國立交通大學

多媒體工程研究所

碩 士 論 文

將空間-時間連續性之輻射狀基底函數應用於角色動作壓縮之研究

Human Motion Compression with Space-time Radial Basis Functions

研 究 生：林昭自

指導教授：林奕成　助理教授

中 華 民 國 九 十 六 年 六 月

將空間-時間連續性之輻射狀基底函數應用於角色動作壓縮之研究

研究生：林昭自　　　指導教授：林奕成　博士

國立交通大學

多媒體工程研究所

摘要

近十年由於電腦圖學相關技術的高度發展，越來越多的虛擬的角色被運用各式各樣不同的應用當中。為了使這些虛擬角色能表現各式各樣具意義的動作，專業的動作捕捉系統便因應而生。動作捕捉系統擷取由真人所表演的動作，而相關的應用程式便可以利用這些動作資料來驅使形形色色的虛擬角色。然而，為了表現動作的多樣性，我們往往需要準備大量的角色動作資料來豐富整個動作資料庫，所以如何有效壓縮動作資料變成為近年來極具有挑戰性的研究主題。

根據觀察，我們可以輕易發現一般的動作往往具有大量冗餘資料。舉例來說，動作由某一時間點轉移到下一個時間點的變化可能十分有限。又譬如某一段時間內，左手臂及右手臂揮舞的方式可能是對稱的。這種情況我們稱此動作具有某種連貫性 (coherence)。因此只要我們能找出動作內的連貫性，我們就有機會抿除冗餘的資料，也就是說，原始的資料以一個較緊實的方式保存下來。

為了能有效運用在空間域及時間域的連貫性，我們將動作資料切割成許多段不同的子動作，對於每一串子動作，我們提出一個簡單有效的方式將具有高度關連性的關節點軌跡排列在一起形成一平滑曲面，最後我們採用一個多維的幅狀基底函數同時在時間域與空間域近似原始的資料，利用遠少於原始資料數量的取樣點所構成的函數來表示動作資料。

我們相信此種演算法具高度實用價值，同時由於此演算法被設計成一個有彈性的壓縮元件，任何未來的研究人員可以將此元件與其他壓縮元件做耦合以求更急遽成長的壓縮倍率。

Human Motion Compression with Space-time Radial Basis Functions

Student: Chao-Chih Lin **Advisor: Dr. I-Chen Lin**

Institute of Multimedia Engineering

National Chiao Tung University

Abstract

In this thesis, we present a space-time compression method for body animation. To extract significant features and reduce redundancy, we utilize both spatial and temporal coherence in motion data. The basic concept of this thesis is fitting the motion trajectories by specific functions.

In order to utilize coherence effectively, we segment the original motion into several sub-clip by incremental principal component analysis. For each sub-clip, we propose an easy and effective method to group several joints with similar trajectories. Finally, we use a modified radial basis functions to approximate these surfaces in temporal and spatial domain simultaneously.

We believe such an approach is a feasible compression technique for common body animation.

Keyword : motion compression, radial basis function, PCA, space-time coherence.

誌　　　謝

兩年的碩士生涯，轉眼間到了尾聲。要感謝的人非常多，但是直接謝天卻又顯得過於兒戲，因此我也不免俗的要感謝：

1. 我的父母。感謝你們從小到大的栽培，讓我最後一段求學的過程中能無後顧之憂，致力於研究，也感謝你們支持我的選擇。
2. 指導教授 林奕成博士。感謝老師這兩年來的指導，在我遇到許多問題時能給予我非常多中肯的意見，同時也不讓我偏離光明的方向。
3. 任右學長。感謝你的 BVH Loader，這個 Loader 簡單易用，讓我能將心力專注在研究上。
4. 藍英綬學弟。寫程式常會遇到形形色色難以解決的問題，感謝你熱心討論以及解答。
5. CAIG 實驗室全體成員。大家有緣而相聚，兩年的時間內，大家一起趕作業，一起打球，留下了很多美好的回憶。

Contents

List of Figures

1. Introduction

1.1 Motivation

Recently, more and more 3D characters or avatars have come to our daily life. To animate these virtual characters, experienced artists usually have to adjust their key postures. On the other hands, motion capture technique (abbreviates as mocap) is a feasible method to produce body animations. The most popular mocap, optic mocap, is based on the theories of computer vision. First, several feature markers are placed on a subject's body. The subject usually wears a black leotard to enhance the distinctness of feature markers in cameras. The 3D trajectory of feature markers can be estimated by triangulation. Mocap data contain the structure of the skeleton, (usually represented in a hierarchy structure.) and the degree of freedom of each joint in the entire animation sequence. To represent variety of human motions, capturing and storing considerable motion data are usually unavoidable. However, due to the hardware limitation (capacity and capability of computation), only a small set of mocap data can be loaded or processed at the same time. An immediate method to overcome this problem is representing the original motion data in a compressed form. Even though, video and audio compression have been developed for decades. However, the characteristic of human motion data is quite different from video or audio ones. How to compress human motion data effectively is still an interesting and challenging research topic. Therefore, we want to find out a suitable and practical approach such that we can store the motion data in a compact form and maintain the quality of animations as well as possible.

1.2 Introduction

In this thesis, we focus on the compression of body motions captured by mocap systems. A well-designed compression algorithm should have a good tradeoff between data size and data quality. Principal component analysis (PCA) is widely used in existent motion compression methods. By projecting original data set into a lower dimensional space, we have smaller data size but loss some low-variation features. This technique utilizes spatial and temporal coherence implicitly and is easily implemented. However, the entire motions have to be loaded in memory in the compressed phase. Besides, the features discarded by PCA are uncontrollable. Sensible jars may appear in some conspicuous motions. Users need to apply additional smoothing approaches or reduce compression ratios.

A traditional technique for image compression uses perceptual models. Pioneers explore the importance of data such that less important or less sensitive features can be omitted. Although various models have been proposed in video or audio compression, there is still much work need to be researched in the region of body animation.

One notable characteristic of motion data is considerable coherence, more specifically both spatial and temporal coherence. Due to the articulation of human skeletons, there exist spatial relationships between the neighbor joints. For instance, the gradual contraction and stretch of muscles make us capable of predict how skeletons move in the next few frames. Therefore we can approximate the body animation in spatial domain, or temporal domain or even both of them.

The basic concept of our compression algorithm is an extension to key-frame animation. In the key-frame animation, users select several important key frames and interpolate the in-between animation. In this thesis, we generalize this concept. Instead of choosing key frames from animations, we select key features. Each key feature represents the position of a marker captured by mocap devices in a particular time click. If we assume the structure (i.e. connectivity) of the markers is consistent, we can access any position of marker by its spatial and temporal index. Furthermore, we not only interpolate the data in temporal domain but also in spatial domain. Once we choose several key features from the original data, we establish an appreciate interpolation method. Given the temporal and spatial index, the other non-key features could be approximated by interpolation. There are many adoptable interpolation techniques that we can choose. Cubic Bezier curve or B-spline is one candidate. But such function approximates animations in temporal domain only. While apply Bezier surfaces, the data have to be sampled regularly. In this thesis, we use radial basis functions to approximate the motion sequence. Each key feature with time position can be thought as a sample or center in the hyper space. (space and time) We select several key features and put them into the radial basis functions network to establish our approximation functions. In the decompressed phase, the other non-key features can be reconstructed according their spatial and temporal index efficiently.

Since our goal is to compress motion data and maintain the visual quality. We believe that well-selected centers have large effects on the decompressed results. Therefore, how to choose the centers of radial basis functions and how many centers are sufficient are also issues. In this thesis, we use a greedy-algorithm to decide these two questions. This is an iterative procedure. In each iterative step, a small set of centers are selected to train the radial basis functions. Our system evaluates the difference between the original data and the fitted function. Features with large residual will be chosen as additional centers. These steps will be repeated in the next loop until stop criterions are satisfied.

