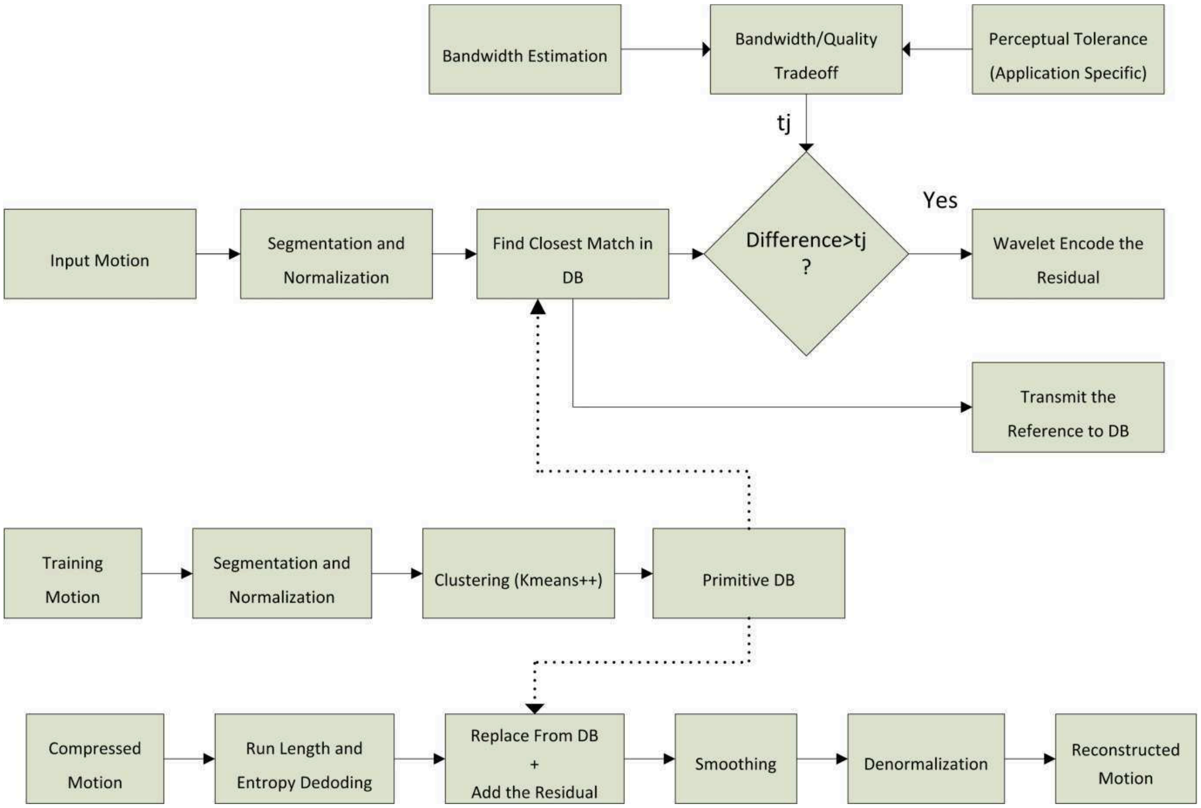

Bandwidth Estimation
Bandwidth/Quality Tradeoff
Perceptual Tolerance (Application Specific)
tj
Yes
Input Motion
Segmentation and Normalization
Find Closest Match in DB
Difference>tj ?
Wavelet Encode the Residual
Transmit the Reference to DB
Training Motion
Segmentation and Normalization
Clustering (Kmeans++)
Primitive DB
Compressed Motion
Run Length and Entropy Dedoding
Replace From DB + Add the Residual
Smoothing
Denormalization
Reconstructed Motion

Figure 5.1: Overview of the proposed compression and decompression method.

We found that acceleration based methods are most suitable for the purpose of this research. PCA based methods were shown to be effective for dividing the motion into separate activities (e.g., running, dancing, jumping, etc.). However, since our main focus is compression of rhythmic motion we try to segment the motion in such a way that different segments correspond to different motion beats.

Similar to [KPS03] we use zero-crossings of the angular acceleration of each joint as the chopping points. In quaternion space the angular velocity can be approximated as:

$$\omega_j(i) = 2\log\left(q_j^{-1}(i-1)q_j(i)\right) \tag{5.1}$$

where q is the quaternion representation of the joint orientation and h is the time interval. The angular acceleration can be approximated by:

$$\alpha_j(i) = \frac{\omega_j(i) - \omega_j(i-1)}{h} \tag{5.2}$$

In order to make sure that each segment is long enough we choose the first zero-crossing that is at least 15 frames away from the start of a segment, as the chopping point. In order to simplify the comparison of segments with various primitives, a normalization process is applied to segments to make them the same length (number of frames). Since in the later stage we use multi-level wavelet compression which requires the input signal to be a power of two, all the segments are normalized to 16 frames. New frames are generated using Spherical Linear Interpolation (SLERP).

5.3.3 Database Creation

In order to create the primitive database we use a series of training motion data. Training data are chosen in a way that they contain all (or most) of the basic movements of the desired motion types. The motion of each joint is segmented and normalized to create input segments. A variation of K-MEANS++ [AV07] clustering in quaternion space is applied to input segments to create primitives. In order to find the best number of clusters (K) we start from an initial K and increase it until the clustering error falls below a predefined threshold or the maximum number of clusters is reached. We measure the distance between two quaternions as:

$$d(q_1, q_2) = cos^{-1}(2(q_1.q_2)^2 - 1) \tag{5.3}$$

Where the dot product is the inner product of $q_1$and q_2 ($q_1$and q_2 are normalized quaternions). The distance between two normalized segments is defined as:

$$(5.4) \qquad D(Q_1, Q_2) = \max\big(Q_1(i), Q_2(i)\big)\ 1 \le i \le N$$

Where *N* is the number of frames (16 in our implementation). In order to find the center of each cluster we use SLERP as follows: Assuming that M motion segments belong to a cluster, when a new segment is to be added to the cluster, we give a weight of 1 to the new segment and *M* to the current cluster center. For each cluster, error is defined as:

$$e(C_k) = \max\big(D(Q_i, C_k)\big)\ 1 \le j \le M_k, 1 \le k \le K \tag{5.5}$$

where C_k the center of the *k-th* cluster. The clustering error is defined as:

$$\max\big(e(C_k)\big)\, 1 \le k \le K \tag{5.6}$$

The reference database is created by using a training set which contains a number of example motions. The compression can be higher if the transmitted motions resemble the training set motions. However, as can be seen in our experiments our approach is effective even if the motions to be compressed vary from the database training set. We achieve this efficiency by generating additional combinations based on the training primitives. The motion of each joint is segmented and normalized to create input segments. A variation of K-MEANS++ [AV07] clustering in quaternion space is applied to input segments to create primitives. The distance between two segments is defined as the maximum distance between pairs of corresponding quaternions. In order to find the best number of clusters (K) we start from an initial K (5 in our experiments) and increase it until the clustering error falls below a predefined threshold or the maximum number of clusters is reached. These two values are 0.01 and 25 in the current implementation as described in the Experimental Setting Section.

5.3.4 Compression and Decompression

Having a database of primitives facilitates higher compression of motions of any length. In order to compress the motion each segment is replaced with its closest primitive in the database. If the error is above a threshold (t_j) then we encode the difference between the primitive and current segment using wavelet transform. This threshold (t_j) is defined separately for each joint and can be found by measuring the sensitivity of human observers to the maximum error in each joint. In the next part we describe a method based on two alternative forced choice (2AFC) [SB94] to find t_j such that most of the human observers do not notice a degradation after compression. The difference between two quaternions can be defined as:

$$diff(q_1, q_2) = q_1^{-1} q_2 \tag{5.7}$$

If $q_1$and $diff(q_1, q_2)$ are known, q_2 can be reconstructed as:

$$q_2 = q_1 diff(q_1, q_2) \tag{5.8}$$

The difference between two segments is the difference between their corresponding quaternions. In order to encode the difference between the current segment and its closest primitive we apply a 4-level CDF 9/7 wavelet transform [CDF92] to the difference. The reason we use CDF 9/7 is its high performance in the compression of images as used in the JPEG2000 standard [SCE01]. We quantize the wavelet coefficients and set the coefficients to zero if their absolute value is below a certain threshold. This threshold is selected in a way that the distance between the reconstructed and the original segment falls below t_j. The output stream is produced by applying run-length encoding and entropy coding. For each segment the decoder receives the number for closest primitives and wavelet encoded difference (when the error is significant).

A smoothing procedure can be applied to consecutive segments to make the output more visually appealing. The procedure for segments $Q_1$and Q_2 can be done as follows:

1-$A = \{Q_1(N - M), \ldots, Q_1(N)\}, B = \{Q_2(1), \ldots, Q_1(M)\}$

2-$(x, y) = argmin\big(A(i), B(j)\big)\ 1 \leq i, j \leq M$

3- Replace $Q_1(k)\ x \leq k \leq N$ with

$$slerp(Q_1(x), Q_2(y), \frac{k}{x+y})$$

4- Replace $Q_2(k)\ 1 \leq k \leq y$ with

$$slerp(Q_1(x), Q_2(y), \frac{k+x}{x+y})$$

5.3.5 Parameter Values Selection

The most important parameter affecting the compression ratio and quality is the maximum error at each joint (t_j). This threshold can be measured by user experiments. However, measuring t_j for each joint is computationally expensive, and redundant if visually insignificant. Note that human viewers tend to be more sensitive to errors in the body parts in contact with a surface (feet in most cases) [Ari06]. The dependency between the joints in the upper body and lower body is low and the position of feet is mostly determined by the rotation angles of lower body joints. For this reason, we define two visually significant thresholds: one for the lower body

(t_L) and one for the upper body joints (t_U). After experimenting with several different proportions between t_L and t_U, we achieved best results (in terms of quality and compression) by setting t_U = 2 x t_L. In order to find t_L we performed a user study following the 2-alternative forced choice (2AFC) methodology using the two-up-one-down staircase algorithm [CB06]. Distorted motions were generated by setting t_j to t_L for all the lower body joints and to t_U for all the upper body joints. Our proposed compression and decompression methods are then applied.