Although many researches show that radial basis functions has an upper-bound of the centers' amount. Fortunately, there are not too many key features of human motion data in the common case. People usually place 20~50 markers on the subject and each clip is rarely longer than a thousand frames. To utilize more coherence, we further propose reorganizing original data. Since different motion behaviors may have different space-time coherence, we segment motion data into several clips according to their movements. In each segment, we group joint samples according to their similarity. We believe that the more coherence we exploit, the more compression ratio we will achieve.

1.3 **Angular or Euclidean system?**

Most motion data are represented in terms of quaternions. The commonly used "BVH" file format is composed of a hierarchical structure in the angular domain. (see Appendix: "BVH file format") Such hierarchical structure is very sensitive to small reconstruction error. This is because error closer to the root will propagate and accumulate to the farer one. In other words, representing posture in Euclidean domain can tolerate more reconstruction error. Therefore, we reinterpret motion data as the global position of each joint frame by frame and perform approximation in Euclidean coordinate system. After this process, we can access the position of each joint at specific time stamp by its joint index and frame index.

1.5 Normalized Motion

The initial positions and orientations are different between motion data. We call such un-aligned motions as "raw motion data". If motion data are aligned with their local coordinate system, we call them as "normalized motion". From our observation, motion clips often have more spatial coherence in normalized motion space. This situation is appreciable when current motion is symmetric especially. For example, if people raise their arms upward simultaneously, joints of left and right arms have more coherence in normalized motion space. Therefore, we convert each motion data into the normalized space for more efficient usages of spatial coherence.

1.6 Flowchart

Figure 1 is our system flowchart. Our system can be divided into 3 major parts: Segmentation, Clustering, and Approximation. After we load a BVH file and represent it as a normalized motion, we perform temporal segmentation on this motion according to their behaviors. Then we group joints with similar trajectories to form a smooth surface. Finally we use space-time radial basis functions to approximate each surface or curve and calculate the difference.

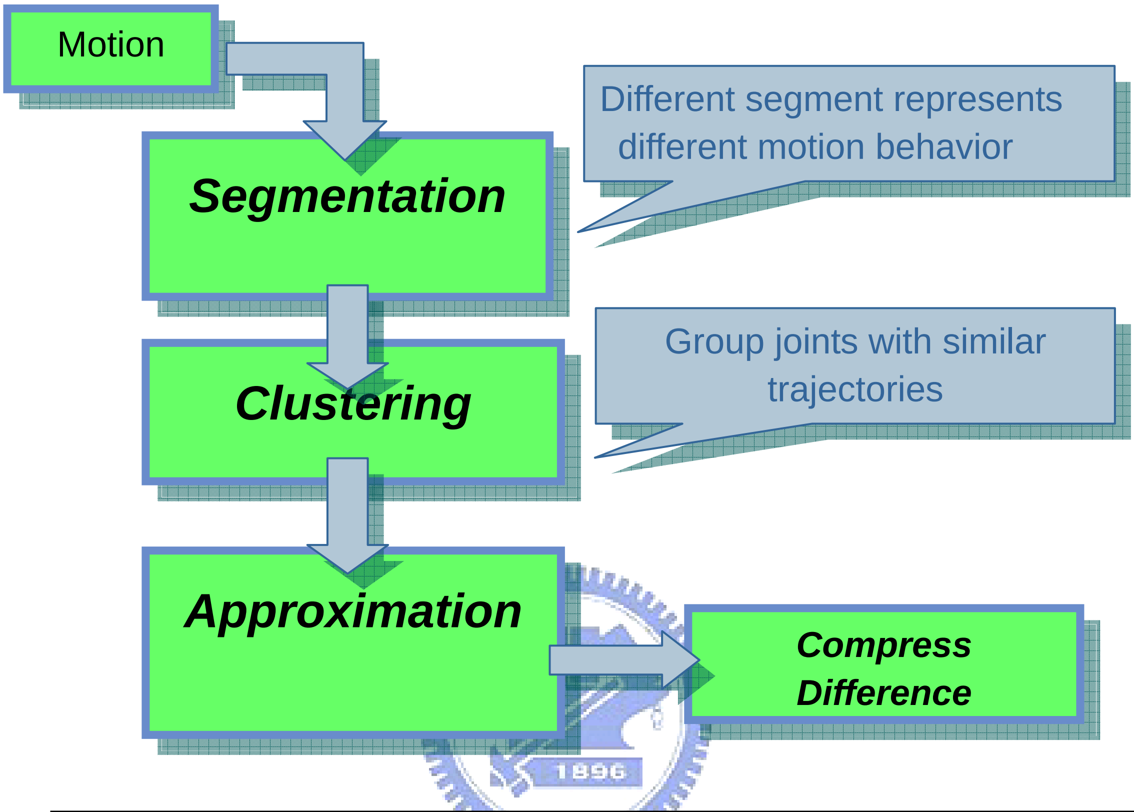


Figure 1: Our system flowchart. There major parts are: “Segmentation”, “Clustering”, and “Approximation”.

2. Related Work

There have been extensive researches on the compression of time-varying data, especially, video and audio. Several animation compression techniques use theories from these researches. Goskov et al [4] developed their compression algorithm through a wavelet framework. They used a multi-resolution approach to encode animation sequences progressively. They also applied interframe difference of wavelet details to improve the compression ratios.

Goskov [4] and other earlier researchers mainly focused on compression of animated meshes. They usually compressed each frame individually. For instance, Rossignac [11] and Karni [8] both proposed algorithms to encode the triangle indices. However, there are both spatial and temporal coherence in common animations. Thus, Ibarria and Rossignac [5] used a space-time predictor and corrector to compress animated meshes with fixed-connectivity.

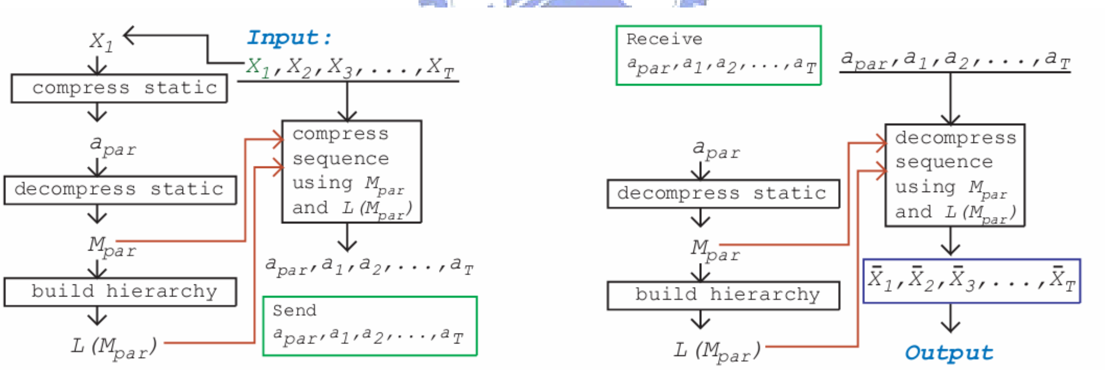


Figure 2Goskov's algorithm [4]

In recent years, more and more motion data have been used in various kinds of applications. For instance, movie and game industries need lots motion data to drive 3D characters. Therefore, skeletal motion compressions become an important research topic.

Principal component analysis, usually abbreviated as PCA is a considerable approach for animation compression. It represents high-dimensional data in a lower dimensionality without loss much information. For instance, Alexa and Muller [1] projected the entire animation sequences into a lower dimension space through PCA such that the motion data can be represented in a more compact form. PCA exploits spatial coherences implicitly. For effectively utilizing these coherences, motion data set was usually clustered into groups before PCA is applied. Such an approach was called clustered- PCA (CPCA). Therefore, perceptually or geometrically similar motion clips can share the same PC but have different coefficients. Arikan [9] provided a hierarchical technique to compress body animation database. The original database was divided into several clips. In order to utilize temporal coherence, author used cubic Bezier curves to approximate the trajectories of joints. Obviously, each clip can not be too long. (usually 16-32 frames.) Then the author clustered these parameterized clips into several groups and performed principal component analysis to reduce the dimensionality of such clips. To preserve meaningful high-frequency features (i.e. feet touching ground), He used a motion-JPEG technique to encode such important contacts. The major limitation of his approach is that contacts need to be known.

In the same year, Liu et al.[3] proposed a segment-based approach to compress human motion data. They segment original motion data into several clips by probabilistic PCA (PPCA). Then they perform PCA analysis on these segments to reduce the dimensionality and use Bezier curves to fit the coefficients of PC finally. They both utilize temporal and spatial coherence, but in two phases.

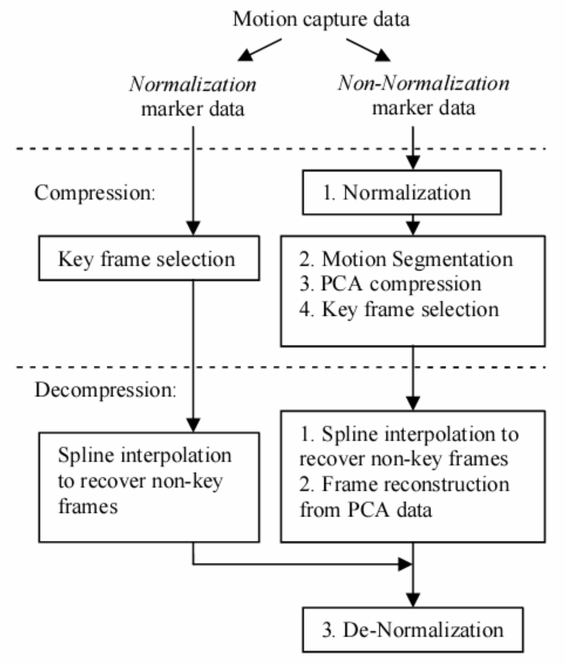


Figure 3: Liu [3] segment motion data and compress each segment by PCA

Besides compressing the captured data directly, an idea is using fewer markers to drive skeletons. Using representative markers lowers the data size and reduces ambiguities during post-processing of mocap data. In other words, the goal is to decide which marker can be removed or which marker must be kept in the data. To find out the redundancy, Liu et al [2] adopted a data-driven modeling approach to learn piecewise local linear models and use a modified principal feature analysis to choose the subset of markers. Original motion can be approximated by using a reduced marker set and these local linear models.

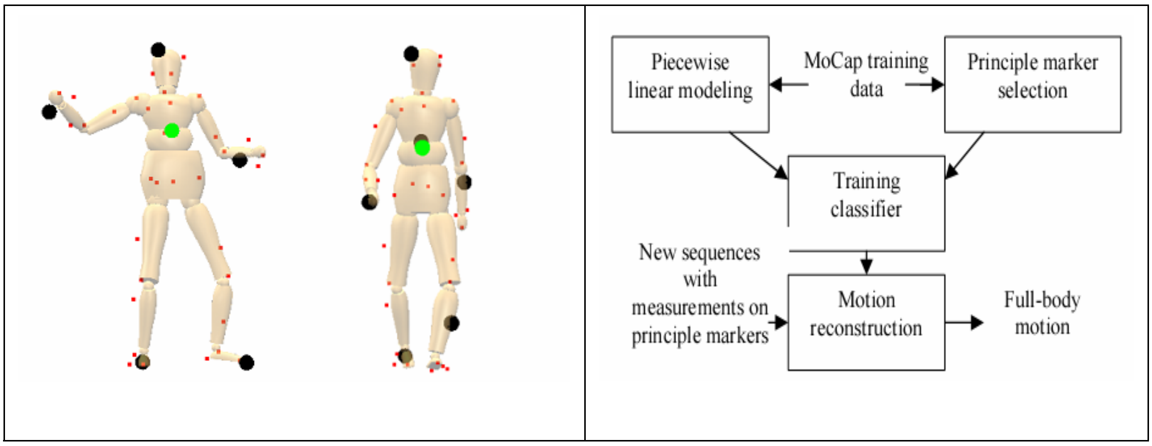


Figure 4: Liu [2] select fewer markers to drive original skeleton.

Approximation/Interpolation of animation is also an interesting approach. Uses choose several key frames and other frames can be estimated by interpolation or even extrapolation. There are many considerable mathematical interpolation methods, for instance, polynomial functions, trigonometric functions, exponential functions or splines. Mukai and Kuriyama [14] thought that motion interpolation can be approximated by weighted combination. They treat motion interpolations as statistical predictions. Arikan and Forsyth [10] synthesized human motions by a cutting-pasting concept. This approach generated smooth motions and satisfied spatial constraints. To approximate animations, the original animations are usually represented as parametric forms such that the motion trajectories can be fitted with curves. Igarashi et al [13] proposed an interpolation technique for performance-driven character animations. Given several predefined key frames, they used radial basis functions in spatial domain to interpolate the in-between frames.

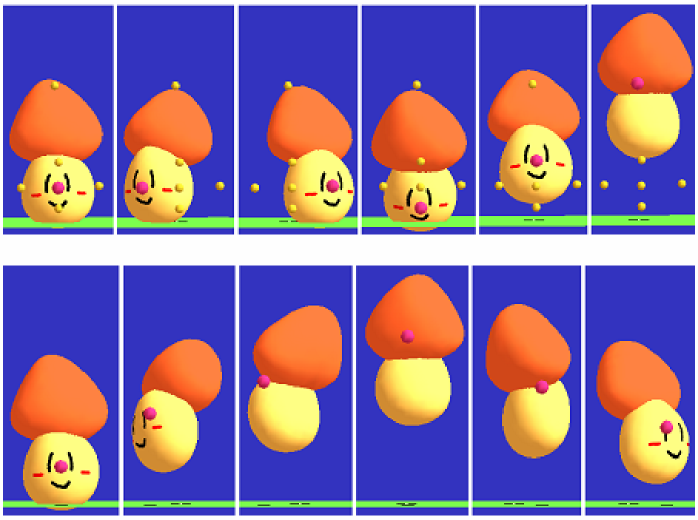


Figure 5: Igarashi et al [13] choose several key postures from the original motion sequence and use radial basis functions to interpolate in-between data.

Generally speaking, radial basis function is a very powerful interpolation tool. Through some proper modifications on distance metric, the radial basis functions can be expended into the hyper space. In other words, information between different dimensions may be shared.

Ravi et al [11] realized this idea. They used a modified radial basis functions to interpolate the BRDF samples between different positions and viewing angles. Such approach makes information can be shared across angular and spatial domain. Since animations have both spatial and temporal coherences, this means the information may spreads in these domains. Therefore we can adopt this concept to exploit information sharing.