Table 5.1: Compression ratio (CR) of the proposed method for $t_L = 0.01$(94_2: Salsa dance, 61_4: Indian dance, 79_17: Violin playing, 79_44: Washing a window, 135_4: Front kick, 138_2: Marching, 90_2: Cartwheel) with an average of 48:1.

Motion	94_2	61_4	79_17	79_44	135_4	138_2	90_2
CR	62:1	51:1	52:1	50:1	40:1	44:1	38:1

Figure 5.1: Screen shots of the seven regular length motion sequences used in our experiments.

5.4 Experimental Setting and Analysis of Results

We implemented the proposed method using the Microsoft Visual C++ 2008 platform. The input motion capture files were from the CMU database [CMU] in AMC format. The skeleton has 30 joints with a total of 56 Degrees of Freedom (DOFs) (53 rotational and 3 translational). Processing times are measured by running the program on a desktop computer with an Intel Core 2Duo 2.66-GHz processor and 2 GB of memory. For each type of motion the number of primitives in the database for each joint was limited to 25. Each wavelet coefficient was quantized to 10 bits using uniform quantization. The 53 rotational DOFs were converted to 30 quaternions (equal to the number of joints) and compressed using the proposed method. All the segments were normalized to 16 frames. The three translational DOFs were divided into segments of 32 frames. The Cohen-Daubechies-Feauveau (CDF) 9/7 algorithm was applied to each segment and the smallest 50% coefficients in absolute value were set to zero. The coefficients were then quantized to 10 bits. Run-length and entropy encoding were applied.

In the study to predict human perceptual sensitivity towards reconstruction error, participants were required to complete a series of comparisons where motions were displayed side by side. In each pair, one of the motions was always the original and the other was a distorted motion. The order (whether to show the original motion on the left or right) was chosen randomly. We followed the procedure described in [Gar98] (where the author had extensive analysis on how to perform the staircase test) and started with a high distortion error value of t_L (0.024) and decreased t_L by 0.002 whenever the participant chose the correct (original) motion twice in a row. For a wrong answer, t_L was increased by 0.002. Participants were asked to choose the more visually appealing and smooth motion. If they were unsure they had to select one randomly. The converging point was set as the average of the last three reversals points [Gar98]. Sequences from five different motions (namely Salsa dance, Indian dance, walking, washing windows, front kick) were alternatively used in the test cases. Participants sat at a viewing distance of about 50 centimeters in front of a 19” wide screen monitor with resolution of 1440x900. The study was performed in a room with indoor incandescent lighting. There were 20 participants and each took between 20 to 25 minutes to complete the tests. All the participants were computer science graduate or undergraduate students. The average converging threshold was $t_L = 0.012$, while the smallest was $t_L = 0.01$ and the largest was $t_L = 0.016$.

We created a reference database using three training motion files, namely 61_04 (Salsa dance), 94_01 (Indian dance) and 2_2 (simple walk) from the CMU database. In order to evaluate our compression ratios, we compressed seven different motions (Figure 5.1), including *Salsa* and *Indian dance,* which were used in the training set. We also used other motions: *violin playing*, *washing window*, *Karate kick*, *cart wheel* and *marching* which were not included in the training set (Table 5.1). To compare with other methods, we use the performance reported in the literature (Table 5.2). Although the performance of Optimized wavelet [BPvdP07] is comparable to our approach, they achieve this result by compressing large sequences. The compression ratio would be lower on regular size sequences when each is compressed separately. For the violin playing motion, the compression ratio could reach 50:1 in [GPD09] when the uncompressed data size exceeds 50 MB. However, for smaller size (less than 20MB) a compression ratio of 25:1 is reported. For similar motion, our compression rate can be maintained at 52:1 even when the data size is less than 1 MB. To summarize comparison results show that our proposed method outperforms others on regular size sequences and is more suitable for compressing dynamic motions in real-time applications.

Table 5.2: Compression ratio (CR) of the current methods reported in the literature.

Method	[Ari06]	[BPV07]	[GPD09]	gZip
CR	30:1	48:1	25:1	3.3:1

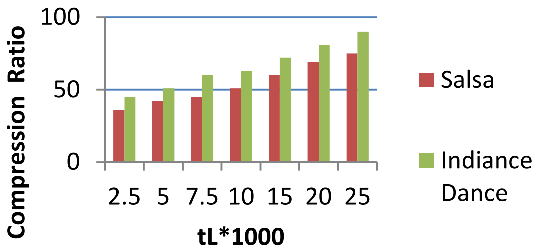


Figure 5.2: The Compression ratio can be increased or decreased by adjusting a single parameter value t_L.

Given the available bandwidth, t_L can be adjusted accordingly. Figure 5.2 shows the effect on compression ratio by varying the parameter t_L from 0.0025 to 0.025 with 0.012 being the average Just-Noticeable-Difference (JND) threshold obtained from our user study. By increasing t_L to the right of 0.012 (high compression), more viewers will be aware of the distortions. One advantage of our method is the capability to incorporate the JND model [CYB12], enabling applications to estimate the average percentage of viewers who will be satisfied with the rendered motions given a compression rate or t_L value.

Table 5.3 presents the compression and decompression time of the proposed method using default parameters and $t_L = 0.01$. The compression time is for online processing which is a deciding factor to support real-time applications and our results demonstrate such performance. The compression time reported for optimized wavelet coefficient selection method [BPV07] is between 133 and 483 *ms* per frame on a 2-GHz AMD Athlon 64 bit, depending on the number of frames in the animation, showing that our proposed method is more efficient in processor usage.

The time for database creation (offline phase) varies greatly depending on the choice of training motions. Motions 61_06 and 61_09 are Salsa dance and motions 94_01 and 94_5 are

Indian dance. The decompression time per frame is quite short, making our algorithm suitable for a wide range of devices with regular processing power.

Table 5.3: Compression and decompression time of different motions for $t_L = 0.01$.

Motion	#Frames	Comp. time Per frame (μs)	Decomp. Time Per frame (μs)
61_06	1772	2140	110
61_09	2575	2370	140
94_01	3602	1710	130
94_05	5272	1660	120

The bandwidth and storage required to transmit the primitive database are low. Using a maximum of 30 primitives per joint for each training motion, the compressed database occupies about 280 Kilo Bytes (KB).

Table 5.4: Compression ratio does not decrease significantly by limiting the maximum number of frames in each segment showing the effectiveness of our method on regular length motion sequences.

S_M	16	32	64
CR (61_06)	46	49	50
CR (91_04)	58	61	63

An important consideration in real-time applications is the maximum delay required to reconstruct the compressed motion. In the proposed method this translates to the maximum number of frames in each segment (S_M). The delay can be controlled by reducing S_M generating a smaller size sequence. Table 5.4 shows that reducing S_M from 64 to 32 frames (t_L = 0.01) has insignificant effect on compression rate. For a 120 frames per second motion sequence the delay would be 120/32 = 0.27 second, showing the effectiveness of our method on regular size motion sequences.

CHAPTER SIX: HIGH EFFICIENCY PERCEPTUALLY MOTIVATED COMPRESSION OF MOTION CAPTURE DATA USING 1-D SPIHT

6.1 Introduction

Using human Motion Capture (MoCap) data is an efficient way to store and transmit body animations in several applications including interactive games, virtual environments and movie production. In online applications the bandwidth that needs to be dedicated to motion capture data can be considerable especially when multiple dynamic characters are present in the scene. MoCap data usually have high temporal (inter-channel) and spatial (intra-channel) correlation that can be exploited to reduce the transmission bandwidth.

Researchers in the field of image and video compression have extensively studied various methods to achieve higher compression based on sensitivity curves of human subjective response to different types of errors. However, research on perceptual factors of animation and its application in compression is still in its early stages. In this paper we study the effectiveness of exploiting several perceptual factors in the compression of animations.

In addition to the compression ratio being superior or at least comparable to current methods, the proposed method has a few properties making it suitable for online multi-character animation. These include:

1) **Real time, precise bandwidth allocation**: In other approaches the bandwidth allocation is equivalent to finding the input parameters (e.g., thresholds) of the compression algorithm in a way that the size of the compressed data is less than the allocated bandwidth. In most cases this is done by either using some estimates like rate-distortion model, or encoding the

data several times with different input parameters to find the optimal parameter setting. However, deriving an accurate rate-distortion model might not be straight-forward for some methods and encoding data several times makes the bandwidth allocation a computationally intensive task. This does not apply to our method, since using 1-D SPIHT makes it possible to simply transmit the data until the allocated bandwidth is used up.

2) **Shorter encoding and decoding time**: In comparison to other high efficiency methods ours needs a significantly lower processing time for both encoding and decoding. For example the encoding time is three orders of magnitudes shorter than [BPvdP07], because we achieve similar or better compression without the processor intensive optimization process. The decompression time is one order of magnitude better than [Ari06].

3) **Scalability:** The proposed method can provide a bit stream that contains several quality layers of the motion. In other words in a single pass of coding we can get a bit stream that can be adapted to the bandwidth and/or processing requirements of multiple devices by simply dropping a set of packets. While the scalability feature could be added to other wavelet-based methods, there would be a significant amount of redundancy, because unlike SPIHT some additional data is needed to transmit the location of non-zero coefficients.

6.2 Background and Related Work

Arikan's method [Ari06] (as discussed in chapter 2) does not seem to be suitable for real time encoding and decoding of relatively short motion clips. This is because they represent each joint with three 3-D marker positions (9 values) instead of just three rotation angles (3 values). This increases the dimensionality of the data three times which is compensated only when PCA is applied to larger clusters.

Wavelet-based compression has been shown to be effective in MoCap compression even without exploiting the spatial redundancy of the input data [BPvdP07, FCB11]. The concept is to apply 1-D wavelet transform to each channel of data, and then find and apply a certain threshold such that only a portion (p_c) of the wavelet coefficient remain non-zero. Compression is achieved by quantization, run-length and entropy coding the thresholded wavelet coefficients. In order for wavelet-based compression to be effective the portion (p_c) should be proportional to the effect of channel c on the quality of the reconstructed motion. Using a position distortion metric Baudin et al. [BPvdP07] propose a global optimization method to find the number of non-zero coefficients assigned to each channel to minimize the distortion metric, given the total number of non-zero coefficients. However, global optimization is computationally expensive, hence not suitable for real time multi-character applications. Instead, in chapter 3 and [FCB11] we assigned an importance score to each channel and allocated the given total non-zero coefficients in proportion to the importance of each channel. The importance score was the product of two perceptual factors namely the joint variation and the length of the bone attached to a joint. Based on the user studies on a variety of MoCap data the users did not notice a significant degradation in reconstructed motion when the data was compressed at the rate of 25:1. This study could be extended by considering the effect of other factors including the depth of the joint in the kinematic chain and different weights given to different kinematic chains.

MoCap compression can also be achieved by defining a database of motion primitives and replacing each segment of the motion with the closest match in the database. Gu et al. [GPD09] use separate set of primitives for different joints. The trajectory of each joint is divided into several segments. The segments are clustered using K-Means and the centers of the clusters are included in the primitive database. The main drawback of this approach is that the database needs to be transmitted along with the compressed data and therefore the encoding is not efficient for small motion sequences. For this reason we proposed (chapter 4 and [FCB13]) creating a general-purpose database of primitives from a set of training motions and storing it on the client side before encoding the motion. Also, the difference between the motion segment and the best matching primitive is transmitted if its energy is above a certain threshold, which is found by a series of user studies using the 2 Alternatives Forced Choice (2AFC) approach. While relatively high compression can be achieved (35:1), the best results are achieved for encoding motion with rhythmic and/or repetitive motion (up to 50:1).

Other related topics are motion level of details and perceptual quality of motion which were discussed in details in chapter 2.

6.3 Proposed Method

Our framework combines 1-D SPIHT with a bit rate allocation mechanism based on perceptual factors. The reasons we use 1-D SPIHT coding are:

1. It can efficiently encode a wide variety of motions from very short clips to very long sequences, as well as from monotonous to dynamic motion.

2. SPIHT supports fast and accurate bandwidth allocation.

Figure 6.1 gives a high level view of the proposed method. The parallelograms denote the input to/output of the processing blocks (rectangles). The details of each step are described in the following sections.

The scene/motion analysis block provides the information required to assign a relative importance score to different characters present in the scene as well as different parts of each character (at different levels, such as kinematic chain level, joint level and DoF level). The bandwidth allocation module works at two levels. At the scene level, the bandwidth available for motion data is distributed between the characters present in the scene based on the visual importance of characters (as will be discussed in section 6.5). At the character level the bandwidth allocated to each character is distributed between its channels of data based on the importance scores of the channels.

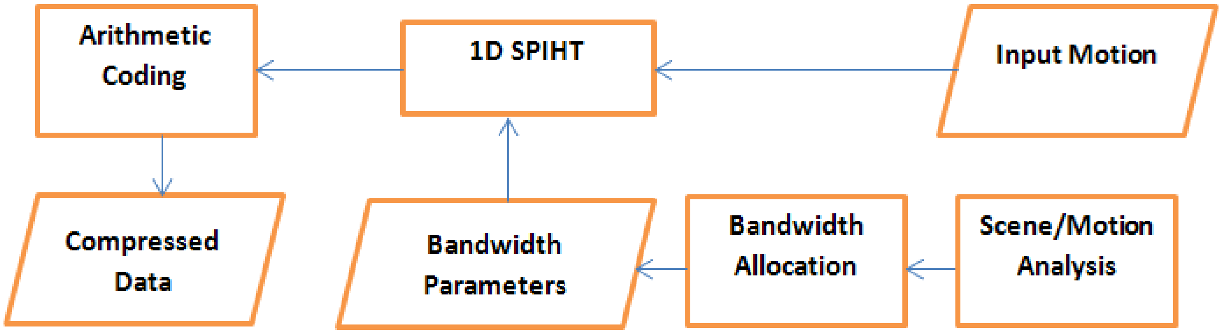


Figure 6.1: High level view of the proposed method.

The 1-D SPIHT provides semi-optimal bandwidth usage at bit level, meaning that the bandwidth allocated to each channel can be used without any waste to transmit a bit-stream with close to best possible reconstruction quality.

6.3.1 Data Representation

Motion capture data is produced by tracking and recording the trajectory of a series of markers positioned at important parts (typically joints) of an actor. In this research we use the hierarchical representation of motion (biped) (Figure 6.2). In this representation instead of the actual position of the marker, the rotation angles of the marker are stored. The rotation of each joint is represented by at most 3 Euler rotation angles which we call rotational degrees of freedom (DOFs). These rotational DOFs plus 3 translational (position) DOFs of the root joint along with the skeleton data are sufficient to reconstruct the actual position of all key points. For compression we define N channels of data (C_i $1 \leq i \leq N$) where each channel is an array storing the time samples of one DOF (data frames). Motion is typically sampled at a rate between 60 to 120 Hz (or frames per second). The data is divided into segments of W frames and each segment is compressed separately. Since wavelet-based compression is used we choose W to be a power of 2.

Encoding each segment separately has two important advantages:

1) It limits the transmission delay. In our experiments we used segment size of 128 frames which translates to a delay 1 to 2 seconds depending on the sampling rate.
2) It provides adaptability to bandwidth fluctuations because the bandwidth allocated to each segment can vary based on the available bandwidth at the time of transmission.

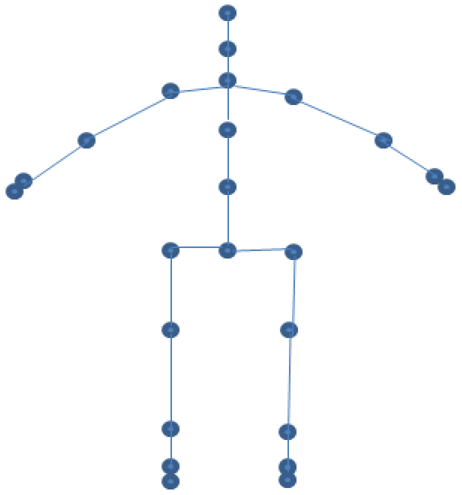


Figure 6.2: A Sample BiPed Hierarchy.

6.3.2 SPIHT Encoding

SPIHT is a progressive (embedded) encoding method that produces a bit stream with increasing accuracy. In other words as the decoder receives more bits it can reconstruct the data more accurately. Like its predecessor EZW it is designed to be applied on multi-level wavelet decomposition of the input data. The high compression achieved by SPIHT is due to the fact that the energy of the wavelet sub-bands decrease with scale. In other words the wavelet coefficients in the higher resolution sub-bands are usually smaller than those in lower resolution sub-bands.

While the original SPIHT works on images (or 2-D data in general), we propose a 1-D variation that works with individual channels of MoCap data. In 1-D Wavelet transform, the data is decomposed into a low-pass and a high pass sub-band, where the length of each sub-band is half the length of its parent. The next level is generated by applying wavelet to the low pass sub-band of the previous level. A sample three-level wavelet decomposition is shown in Figure 6.3.

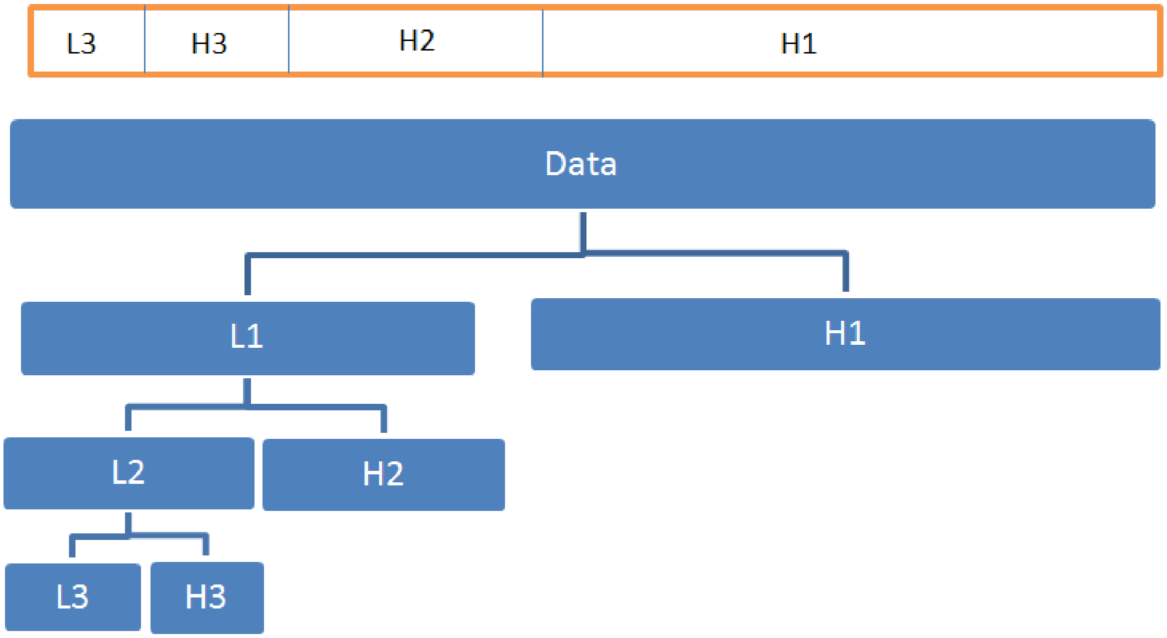


Figure 6.3: A sample three-level 1-D wavelet decomposition.