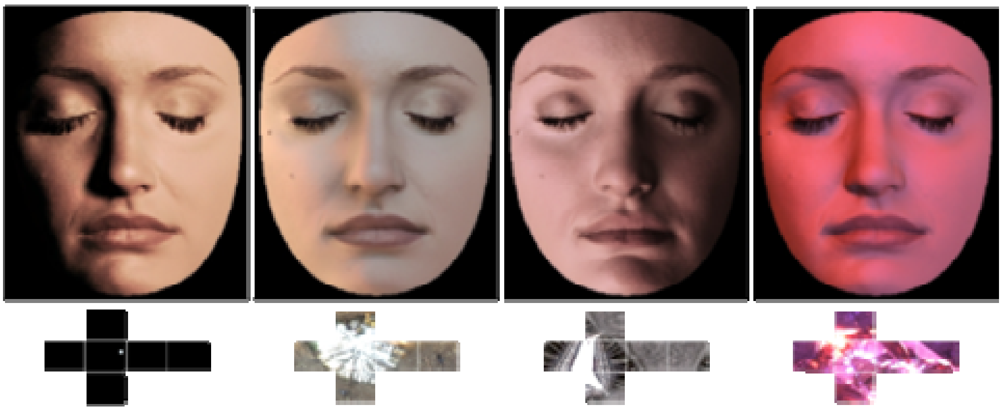


Figure 6: Ravi [11] used a modified radial basis functions to utilize data coherence across different domains.

3. Compression through Surface Approximation

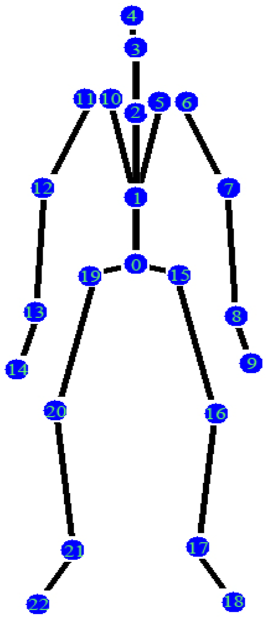


Figure 7: Illustration of joint index.

For the convenience of implementation, we assign an unique joint index to each joint after converting them from angular offset to global positions. Figure 7 is an illustration of the joint indices. For example, joint index 0 is the hip, and joint index 9 is the left hand. With these indices, we can store their positions in array and access them by indices.

As we pointed out in the introduction, the trajectory of each joint is essentially a curve. Therefore, if we lay several joint trajectories together, we have a surface. Figure 8 is an example. In this figure, joints highlighted in yellow have similar trajectories. We collected these trajectories and form a smooth surface. The red arrow indicates the temporal direction/domain and the green arrow indicates different joint indices (i.e. spatial domain).

Since a smooth surface means the grouped data are more consistent and can be encoded with fewer parameters, therefore, our goal become rearranging data. The following chapters “Segmentation” and “Clustering” will introduce how we choose these samples.

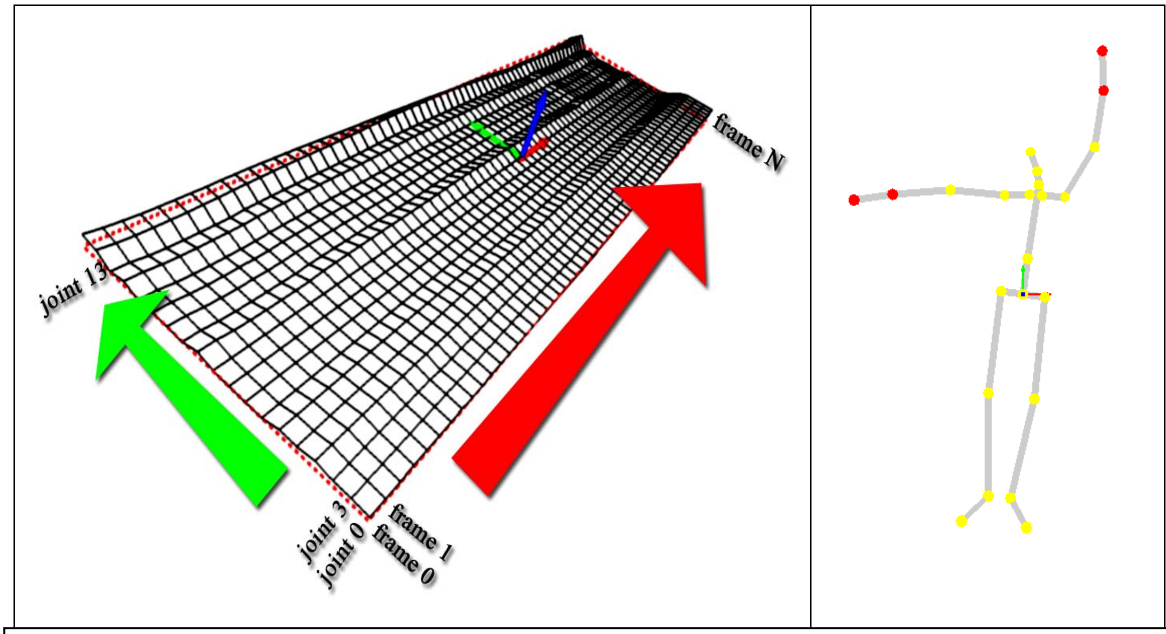


Figure 8: An example of constructed surface. The left one illustrates that several joints with similar trajectories form a smooth surface. The right one represents the corresponding joints. (colored in yellow.)

4. Segmentation

Generally speaking, any motion data can be thought as a concatenation of different logical behaviors. For example, a motion may have two logical states: from walking to running. If we carefully segment motion data into several distinct motion behaviors, we may have more spatial coherence to utilize. Besides, behavior transition usually results in intense variations. Therefore, if we can cut appropriately, we can alleviate such high variations and preserve the smoothness in local surface.

In this thesis, we adopt Barbic, J et al [6]'s method to segment our motion data. Although they proposed 3 methods to segment motion automatically, we choose incremental PCA-based method. The idea is based on the observation where simple motion can be represented better in lower dimensions than the complex one in the same dimensions.

When projecting motion data into a lower dimensional space, unavoidable error will be introduced. The definition of error term is:

$$e = \sum_{j=r+1}^{N} \sigma_j^2 \qquad (1)$$

where r is the number of principal components, N means original dimension. σ is the singular value of SVD.

The ratio

$$E_r = \frac{\sum_{j=1}^{r} \sigma_j^2}{\sum_{j=1}^{N} \sigma_j^2} \qquad (2)$$

is an indicator to tell us how much information retained after projection. Once we decide how much error we can accept, the number of principal components is decided. Given a starting short clip, we calculate its principal components. Then we append a new frame and perform PCA on this segment again with same number of principal components again. For simple motion, the error ratio will rise steady. If the motion is transiting into a new behavior, the error ratio will rise much quickly and a cut should be placed.

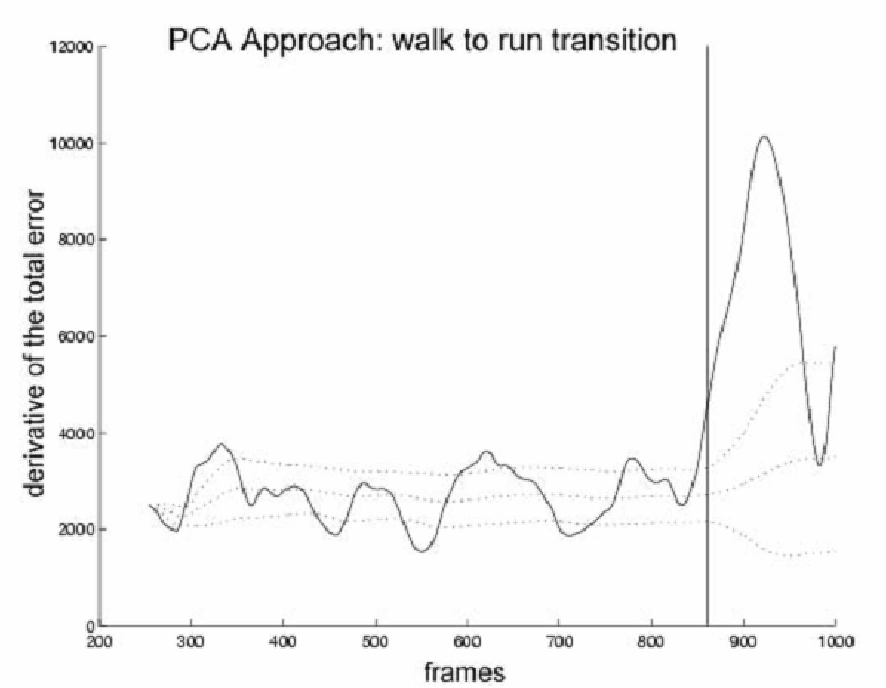


Figure 9: J. Barbic [6] used IPCA to segment motion. Once the error with same number of PC rise quickly, we assign a cut here.