The original SPIHT algorithm [SP96] defines a series of tree structures and takes advantage of the high correlation between wavelet coefficients across sub-bands in each tree. In our 1-D implementation, with the exception of the highest and lowest levels, the *j*th node is parent to nodes 2*j* and 2*j*+1 (the index starts from zero). A sample 1-D and 2-D tree structure is depicted in Figure 6.4. Other details remain the same as the original SPIHT.

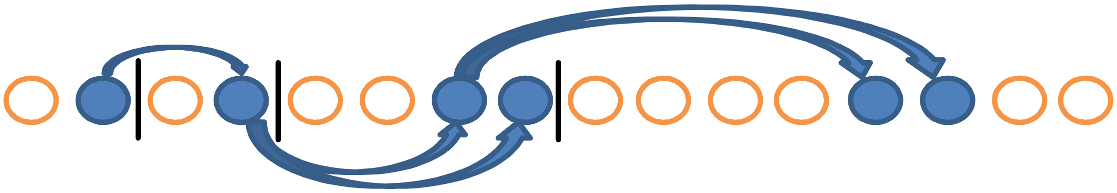


Figure 6.4: A zero-tree in a 3-level wavelet decomposition.

6.3.3 Character Level Bitrate Allocation

The concept behind perceptually-motivated bit allocation is to allocate more bits to channels with higher impact on the perceptual quality of the reconstructed motion. The best solution is to find the bitrate distribution that minimizes the error metric (solving an optimization problem). However, there are two main draw backs to this approach:

A) There is no existing error metric that correlates well with the perceptual quality of motion.

B) Since the dimension of data is relatively high (56 channels for a regular biped skeleton), the optimization process is very time consuming and therefore not suitable for real-time applications.

What can be done instead is to find the semi-optimal solution using a set of heuristic rules. In this research we assign an importance score (F_c) to each channel (c) and distribute the bandwidth between the channels in proportion with their importance score. The bandwidth (B_c) allocated to each channel can be defined as:

$$B_c = \frac{F_c B_T}{\sum_{i=1}^{C} F_i} \tag{6.1}$$

where $\boldsymbol{B_T}$ is the total bandwidth allocated to the current segment of the character's motion and C is the number of channels. We calculate the importance factor by combining (multiplying) several factors as described below. These factors are selected based on a) the extent to which error in the joint affects other joints (error propagation), b) the noticeability of the error around the joint and the attached bone area. Some of the factors like energy are specific to each channel while others are common (equal) between one or a series of joints.

A) Body part score: We divide the character into 6 kinematic chains namely right and left arm, right and left leg, torso and head**.** We assign a score between 0 and 1 (S_k) to each kinematic chain k, indicating its relative importance in the perceptual quality of motion. From previous research [Ari06] we know that error in feet movement are more noticeable than other parts of the body, especially because of the foot sliding effect that can happen when the reconstruction error is over a certain threshold. While the optimal score given to each kinematic chain may vary depending on the type of motion, in our experiments with several different types of motions we found that assigning a score of 0.5 to the arms, torso and head and 1 to the legs results in a higher or at least equal quality animation compared to giving equal weights to all body parts.

B) Position in the kinematic chain (Depth factor): Joints in the higher levels of the kinematic chain (like hip and shoulders) have a stronger effect on the motion than leaf joints (like fingers) and therefore should receive a higher bandwidth. We define the depth factor of the k-th joint (k) as:

$$P(j_k) = \frac{h - D(j_k) + 1}{h + 1} \tag{6.2}$$

where h is the maximum depth of the kinematic chain and $D(j_k)$ is the depth of the joint in the tree (each chain starts from depth 0).

C) Channel energy (variation): We define the energy of the *i*-th channel (C_i) as:

$$E(C_i) = \frac{1}{WD_i} \sum_{f=1}^{W-1} (C_i(f+1) - C_i(f))^2 \tag{6.3}$$

where W is the number of frames (window size) in the current segment, and D_i is $\max_{1 \le f \le W-1}(C_i(f+1) - C_i(f))^2$. This factor indicates the amount of variation in the channel. We incorporated this factor for two reasons: 1) For a fixed bit-rate the reconstruction error of different channels vary depending on their energy. In other words, channels with little energy can be reconstructed with higher precision using fewer bits. 2) It is an indicator of the amount of activity around the joint area. Areas with higher amount of activity tend to grab more attention [LTGH07][FCB11].

D) **Length of the bone attached to the joint**: Joint rotation leads to the rotation of the bone attached to it. So, the rotations of the joints with longer attached bones (like humerus) have a more noticeable effect on the pose of the skeleton than shorter bones (like fingers). We normalize the bone lengths by dividing them by the longest bone in their kinematic chain.

As an example consider the left shoulder joint. It has three DoFs and therefore three channels of data. Each channel gets the same scores for body part, depth factor and bone length but different scores for energy. The depth factor would be 1.0 (shoulder is the first joint in the arm kinematic chain, hence depth 0). The bone length factor would be 1.0 in most cases because the humerus is typically the largest bone in the arm kinematic chain.

6.4 Experiments

6.4.1 User Study Methodology

In order to validate our parameter selection and also compare the proposed method with other methods in the literature we use the Double Stimulus Continuous Quality Scale (DSCQS) method as discussed in ITU-R recommendation BT.500 [ITU]. DSCQS has been widely used for the quality assessment of video [CSRK11]. This method enables simultaneous assessment of the difference in quality between the reference and compressed animations, as well as the absolute quality of the compressed animation. In this method each test case consists of playing the reference (uncompressed) and compressed animations twice and then asking the user to give a quality score to each animation. The animations are displayed in random order (e.g., reference, compressed, reference, compressed, compressed, reference, etc.) and the user is not told which animation is the reference (Figure 6.5).

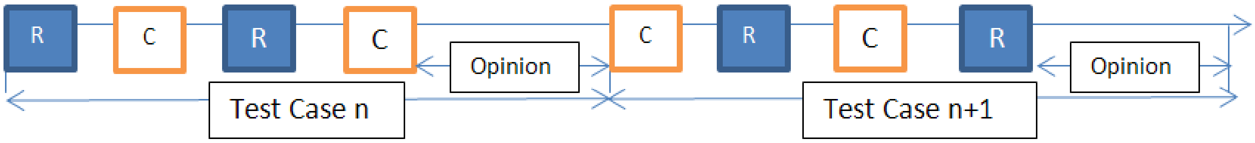


Figure 6.5: The DSCQS Method (R: Reference animation, C: Compressed animation).

In each test case the user is asked to give a score between 1 to 5 (1:Bad, 2:Poor, 3:Fair, 4: Good, 5: Excellent) to each of the two animations. The difference between the score of the pair is calculated and stored. The average value of this difference would be the DSCQS value of the compressed video.

6.4.2 User Study Details

In order to see the effect of the type of motion and also to increase the variety of animations presented to the user (to avoid fatigue) we used three different motion sequences, namely: (1) Walking, (2) Salsa dance, and (3) Window washing (Figure 6.6). These sequences were selected from the CMU MoCap database [CMU]. A total of 18 test cases were presented to the user. The compressed motion in each test case was the result of applying the proposed method with a specific bitrate. Also 6 of the test cases were used to validate our choice of weight given to each kinematic chain. Before starting the test the samples of high quality, medium quality (some visible artifacts) and low quality (lots of artifacts) animation were presented to the user. The MoCap files had 30 joints with a total of 56 Degrees of Freedom (DOFs) (53 rotational and 3 translational). More details about the user study are presented in Table 6.1.

Table 6.1: Details of the user study.

Total Test Time per subject: 20-25 Minute
Lighting: indoor incandescent
Screen: 19” wide screen monitor, resolution: 1440x900
Participant background: University students (undergraduate and graduate), 12 Males, 8 Females

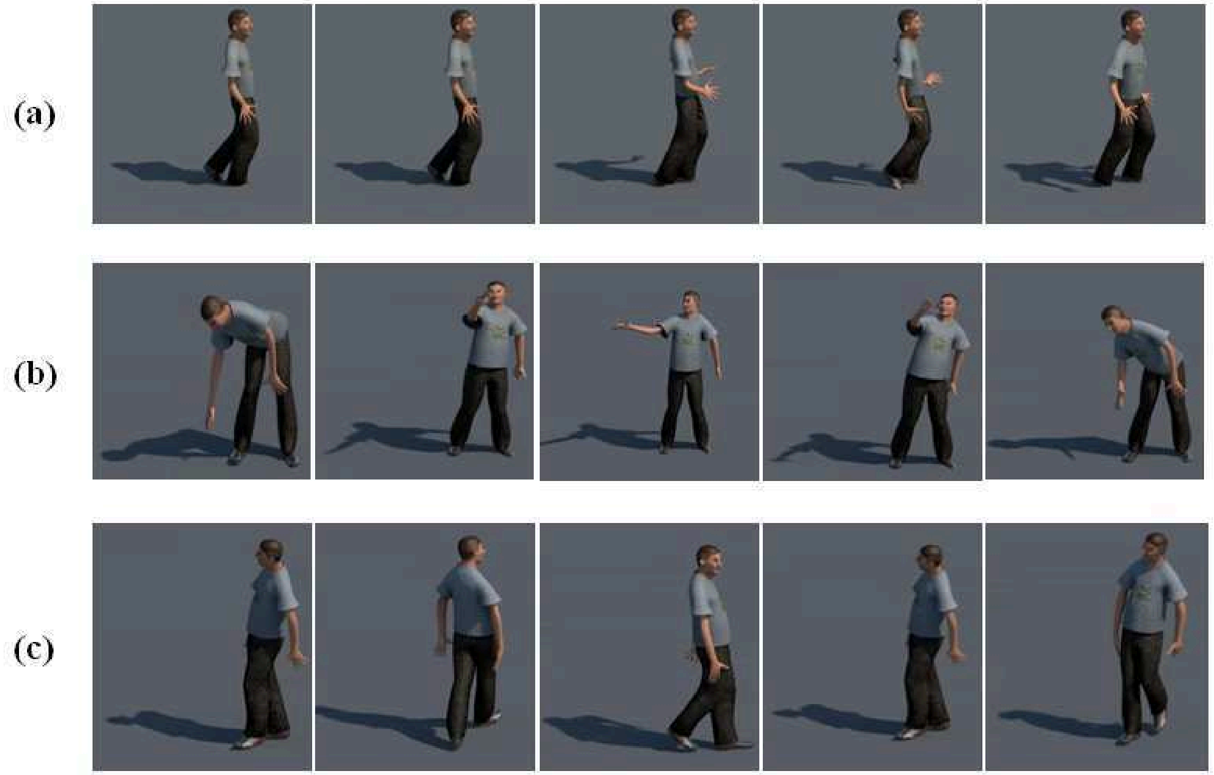


Figure 6.6: Animations used in the user study: (a) Salsa dance, (b) Window washing and (c) Walking.

6.4.3 Results

Figure 6.7 shows Differential Mean Opinion Scores (DMOSs) vs. compression ratio (CR) graph for the motions used in the user studies. When calculating CR the size of the input consists of arrays of 32 bit IEEE floats. The number of arrays is 56 (equal to the number of channels) and length of each array is the same as the length of the animation in frames.

For a compression ratio of 60:1, DMOS is less than 0.2 for the salsa dance and window washing sequences, and around zero for walking. This means that approximately *80%* of the users did not notice a significant degradation in the quality of the motion at this compression rate. In other words the proposed method is capable of compressing MoCap data at a rate of 60:1 with close to perfect perceptual reconstruction quality.

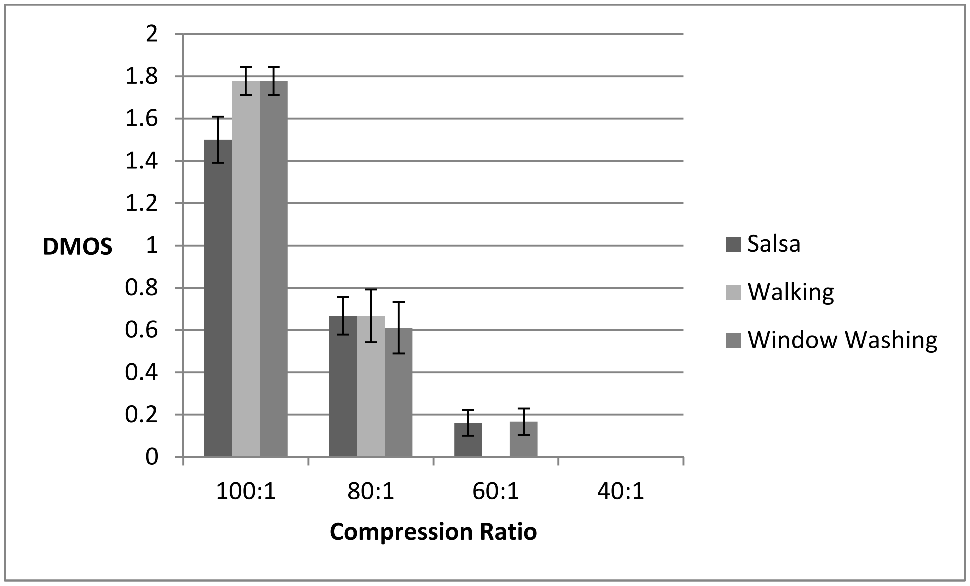


Figure 6.7: DMOS of the motions used in the user study vs. compression ratio.

We can see that for a compression ratio of 40:1 the DMOS is almost zero for all the motions.

The average and best compression reported by other methods in the literature is presented in Table 6.2. Observe that even with perfect reconstruction the proposed method outperforms other methods in the literature in terms of compression. With minimal degradation (DMOS<0.2) our method achieves a significantly higher compression. The method proposed in [Ari06] achieves the 30:1 ratio when compressing large databases of MoCap. While [BPvdP07] performs well on shorter motion clips the compression is very time consuming because of the optimization process. The method proposed in [GPD09] requires a preprocessing phase to create database of motion primitives making it unsuitable for real-time compression of short clips.

Table 6.2: Compression ratio (CR) of the current methods reported in the literature (CR=80:1)..

Method	[Ari06]	[BPV07]	[GPD09]	gZip
Average CR	30:1	25:1	25:1	3.3:1
Best CR	30:1	48:1	50:1	3.3:1

Table 6.3 compares the results of compressing the motion capture files with different weight combinations given to kinematic chains. It can be seen that giving a weight of 1 to legs and 0.5 to spine and arms will result in higher reconstruction quality for all the three MoCap files.

Table 6.3: DMOS for different combinations of weights given to kinematic chains.

Motion	Salsa Dance	Walking	Window washing
Arms= Spine= Legs	0.83	0.88	0.75
Arms=Spine=0.5, Legs=1	0.66	0.63	0.61
Arms=Spine=0.3, Legs=1	1.33	1.31	1.35

We measured the compression and decompression time on a desktop computer with an Intel Core 2Duo 2.66 GHz processor and 2GB of memory. The compression and decompression time of 20 MoCap files from the CMU database with different lengths and types of activity were measured (the CR was set to 60:1 to be comparable with other approaches). There was no significant difference between the compression and decompression times of different MoCap files. Table 6.4 compares the processing time of the proposed method with estimated times for other methods, showing that (1) the proposed method satisfies the requirements of online applications in terms of compression and decompression time; and, (2) the proposed method is superior to the optimized wavelet coefficient selection method [BPvdP07] in terms of compression time requirement.

Table 6.4: Average compression and decompression times of different methods in the literature.

Method	Comp. time Per frame (μs)	Decomp. Time Per frame (μs)
Proposed	520	100
[BPvdP07]	133000 to 483000	100
[Ari06]	1000	1200
[GPD09]	Depends on the size of input	100

6.5 Application in Crowd Motion Compression

When more than one character are present in the scene the bandwidth allocation module will distribute the total bandwidth allocated to motion data between the characters based on their “importance score”. The importance score is indicator of the extent to which errors in the motion of a character affects the perceptual quality of the scene. The importance score could be a combination of several factors, including the distance of the character from camera and the level and type of activity of the character, and occlusion. The effect of each factor in perceptual quality of the animation would require an extensive study. In this work we examine the effect of the distance of the character from the camera through a user study assuming other factors are similar for the characters present in the scene.

The study consisted of three similar characters performing a similar movement synchronously (figure 6.8). One character was placed closer to the camera and two other characters were placed at the same distant in the back. In order to examine the effect of distance from camera the back characters were placed at three different levels. The user study consisted of two runs, with each run consisting of 12 trials. In each trial the reference (uncompressed) clip was displayed first, followed by the compressed motion. The compression ratio of the front character was fixed at (40:1), while the compression ratio of the back character varied in each clip. In the first run subjects were asked to rate both the reference and the compressed clip between 1 and 5 (with 5 being the best quality) based on the general impression. In the second run the same experiment was repeated but the subjects were asked to pay attention to the characters at the back. 10 subjects (6 males, 4 females) with the similar background as the previous section participated in this study. Other test conditions were similar to the previous experiment.

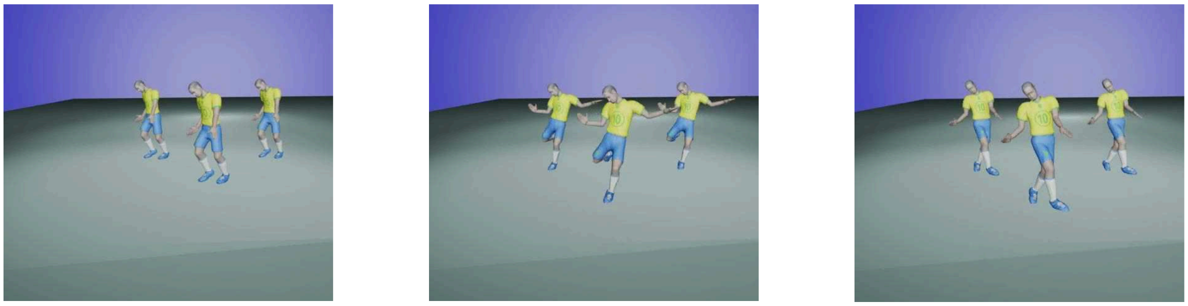
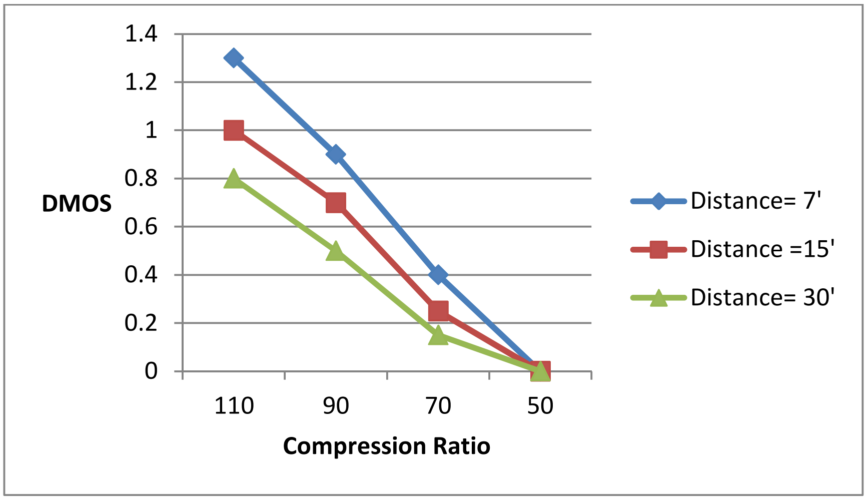
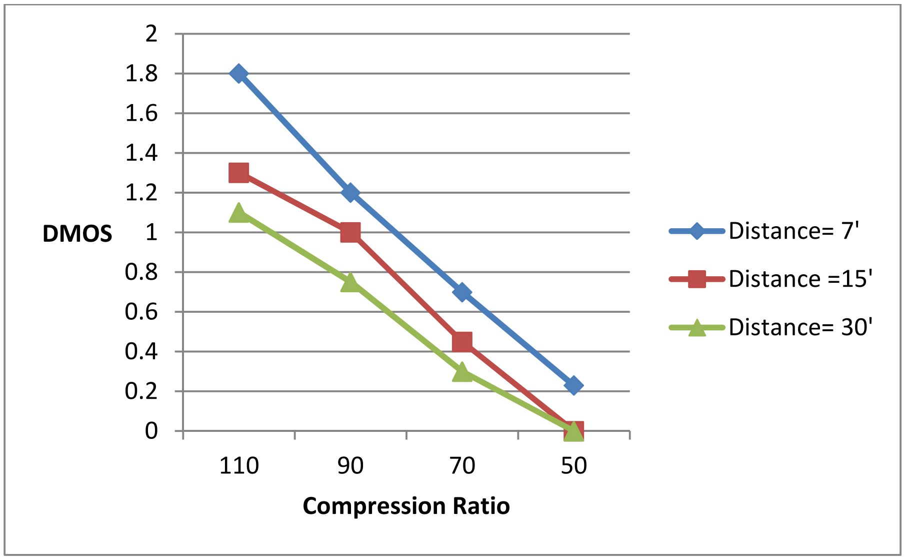