Each motion segment represents a different motion behavior. Although we avoid high variations in the temporal domain by segmentation, there still exist high variations between joints in spatial domain with in the segment.

A simple method to alleviate this situation is to compute the mean posture of each segment as a reference. And this reference will be subtracted from each joint sample in this segment. Our experiments show that the equalization process is worth. The reason is that these joints form a very smooth surface after position shifting. When we use radial basis functions to approximate these surfaces, far fewer important samples are sufficient. In other words, this process can help us finding more coherence in spatial domain. Of course, we have to pay efforts to store these references.

Let us consider an example in figure 10. The left figure and right figure are both same clusters, but the left one does not equalize to the reference posture. The right one is smoother than the left one obviously.

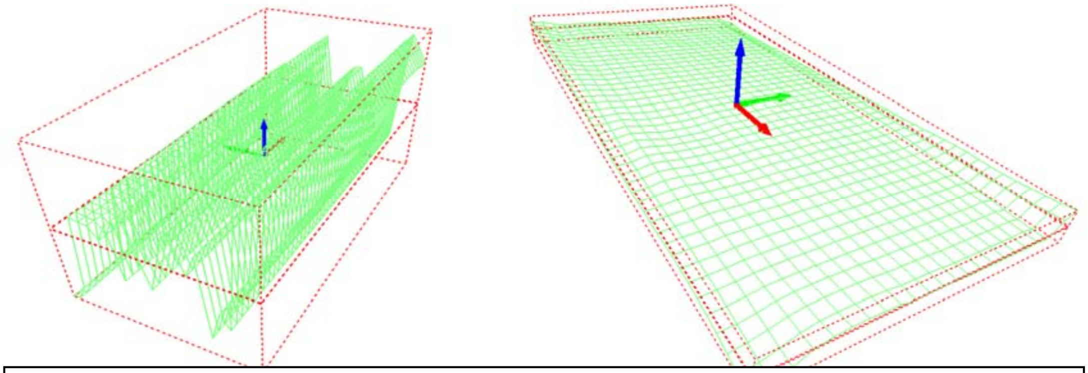


Figure 10: The comparison of shifting joint trajectories to reference posture.

5. Clustering

To exploit more coherence, we conduct many experiments for evaluation. From observation, we realize joints have more spatial coherence or relation with near one. For example, there exists high relationship between the hip and the chest but not hands or legs. Therefore we may separate entire body into several parts: head, extremities, and torso. If we can approximate a trajectory of joint by curve fitting, we can group many joints to form a surface and approximate it. Afterward, we perform surface approximation for each body part independently. However, there are some problems here. When people running, there exists dramatically variation in the parts of arms or legs. If we collect joints within this fixed part to form a single surface, such a surface is usually jarring and very difficult to find out useful spatial coherence. Besides, each body part is independent. Even some joints may have spatial coherence, we can not utilize them when they are assigned in different body-parts. Therefore, a practical approach is to analyze the behaviors of joints in the current segment and group joints with similar trajectories into a single cluster. A straightforward method is using the correlation of joints. But our experiments show that this is an unsuitable approach because some joint's positions may be fixed in a specific time interval. We can not find the linear relationship with other joints.

In order to solve this problem, we calculate the difference between each joint pair frame by frame. And the similarity was defined by the standard deviation of these differences. When any two joints have similar trajectories, the difference values do not spread in wide range. In other words, the standard deviation will be a lower value. After defining the similarity of any joint pair, we can group similar joints according to these values. In our thesis, we propose using an algorithm based on the k-means clustering but replace the distance metric with the similarity.

The number of cluster is another issue we need to consider. For example, if all of the joints do not move quickly in current segment, we should group all of them in a single cluster. By contrast, if joints move quickly, usable coherences will be relatively lower and the amount of cluster elements should be much fewer.

In order to adjust the number of clusters dynamically, we have to check the average standard deviation of the current cluster when the k-means algorithm converges. If this value exceeds a predefined threshold, the current cluster will be subdivided and re-clustering until the average standard deviation of each cluster is smaller than threshold.

Here is an example of our clustering result. In this segment, people rises his arms almost symmetrically from frame 497 to frame 557. According to our thesis, joints with similar trajectories will be grouped together. Figure 12 shows that we have 4 clusters in this segment. They are {joint 7, joint 12}, {joint 9, joint 14}, {joint 13, joint 8}, and residual joints.

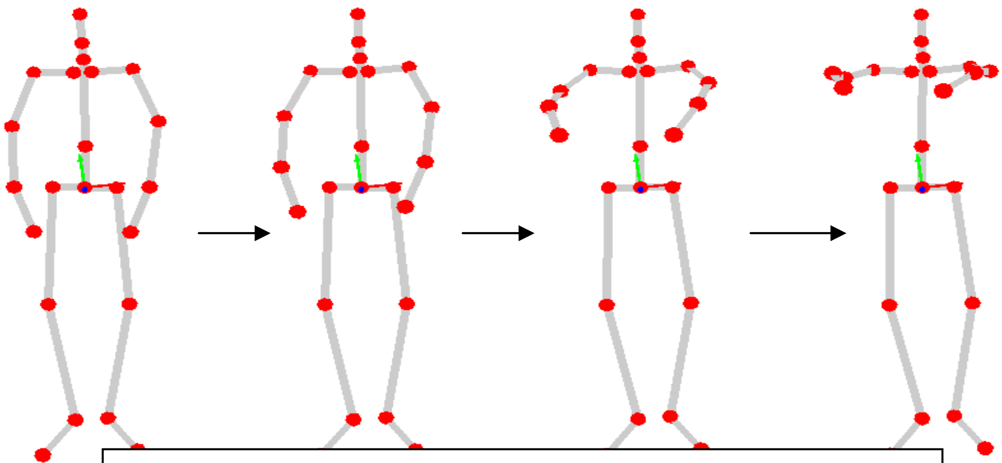
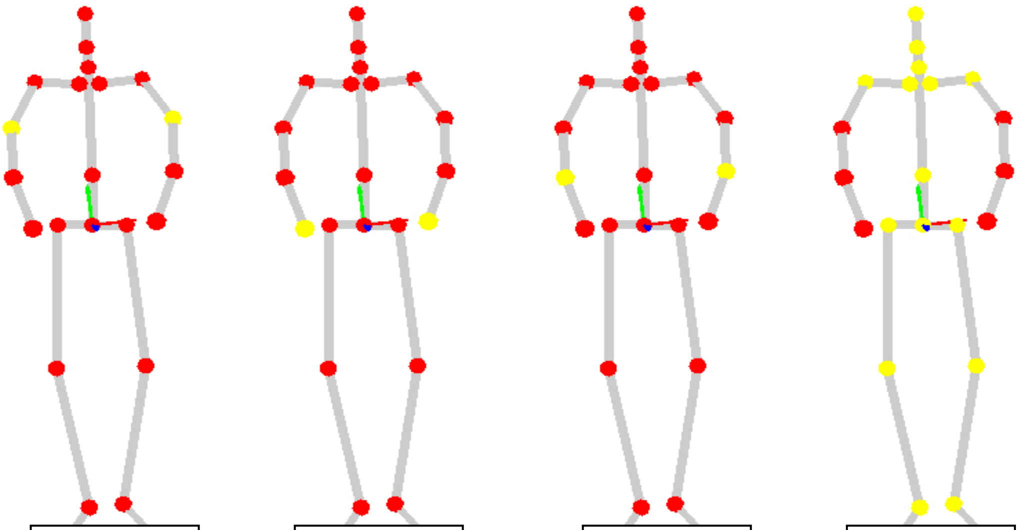


Figure 11: A motion segment with rising arms symmetrically.