Figure 6.8: Crowd motion compression experiment. All the characters perform the same movement (kicking soccer ball).

The results of the user study are presented in figure 6.9. We can observe that the back characters can be compressed at a higher rate without noticeable degradation in the perceptual quality of the animation. Also, when users focused on the back characters the degradation became less noticeable when the back characters were farther from the camera.



(a)



(b)

Figure 6.9: Result of the user study when (a) Users are asked for a general impression, (b) Users are asked to pay attention to the characters at the back.

This initial experiment suggests that when compressing crowd motion using the proposed method, higher compression without perceptual quality loss can be achieved by allocating bandwidth to the characters in proportion with their distance to the camera. For example, if d_m is the distance of the front character to the camera, then for each character (C) we can define a bandwidth score as:

$$S_C = \frac{d_m}{d_C} \tag{6.4}$$

where d_C is the distance of the character from the camera. The bandwidth allocated to the k-th character B_k could be:

$$B_k = \frac{S_k B}{\sum_{i=1}^{M} S_i} \tag{6.5}$$

where B is the total bandwidth allocated to transmitting all motion data, and M is the number of characters present in the scene. This strategy could be implemented in the scene/motion analysis step of the proposed bitrate allocation framework (Figure 6.1).

CHAPTER SEVEN: CONCLUSION

7.1 Summary

We investigated the effect of incorporating human perceptual factors on compression and synthesis of MoCap data. We show how incorporating perceptual factors can have a significant impact on bandwidth and processing time requirements in different scenarios.

In chapter three 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 are more robust to constrained bandwidth, which is especially important in a mobile environment.

Our initial results in chapter four show that using a database of motion primitives we achieve high compression rates on rhythmic motion data, especially for long sequences. The proposed spatial segmentation step combined with temporal blending, reduces visual artifacts and produces satisfactory quality of experience in the reconstructed animation.

In chapter five we proposed a real-time compression technique making use of encoding incremental primitives and perceptual tolerance to exploit temporal motion redundancy. Results show that our method, in comparison with other state-of-the-art techniques, can achieve higher compression while preserving visual quality, with lower processing time. In addition, the rendering delay can be controlled by limiting the number of frames in each segment (S_M) without adverse impact on the compression rate. A distinct feature of our method is that the trade-off between quality and bandwidth can be easily controlled by varying a single parameter (t_L).

In chapter six we proposed a novel method that is useful for applications where one or more of the following is needed: 1) Low bandwidth (High compression) 2) Fast compression and decompression time 3) Scalability 4) Small transmission delay. We achieved this by combining the power of 1-D SPIHT with a heuristic function for bandwidth distribution. The heuristic function takes into account 4 perceptual factors of animation, namely: (1) Importance of the kinematic chain (body part); (2) Depth of the joint in the kinematic chain, (3) The length of the bone attached to the joint, and (4) The energy of the channel.

7.2 Future Work

In chapter three 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 [IAF06] could be further studied. We believe the work presented in that chapter 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.

We would also like to evaluate the method proposed in chapter five in interactive real-time training applications (like dance training systems [DLGY11]), and perform a larger scale user study to examine the effect of other perceptual parameters, including viewer’s region of interest.

We believe that the proposed method in chapter six can be extended to achieve even higher compression for crowd animation by exploiting: (1) The partial/full occlusion; (2) Similarity between the motions of different characters; and (3) Attention given to characters with a low importance score.

REFERENCES

[ACBD03] Agrafiotis, D., N. Canagarajah, D. R. Bull, and M. Dye. "Perceptually optimised sign language video coding based on eye tracking analysis." *Electronics letters*39, no. 24 (2003): 1703-1705.

[ADCB07] Agrafiotis, Dimitris, Sam JC Davies, N. Canagarajah, and David R. Bull. "Towards efficient context-specific video coding based on gaze-tracking analysis." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* 3, no. 4 (2007): 4.

[Ari06] Arikan, Okan. "Compression of motion capture databases." In *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 890-897. ACM, 2006.

[AV07] Arthur, David, and Sergei Vassilvitskii. "k-means++: The advantages of careful seeding." In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027-1035. Society for Industrial and Applied Mathematics, 2007.

[AW04] Ahn, Junghyun, and Kwangyun Wohn. "Motion level-of-detail: A simplification method on crowd scene." *Proceedings Computer Animation and Social Agents (CASA)* 129 (2004): 137.

[BSW93] Basu, Anup, Allan Sullivan, and Kevin Wiebe. "Variable resolution teleconferencing." In *Systems, Man and Cybernetics, 1993.'Systems Engineering in the Service of Humans', Conference Proceedings., International Conference on*, pp. 170-175. IEEE, 1993.

[Bin00] Bindiganavale, Ramamani N. "Building parameterized action representations from observation." PhD diss., University of Pennsylvania, 2000.

[BMN*08] Boccignone, Giuseppe, Angelo Marcelli, Paolo Napoletano, Gianluca Di Fiore, Giovanni Iacovoni, and Salvatore Morsa. "Bayesian integration of face and low-level cues for foveated video coding." *Circuits and Systems for Video Technology, IEEE Transactions on* 18, no. 12 (2008): 1727-1740.

[BPvdP07] Beaudoin, Philippe, Pierre Poulin, and Michiel van de Panne. "Adapting wavelet compression to human motion capture clips." In *Proceedings of Graphics Interface 2007*, pp. 313-318. ACM, 2007.

[BSO11] Balle, Johannes, Aleksandar Stojanovic, and J-R. Ohm. "Models for static and dynamic texture synthesis in image and video compression." *Selected Topics in Signal Processing, IEEE Journal of* 5, no. 7 (2011): 1353-1365.

[BSP*04] Barbič, Jernej, Alla Safonova, Jia-Yu Pan, Christos Faloutsos, Jessica K. Hodgins, and Nancy S. Pollard. "Segmenting motion capture data into distinct behaviors." In *Proceedings of Graphics Interface 2004*, pp. 185-194. Canadian Human-Computer Communications Society, 2004.

[BW95] Bruderlin, Armin, and Lance Williams. "Motion signal processing." In*Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 97-104. ACM, 1995.

[CB06] Cheng, Irene, and Walter Bischof. "A Perceptual Approach to Texture Scaling based on Human Computer Interaction." In *Proc. EUROGRAPHICS*. 2006., Short Paper, pp. 49-52.

[CB07] Cheng, Irene, and Anup Basu. "Perceptually optimized 3-D transmission over wireless networks." *Multimedia, IEEE Transactions on* 9, no. 2 (2007): 386-396.

[CCYJ08] Chi, Ming-Chieh, Mei-Juan Chen, Chia-Hung Yeh, and Jyong-An Jhu. "Region-of-interest video coding based on rate and distortion variations for H. 263+."*Signal Processing: Image Communication* 23, no. 2 (2008): 127-142.

[CDF92] Cohen, Albert, Ingrid Daubechies, and J-C. Feauveau. "Biorthogonal bases of compactly supported wavelets." *Communications on pure and applied mathematics* 45, no. 5 (1992): 485-560.

[CHN06] Chen, Zhenzhong, Junwei Han, and King Ngi Ngan. "Dynamic bit allocation for multiple video object coding." *Multimedia, IEEE Transactions on* 8, no. 6 (2006): 1117-1124.

[CMU]CMU GRAPHICS LAB: CMU graphics lab motion capture database [Online]. Avialable: http://mocap.cs.cmu.edu.

[CN99] Chai, Douglas, and King N. Ngan. "Face segmentation using skin-color map in videophone applications." *Circuits and Systems for Video Technology, IEEE Transactions on* 9, no. 4 (1999): 551-564.

[CS11] Wang, Zhou, Ligang Lu, and Alan C. Bovik. "Video quality assessment based on structural distortion measurement." *Signal processing: Image communication* 19, no. 2 (2004): 121-132.

[CSRK11] Chikkerur, Shyamprasad, Vijay Sundaram, Martin Reisslein, and Lina J. Karam. "Objective video quality assessment methods: A classification, review, and performance comparison." *Broadcasting, IEEE Transactions on* 57, no. 2 (2011): 165-182.