Cluster 0	Cluster 1	Cluster 2	Cluster 3

Figure 12: These are results after clustering. The highlight joints (yellow color) indicate that they are grouped together.

6. Approximation

6.1 Radial Basis Function

Since a general trajectory of human motion is continuous and smooth. Arikan [10] used cubic Bezier curves to approximate the trajectory of each joint. But such functions only approximate animation in temporal domain. Each joint is independent in spatial domain. To utilize both spatial and temporal coherence, we modify the radial basis functions to approximate body movement in two domains concurrently.

Radial basis functions are commonly used in function approximation, scatter-data interpolation, and time series prediction. The basic concept is that any smooth function can be approximated by weighted combination of basis functions. Since radial basis functions are constructed from scatted data sets, we call these scattered data as samples or centers because the basis is positioned there. We adjust the amplitude and width of our basis to form a smooth function. A general equation can be written as:

$$y(x) = \sum_{i=1}^{N} w_i \phi(\|x - x_i\|) + p(x) \tag{4}$$

where *y(x)* is the function value at position ***x***, *N* is number of centers, coefficients *w* means the amplitude of each basis ϕ. ||.|| denotes the distance between centers. *P(**x**)* is a low polynomial term or called the affine term also. The commonly chosen of this basis function may be:

$$\phi(r) = r^3 \tag{5}$$

$$\phi(r) = e^{-kr^2} \tag{6}$$

$$\phi(r) = \sqrt{r^2 + k^2} \tag{7}$$

In our thesis, we choose Gaussian basis e^{-kr^2}. The constant *k* affects the width of each basis. Once *k* was chosen, we picked several samples as centers to train radial basis functions (i.e. solve coefficients *w*, See Appendix: “How to solve RBFs coefficients”) and we have a smooth curve or surface which is close to the real data.

6.2 Space-time Radial Basis Functions

If we have any two joints $Joint_i$ and $Joint_j$ whose coordinates are (x_i, y_i, z_i, t_i) and (x_j, y_j, z_j, t_j) in hyper-space. Typically, the distance term should be defined as:

$$r^2 = (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 + (t_i - t_j)^2 \tag{8}$$

However, we reorganized the joint sequence due to the processing mentioned previously. In our thesis, we redefine the basis with distance as:

$$\phi = e^{-\left(\frac{r_s^2}{c_s} + \frac{r_t^2}{c_t}\right)} \tag{9}$$

$$r_s^2 = (joint\_index_i - joint\_index_j)^2 \tag{10}$$

$$r_t^2 = (frame\_index_i - frame\_index_j)^2 \tag{11}$$

where c_t and c_s are constants which control the shape of basis.

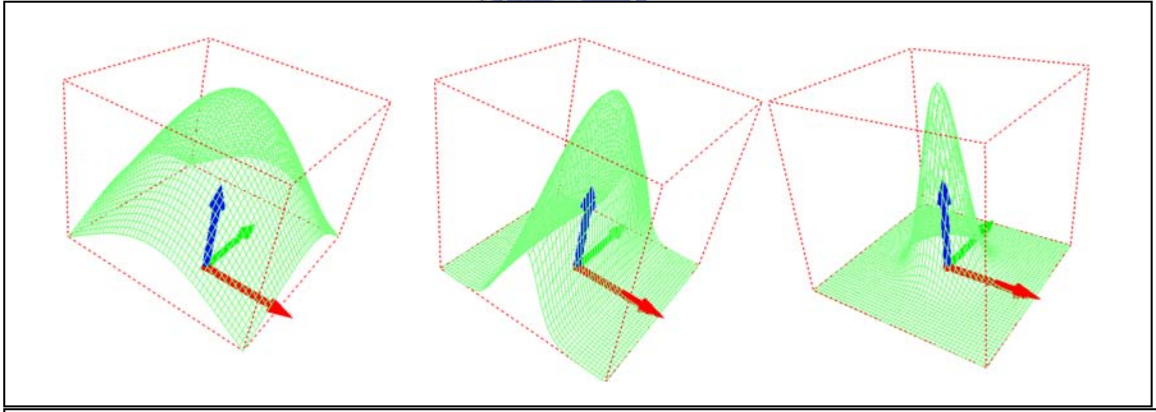


Figure 13: Examples of different shapes of basis. $\boldsymbol{C_s}$ and $\boldsymbol{C_t}$ are 64 in the left one, 64, 4 in middle one and 4, 4 in the right one.

6.3 Approximation

Different composition of centers may have great influence on the compression or approximation results. Since our goal is to approximate the motion variations by radial basis function, we utilize an iterative greedy-algorithm propose by Carr [8]. For each cluster, we choose initial centers to train the radial basis functions network at beginning. If current cluster forms a surface, we choose the corners of surface as our initial centers. If current cluster is a curve, we choose the initial centers at first and last frame. Then we use this function to reconstruct the animation segment. Samples with larger residuals will be chosen as new centers and we re-train the radial basis functions. The iterative step will continue until stop criterions are satisfied. (i.e. approximation error smaller than the predefined threshold.)

Figure 14 is the iteration flowchart which illustrates how we use radial basis functions to approximate motion data gradually. Figure 15 is an example of approximating a curve. The green curve is the real data. The red curve represents the approximated curve. And blue dots mean centers.

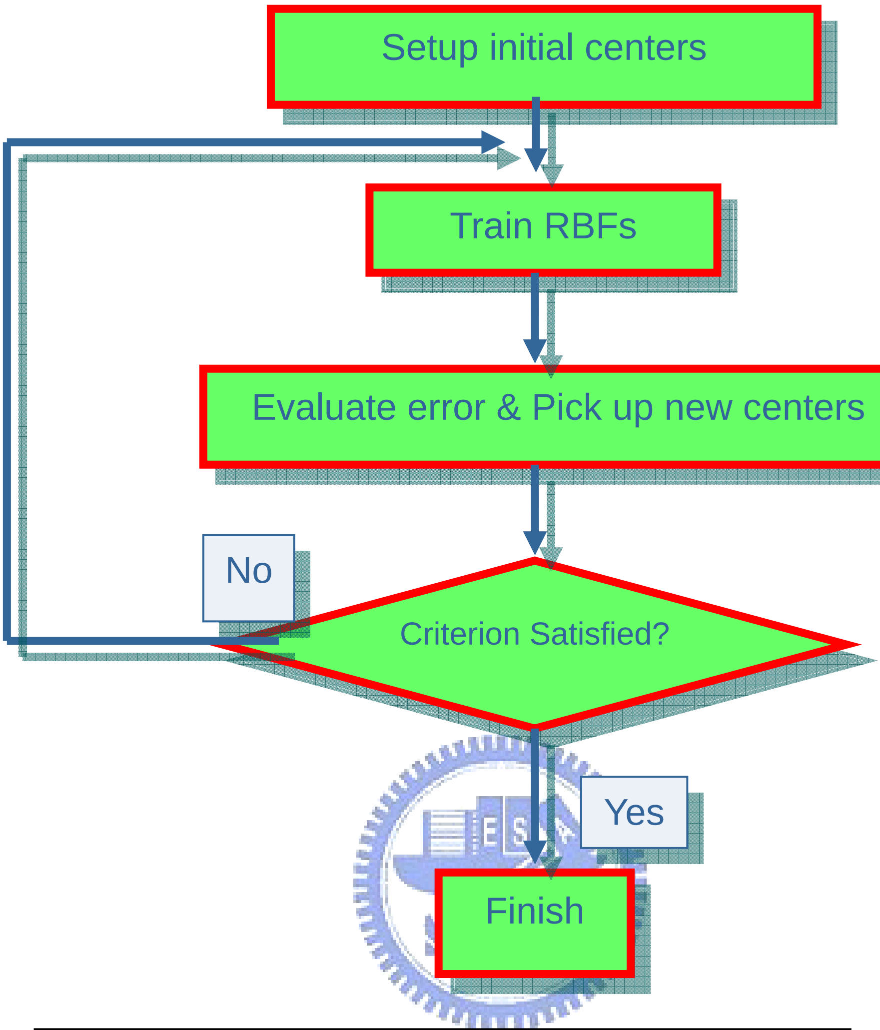


Figure 14: An illustration of the approximation steps.

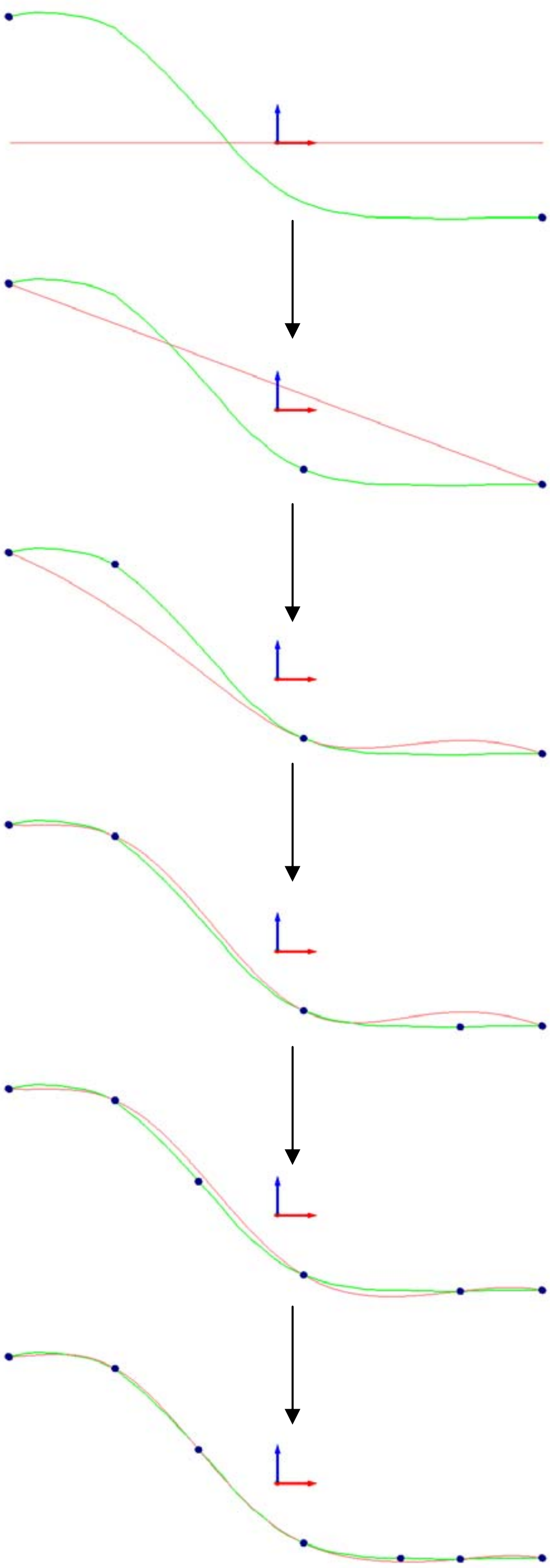


Figure 15: An example of approximating a curve. Green curve and red curve are real data and approximated function respectively. Blue dots are centers of radial basis functions.

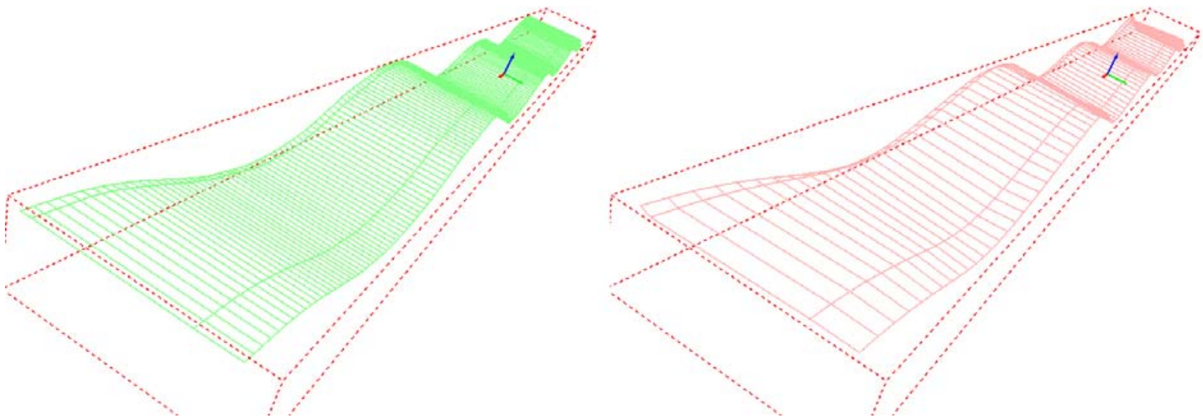


Figure 16: An example of real data surface (green one) and approximated surface (red one).

Experiment and Result

In order to prove our thesis utilizing spatial and temporal coherence, we prepared 16 testing data and design several experiments. The original file format is the commonly used BVH. However, as we mentioned previously, we converted the motion data from hierarchical angular domain to Euclidean system before starting compression. We stored such converted data in "BIN" file format and the file extension is "bin". The BIN file contains the global positions of joints frame by frame in binary. The compressed motion data were encoded in "R" file format. Therefore, our compression ratio is defined as:

Compression_ratio = (size of BIN file)/(size of R file) (12)

Since our thesis is a lossy compression method. We defined the error threshold is 5 cm if the height of the subject is 1.8 m. Such error threshold is sensible hardly for human eyes.

The first experiment compressed these testing data directly. It is worth noting that different motion data have different coherence that we can utilize. Therefore we have to adjust the basis shape (i.e. $\boldsymbol{C_s}$ & $\boldsymbol{C_t}$) to produce better results. Generally speaking, when the target surfaces or curves are smoother, $\boldsymbol{C_t}$ and $\boldsymbol{C_s}$ should be larger such that far fewer centers are needed.

Motion	Cs	Ct	.bin	.r	Ratio
Ballet05	64	256	200,687	5,187	1:38.7
Ballet23	4	256	117,059	3,869	1:30.3
Cowboy3	16	32	191,027	10,127	1:18.9
Cowboy4	16	32	165,083	12,246	1:13.4
Drunk5	16	64	269,135	26,962	1:9.9
Faint5	4	32	95,255	9,571	1:9.9
ShotShoulder03	4	32	52,475	6,146	1:8.5
Sit21	4	256	72,899	5,540	1:13.2
Sneak01	32	128	148,523	8,359	1:17.8
Stand03	32	128	62,963	5,390	1:11.7
Tired05	32	128	203,999	16,622	1:12.3
Walk25*	32	64	248,711	12,774	1:19.5
Walk34*	32	64	241,811	10,987	1:22

Table 1: Compression Results

Second, we compared two compression approaches: a space-time method and a traditional approximation method in temporal domain only. In this experiment, the temporal constant must be same value between these two methods.