[CYB12] Cheng, Irene, Lihang Ying, and Anup Basu. "Perceptually Coded Transmission of Arbitrary 3D Objects over Burst Packet Loss Channels Enhanced with a Generic JND Formulation." *Selected Areas in Communications, IEEE Journal on* 30, no. 7 (2012): 1184-1192.

[CYC09] Chi, Ming-Chieh, Chia-Hung Yeh, and Mei-Juan Chen. "Robust region-of-interest determination based on user attention model through visual rhythm analysis."*Circuits and Systems for Video Technology, IEEE Transactions on* 19, no. 7 (2009): 1025-1038.

[CZYZ08] Chen, Qian, Guangtao Zhai, Xiaokang Yang, and Wenjun Zhang. "Application of scalable visual sensitivity profile in image and video coding." In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pp. 268-271. IEEE, 2008.

[DGL09] Deng, Zhigang, Qin Gu, and Qing Li. "Perceptually consistent example-based human motion retrieval." In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, pp. 191-198. ACM, 2009.

[DLGY11] Deng, Liqun, Howard Leung, Naijie Gu, and Yang Yang. "Real-time mocap dance recognition for an interactive dancing game." *Computer Animation and Virtual Worlds* 22, no. 2-3 (2011): 229-237.

[DMR01] Daly, Scott, Kristine Matthews, and Jordi Ribas-Corbera. "As plain as the noise on your face: Adaptive video compression using face detection and visual eccentricity models." *Journal of Electronic Imaging* 10, no. 1 (2001): 30-46.

[EH72] Eriksen, Charles W., and James E. Hoffman. "Temporal and spatial characteristics of selective encoding from visual displays." *Perception & psychophysics* 12, no. 2 (1972): 201-204.

[FAI*05] Arikan, Okan, and Leslie Ikemoto. *Computational Studies of Human Motion: Tracking and Motion Synthesis*. Now Publishers Inc, 2006.

[FCB11] Firouzmanesh, Amirhossein, Irene Cheng, and Anup Basu. "Perceptually guided fast compression of 3-D motion capture data." *Multimedia, IEEE Transactions on* 13, no. 4 (2011): 829-834.

[FCB13] Firouzmanesh, Amirhossein, Irene Cheng, and Anup Basu. "Perceptually Motivated Real-Time Compression of Motion Data Enhanced by Incremental Encoding and Parameter Tuning." In *Eurographics 2013-Short Papers*, pp. 61-64. The Eurographics Association, 2013.

[FCB13b] Firouzmanesh, Amirhossein, Mitch Lindgren, Teri Drummond, Irene Cheng, and Anup Basu. "Efficient compression of rhythmic motion using spatial segmentation and temporal blending." In *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*, pp. 1-4. IEEE, 2013.

[FCB14] Firouzmanesh, Amirhossein, Irene Cheng, and Anup Basu. "High Efficiency Perceptually Motivated Compression of Motion Capture Data Using 1-D SPIHT. " Submitted to *Multimedia, IEEE Transactions on*.

[Fin09] Findlay, John M. "Saccadic eye movement programming: Sensory and attentional factors." *Psychological research* 73, no. 2 (2009): 127-135.

[FP03] Fang, Anthony C., and Nancy S. Pollard. "Efficient synthesis of physically valid human motion." *ACM Transactions on Graphics (TOG)* 22, no. 3 (2003): 417-426.

[FXG12] Fan, Rukun, Songhua Xu, and Weidong Geng. "Example-Based Automatic Music-Driven Conventional Dance Motion Synthesis." *Visualization and Computer Graphics, IEEE Transactions on* 18, no. 3 (2012): 501-515.

[Gal09] Gall, Jurgen. “Filtering and optimization strategies for markerless human motion capture with skeleton-based shape models.” PhD Thesis, Universität des Saarlandes, Saarbrücken, Germany (2009).

[Gar98] García-Pérez, Miguel A. "Forced-choice staircases with fixed step sizes: asymptotic and small-sample properties." *Vision research* 38, no. 12 (1998): 1861-1881.

[Gle97] Gleicher, Michael. "Motion editing with spacetime constraints." In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pp. 139-ff. ACM, 1997.

[GPD09] Gu, Qin, Jingliang Peng, and Zhigang Deng. "Compression of human motion capture data using motion pattern indexing." In *computer graphics forum*, vol. 28, no. 1, pp. 1-12. Blackwell Publishing Ltd, 2009.

[GW08] Gonzalez, Rafael, and Richard Woods, *Digital Image Processing*, Pearson Education Inc., Third Edition (2008):604-614.

[HRvdP04] Harrison, Jason, Ronald A. Rensink, and Michiel Van De Panne. "Obscuring length changes during animated motion." *ACM Transactions on Graphics (TOG)* 23, no. 3 (2004): 569-573.

[IAF06] Ikemoto, Leslie Kanani Michiko, Okan Arikan, and David Forsyth. "Quick motion transitions with cached multi-way blends." *University of California, Berkeley Technical Report No. UCB/EECS-2006-14* (2006).

[IF04] Ikemoto, Leslie, and David A. Forsyth. "Enriching a motion collection by transplanting limbs." In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 99-108. Eurographics Association, 2004.

[IAF06b] Ikemoto, Leslie, Okan Arikan, and David Forsyth. "Knowing when to put your foot down." In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pp. 49-53. ACM, 2006.

[II86] Iwasaki, Masayuki, and H. Inomata. "Relation between superficial capillaries and foveal structures in the human retina." *Investigative ophthalmology & visual science* 27, no. 12 (1986): 1698-1705.

[Itti07] Itti, Laurent. "Visual salience." *Scholarpedia* 2, no. 9 (2007): 3327.

[IKN98] Itti, Laurent, Christof Koch, and Ernst Niebur. "A model of saliency-based visual attention for rapid scene analysis." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 20, no. 11 (1998): 1254-1259.

[ISO04] ISO/IEC JTC1/SC29/WG11. “Animation Framework eXtension (AFX).” Published by ISO (2004).

[ITU] BT, ITU-R. Recommendation. "500-11. Methodology for the subjective assessment of the quality of television pictures." *International Telecommunication Union, Geneva, Switzerland* (2002): 53-56.

[Itt04] Itti, Laurent. "Automatic foveation for video compression using a neurobiological model of visual attention." *Image Processing, IEEE Transactions on* 13, no. 10 (2004): 1304-1318.

[Joh73] Johansson, Gunnar. "Visual perception of biological motion and a model for its analysis." *Perception & psychophysics* 14, no. 2 (1973): 201-211.

[Jon80] Jonides, John. "Towards a model of the mind's eye's movement." *Canadian Journal of Psychology/Revue canadienne de psychologie* 34, no. 2 (1980): 103.

[KB96] Kaiser, Peter K., and Robert M. Boynton. *Human color vision.* Vol. 287. Washington DC: Optical Society of America, 1996.

[KG03] Kovar, Lucas, and Michael Gleicher. "Flexible automatic motion blending with registration curves." In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 214-224. Eurographics Association, 2003.

[KPS03] Kim, Tae-hoon, Sang Il Park, and Sung Yong Shin. "Rhythmic-motion synthesis based on motion-beat analysis." In *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 392-401. ACM, 2003.

[Kuf53] Kuffler, Stephen W. "Discharge patterns and functional organization of mammalian retina." *J Neurophysiol* 16, no. 1 (1953): 37-68.

[LCR*02] Lee, Jehee, Jinxiang Chai, Paul SA Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. "Interactive control of avatars animated with human motion data." In *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3, pp. 491-500. ACM, 2002.

[LD04] Li, Ze-Nian, and Mark S. Drew, *Fundamental of Multimedia*, Pearson Education Inc., First Edition (2004) chapter 7.

[LE12] Lee, Jong-Seok, and Touradj Ebrahimi. "Perceptual Video Compression: A Survey." *Ieee Journal Of Selected Topics In Signal Processing (ISSN: 1932-4553)* 6, no. 6: 684-697.

[LFA*10] Li, Yi, Cornelia Fermuller, Yiannis Aloimonos, and Hui Ji. "Learning shift-invariant sparse representation of actions." In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2630-2637. IEEE, 2010.

[LJ11] Lin, Weisi, and C-C. Jay Kuo. "Perceptual visual quality metrics: A survey."*Journal of Visual Liu, Guodong, and Leonard McMillan. "Segment-based human motion compression." In Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 127-135. Eurographics Association, 2006.*

[LP02] Liu, C. Karen, and Zoran Popović. "Synthesis of complex dynamic character motion from simple animations." In *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3, pp. 408-416. ACM, 2002.

[LPB01] Lee, Sanghoon, Marios S. Pattichis, and Alan C. Bovik. "Foveated video compression with optimal rate control." *Image Processing, IEEE Transactions on* 10, no. 7 (2001): 977-992.

[LQI11] Li, Zhicheng, Shiyin Qin, and Laurent Itti. "Visual attention guided bit allocation in video compression." *Image and Vision Computing* 29, no. 1 (2011): 1-14.

[LTG*07] Lowry, Paul, Nathan Twyman, James Gaskin, Bryan Hammer, Aaron Bailey, and Tom Roberts. "Proposing the Interactivity-Stimulus-Attention Model (ISAM) to explain and predict enjoyment, immersion, and adoption of purely hedonic systems." In *Special Interest Group on Human-Computer Interaction Pre-ICIS Workshop (best-paper nomination)*, pp. 72-76. 2007.

[MH10] Martini, Maria G., and Chaminda TER Hewage. "Flexible macroblock ordering for context-aware ultrasound video transmission over mobile WiMAX."*International journal of telemedicine and applications* 2010 (2010): 6.

[MNO07] McDonnell, Rachel, Fiona Newell, and Carol O'Sullivan. "Smooth movers: perceptually guided human motion simulation." In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 259-269. Eurographics Association, 2007.