Motion	.bin	s-t ratio	.r (space-time)	Ratio (space-time)	.r (time)	Ratio (time)
Ballet05	200,687	2.33	5,187	1:38.7	8136	1:24
Ballet23	117,059	2.22	3,869	40.97	6056	1:19.3
Cowboy3	191,027	0.79	10,127	1:18.9	14472	13.19
Cowboy4	165,083	1.0	12,246	1:13.4	14280	11.56
Drunk05	269,135	0.13	26,962	1:9.9	26436	1:9
Faint05	95,255	0.2	9,571	1:9.9	10624	1:8.97
Shot Shoulder03	52,475	0.12	6,146	1:8.5	6260	1:8.38
Sit21	72,899	0.52	5,540	1:13.2	6,852	1:10.6
Sneak01	148,523	0.68	8,359	1:17.8	9964	1:14.9
Stand03	62,963	0.26	5,390	1:11.7	5768	1:10.91
Tired05	203,999	0.56	16,622	1:12.3	15268	1:13.36

Table 2: The Comparison time domain only and space-time approach.

The third experiment shows how the segment length affects our compression results. Since IPCA will produce uncontrollable segment length, we propose use several fixed length segment to test.

Motion	Frame	Seg. 100	Seg. 200	Seg. 300
Ballet05	728	29.95	39.28	43.45
Ballet23	425	26,65	31.28	33.66
Cowboy3	693	13.5	15.39	15.91
Cowboy4	599	12.6	13.1	14.11
Drunk05	976	8.6	9.1	9.3
Faint05	346	8.85	9.8	10.1
Shot Shoulder03	191	8.35	9.6	9.9
Sit21	265	12.14	13.2	13.4
Sneak01	539	13.14	14.6	15.1
Stand03	229	9.9	11.1	11.4
Tired05	740	10.5	11.3	11.5

Table 3: A discussion of segment length.

Obviously, the compression ratios are better in the longer segments. This is because longer segment may propose more coherence in temporal domain. However, if segments were too long, we may find fewer joints with similar trajectories. In other words, we miss mush spatial coherence. Therefore, the length of each segment is an important tradeoff we have to concern.

The final experiment discussed the performance of our approach. The platform is P4 3.2GHz with 1 GB RAM.

motion	Number of frames	Length (sec)	Compression FPS	Decompression FPS
Ballet05	728	24.27	91	16930
Ballet23	425	14.17	106	21250
Drunk05	976	32.53	8.79	863
Run02	621	20.7	9.26	1321
Run05	571	19	10.38	2719
Shot Shoulder03	191	6.37	13	9550
Sit21	265	8.83	24	13250
Stand03	229	7.6	19.08	11450

Table 4: Algorithm performance

Obviously, the compression time is much longer than decompression time. This is because iteration process. Solving the coefficients of radial basis functions is inverting a large matrix essentially. Fortunately, decompression speed is more important than compression ones in practical. Therefore this thesis is still practicable and feasible.

8. Conclusion

In this thesis, we propose a compression method for human motion data. Unlike previous studies, our method utilizes radial basis functions to approximate motion in both spatial and temporal domain simultaneously. In order to find out more coherence, we analyze motion data in temporal and spatial domain and reorder the joint sequence such that we have a smooth curve or surface.

However, if we can not find out sufficient spatial coherence in motion data, the compression ratio will be close temporal-domain compression only. This is because most clusters form curves. Besides, solving the coefficients of radial basis functions is inversing a matrix essentially. This means the matrix size can not be too large. Although we may find a cluster with much useful coherence, we still need to care about the amount of available samples.

9. Future Work

Important future research directions are listed as follows: First, since our method is a flexible compression component. It can be combined with other compression component to achieve better results. For example, there may be repetitive motion behaviors in some motion data. We can retrieve such motion and represent it more efficiency.

Second, in the typical motion data, the length of bone is fixed and this data is usually known in advance. Thus, we can use it as an additional constraint. In details, we can approximate motion data roughly. During decompression, we may utilize this constraint to enhance the reconstructed joint position more correctly.

Although we proposed a efficient framework to exploit coherence, more sophisticated methods, e.q. PPCA segmentation may improve our analysis.

Hierarchical Radial Basis Functions

Automatically deciding the width of each center is very difficult. Different human motions are quite different in behavior. Therefore, approximating surfaces with same ct and cs can not have a good compression result. Future researches may adopt the hierarchical radial basis functions to overcome this problem. For example, we use wider basis to capture the rough shapes of surfaces and use thicker basis to capture the high-variation part. In other words, we may approximate a surface in several levels. Each level represents the different details or frequencies.

Reference

[1] Alexa, M and Muller, W. 2000, “Representing animations by principal components”, In Eurographics Computer Animation and Simulation, vol. 19, 411-418.

[2] Gaudong Liu, Jingdan Zhang, Wei Wang, and Leonard McMillan, “Human Motion Estimation from a Reduced Marker Set”, Proceedings of the 2006 symposium on Interactive 3D graphics and games, Pages: 35-42, 2006.

[3] Gaudong Liu and McMillan, “Segment-Based Human Motion Compression”, In Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, Pages: 127-135, 2006.

[4] Igor Goskov, and Andrei Khodakovsky, “Wavelet compression of parametrically coherent mesh sequences”, In Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp 183-192, 2004.

[5] Ibarria, L and Rossignac, J, “Dynapack: space-time compression of the 3d animations of triangle meshes with fixed connectivity”, In Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation, Pages: 126-135, 2003.

[6] Jernej Barbic, Alla Safonova, Jia-Yu Pan, Christors Faloutsos, Jessica K Hodgins, and Nancy S Pollard, “Segmenting Motion Capture Data into Distinct Behaviors”, Graphics Interface (2004), pp. 185-194

[7] Jerome Edwars Lengyel, “Compression of time-dependent geometry”, In Proceedings of the 1999 ACM SIGGRAPH Symposium on Interactive 3D Graphics (1999), pp. 89-95

[8] J. C. Carr., R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, “Reconstruction and representation of 3D objects with radial basis functions”, In Proceedings of the 28th annual conference on Computer

graphics and interactive techniques. Pp. 67-76, 2001.

[9] Karni, Z., and Gotsman, C. 2000, “Spectral compression of mesh geometry”, In Proceedings of SIGGRAPH 200, pp. 279-286.

[10] Mirko Sattler, Ralf Sarlette, and Reinhard Klenin, “Simple and efficient compression of animation sequence”, In Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation (2005), pp. 209-217

[11] Okan Arikan. “Compression of Motion Capture Databases”, International Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH 2006), pp 890--897, Boston, USA, 2006.

[12] Ravi R, Sebastian E, and Peter N, “Reflectance Sharing: Predicting Appearance from a Sparse Set of Images of a Known Shape”, In IEEE trans on Pattern Analysis and Machine Intelligence, Pages: 1287-1302, 2006.

[13] Jarek Rossignac, “Edgebreaker: Connectivity Compression for Triangle Meshes”, IEEE Transactions on Visualization and Computer Graphics, vol. 5, pp. 47-61,1999.

[14] Takeo Igarashi, Tomer Moscovich and John F. Hughes, "Spatial Keyframing for Performance-driven Animation", ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Pages: 107-115, 2005.

[15] Tomohiko Mukai, Shigeru Kuriyama, "Geostatistical Motion Interpolation", ACM Transactions on Graphics, Vol. 24, Issue. 3, pp.1062-1070, 2005.7.

[16] Zachi K and Craig G, “Spectral compression of mesh geometry”, In Proceedings of the 27th annual conference on Computer graphics and interactive techniques, Pages: 279-286, 2000.

[17] Zachi Karni, and Craig Cotsman, “Compression of soft-body animation sequences”, Computer and Graphics 28 (2004), pp. 25-34.