[NH10] Nyström, Marcus, and Kenneth Holmqvist. "Effect of compressed offline foveated video on viewing behavior and subjective quality." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* 6, no. 1 (2010): 4.

[OGO*08] Onder, Onur, U. Gudukbay, B. Ozguc, Tanju Erdem, Cigdem Erdem, and Mehmet Ozkan. "Keyframe reduction techniques for motion capture data." In*3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2008*, pp. 293-296. IEEE, 2008.

[PCB06] Pan, Yixin, Irene Cheng, and Anup Basu. "Quality metric for approximating subjective evaluation of 3-D objects." *Multimedia, IEEE Transactions on* 7, no. 2 (2005): 269-279.

[PB00] Pollard, Nancy S., and Fareed Behmaram-Mosavat. "Force-based motion editing for locomotion tasks." In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1, pp. 663-669. IEEE, 2000.

[PB02] Pullen, Katherine, and Christoph Bregler. "Motion capture assisted animation: Texturing and synthesis." In *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3, pp. 501-508. ACM, 2002.

[PJAP07] Preda, Marius, Blagica Jovanova, Ivica Arsov, and Françoise Prêteux. "Optimized MPEG-4 animation encoder for motion capture data." In *Proceedings of the twelfth international conference on 3D web technology*, pp. 181-190. ACM, 2007.

[PHM03] Pollick, Frank E., Joshua G. Hale, and Phil McAleer. "Visual perception of humanoid movement." (2003): 107-114.

[PMM04] Pierrot-Deseilligny, Charles, Dan Milea, and René M. Müri. "Eye movement control by the cerebral cortex." *Current opinion in neurology* 17, no. 1 (2004): 17-25.

[PP10] Pejsa, Tomislav, and Igor S. Pandzic. "State of the Art in Example-Based Motion Synthesis for Virtual Characters in Interactive Applications." In *Computer Graphics Forum*, vol. 29, no. 1, pp. 202-226. Blackwell Publishing Ltd, 2010.

[PW99] Popović, Zoran, and Andrew Witkin. "Physically based motion transformation." In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 11-20. ACM Press/Addison-Wesley Publishing Co., 1999.

[RCB98] Rose, Charles, Michael F. Cohen, and Bobby Bodenheimer. "Verbs and adverbs: Multidimensional motion interpolation." *Computer Graphics and Applications, IEEE* 18, no. 5 (1998): 32-40.

[RP03] Reitsma, Paul SA, and Nancy S. Pollard. "Perceptual metrics for character animation: sensitivity to errors in ballistic motion." In *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 537-542. ACM, 2003.

[RPE*05] Ren, Liu, Alton Patrick, Alexei A. Efros, Jessica K. Hodgins, and James M. Rehg. "A data-driven approach to quantifying natural human motion." In *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 1090-1097. ACM, 2005.

[Ren06] Ren, Liu. "Statistical analysis of natural human motion for animation." PhD diss., *Georgia Institute of Technology,* 2006.

[SALZ06] Sun, Yu, Ishfaq Ahmad, Dongdong Li, and Ya-Qin Zhang. "Region-based rate control and bit allocation for wireless video transmission." *Multimedia, IEEE Transactions on* 8, no. 1 (2006): 1-10.

[SB94] Sekuler, R., and R. Blake. *Perception,* McGraw-Hill Inc (1994).

[SCE01] Skodras, Athanassios, Charilaos Christopoulos, and Touradj Ebrahimi. "The JPEG 2000 still image compression standard." *Signal Processing Magazine, IEEE* 18.5 (2001): 36-58.

[SEB03] Sheikh, Hamid R., Brian L. Evans, and Alan C. Bovik. "Real-time foveation techniques for low bit rate video coding." *Real-Time Imaging* 9, no. 1 (2003): 27-40.

[SHP04] Safonova, Alla, Jessica K. Hodgins, and Nancy S. Pollard. "Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces."*ACM Transactions on Graphics (TOG)* 23, no. 3 (2004): 514-521.

[SI08] Shiratori, Takaaki, and Katsushi Ikeuchi. "Synthesis of dance performance based on analyses of human motion and music." *Information and Media Technologies* 3, no. 4 (2008): 834-847.

[SK08] Souto, David, and Dirk Kerzel. "Dynamics of attention during the initiation of smooth pursuit eye movements." *Journal of Vision* 8, no. 14 (2008).

[SKG03] Shin, Hyun Joon, Lucas Kovar, and Michael Gleicher. "Physical Touch-Up of Human Motions." In *Pacific Conference on Computer Graphics and Applications*, pp. 194-203. 2003.

[SKL07] Sok, Kwang Won, Manmyung Kim, and Jehee Lee. "Simulating biped behaviors from human motion data." *ACM Transactions on Graphics (TOG)* 26, no. 3 (2007): 107.

[Slo07] Slot, Kristine. *Motion blending*. Copenhagen University, Department of Computer Science, February 2007.

[SM08] Savoye, Yann, and Alexandre Meyer. "Multi-layer level of detail for character animation." *Virtual Reality Interaction and Physical Simulation (VRIPHYS'08)*(2008).

[SP96] Said, Amir, and William A. Pearlman. "A new, fast, and efficient image codec based on set partitioning in hierarchical trees." *Circuits and Systems for Video Technology, IEEE Transactions on* 6, no. 3 (1996): 243-250.

[SY09] Sauer, Danielle, and Yee-Hong Yang. "Music-driven character animation." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* 5, no. 4 (2009): 27.

[TCA*09] Tournier, Maxime, Xiaomao Wu, Nicolas Courty, Elise Arnaud, and Lionel Reveret. "Motion compression using principal geodesics analysis." In *Computer Graphics Forum*, vol. 28, no. 2, pp. 355-364. Blackwell Publishing Ltd, 2009.

[TCYT06] Tang, Chih-Wei, Ching-Ho Chen, Ya-Hui Yu, and Chun-Jen Tsai. "Visual sensitivity guided bit allocation for video coding." *Multimedia, IEEE Transactions on* 8, no. 1 (2006): 11-18.

[TXC*09] Tournier, Maxime, Xiaomao Wu, Nicolas Courty, Elise Arnaud, and Lionel Reveret. "Motion compression using principal geodesics analysis." In *Computer Graphics Forum*, vol. 28, no. 2, pp. 355-364. Blackwell Publishing Ltd, 2009.

[Tro02] Troje, Nikolaus F. "Decomposing biological motion: A framework for analysis and synthesis of human gait patterns." *Journal of vision* 2, no. 5 (2002).

[TRP07] Tsapatsoulis, Nicolas, Konstantinos Rapantzikos, and Constantinos Pattichis. "An embedded saliency map estimator scheme: Application to video encoding."*International Journal of Neural Systems* 17, no. 04 (2007): 289-304.

[TWC*09] Tournier, Maxime, Xiaomao Wu, Nicolas Courty, Elise Arnaud, and Lionel Reveret. "Motion compression using principal geodesics analysis." In *Computer Graphics Forum*, vol. 28, no. 2, pp. 355-364. Blackwell Publishing Ltd, 2009.

[vWvBE*10] Van Welbergen, Herwin, Ben JH Van Basten, Arjan Egges, Zs M. Ruttkay, and Mark H. Overmars. "Real Time Animation of Virtual Humans: A Trade-off Between Naturalness and Control." In *Computer Graphics Forum*, vol. 29, no. 8, pp. 2530-2554. Blackwell Publishing Ltd, 2010.

[WB03] Wang, Jing, and Bobby Bodenheimer. "An evaluation of a cost metric for selecting transitions between motion segments." In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 232-238. Eurographics Association, 2003.

[WB04] Wang, Jing, and Bobby Bodenheimer. "Computing the duration of motion transitions: an empirical approach." In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 335-344. Eurographics Association, 2004.

[WB08] Wang, Jing, and Bobby Bodenheimer. "Synthesis and evaluation of linear motion transitions." *ACM Transactions on Graphics (TOG)* 27, no. 1 (2008): 1.

[Wiki1] Wikipedia, “Motion capture,” available:http://en.wikipedia.org/wiki/Motion_capture.

[WIKI1] http://en.wikipedia.org/wiki/File:Schematic_diagram_of_the_human_eye_en.svg

[WIKI2] http://en.wikipedia.org/wiki/File:Wikipedia-spotlight.jpg

[WIKI3] http://en.wikipedia.org/wiki/Contrast_(vision)

[WIKI4]

http://commons.wikimedia.org/wiki/File:Contrast_Sensitivity_vs._Spacial_Frequency.png

[WL65] Wetherill, G. B., and H. Levitt. "Sequential estimation of points on a psychometric function." *British Journal of Mathematical and Statistical Psychology* 18, no. 1 (1965): 1-10.

[WLB03] Wang, Zhou, Ligang Lu, and Alan C. Bovik. "Foveation scalable video coding with automatic fixation selection." *Image Processing, IEEE Transactions on* 12, no. 2 (2003): 243-254.

[WB01] Wiebe, Kevin James, and Anup Basu. "Improving image and video transmission quality over ATM with foveal prioritization and priority dithering." *Pattern Recognition Letters* 22, no. 8 (2001): 905-915.

[WP95] Witkin, Andrew, and Zoran Popovic. "Motion warping." In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 105-108. ACM, 1995.

[WZLG09] Wang, Minghui, Tianruo Zhang, Chen Liu, and Satoshi Goto. "Region-of-interest based dynamical parameter allocation for H. 264/AVC encoder." In *Picture Coding Symposium, 2009. PCS 2009*, pp. 1-4. IEEE, 2009.

[ZCYZ08] Zhai, G., Q. Chen, X. Yang, and W. Zhang. "Scalable visual significance profile estimation." *submitted to International Conference on Acoustics, Speech, and Signal Processing*. 2008.

[Zha01] Zhao, Liwei, and Norman I. Badler. "Synthesis and acquisition of laban movement analysis qualitative parameters for communicative gestures." (2001).