**Master Seminar**

**A Generative Statistical Model for Human Motion Synthesis**



Submitted by: Han Du

Supervisor 1: Prof. Dr. Dietrich Klakow

Supervisor 2: Prof. Dr. Philipp Slusallek

Advisor: Dr. Martin Manns

Saarland University 14. April 2014

# Abstract:

*Digital assembly planning and verification is current trend in automotive industry. A realistic human motion synthesis model is a crucial part of this approach. This report firstly introduces the motivation for our work, then reviews the most popular used motion synthesis methods in past two decades. We compare the advantages and disadvantages of these approaches, and finally turn to the approach of Motion Graphs++ [4], which is used in our work. Motion Graphs++ is a motion capture data driven approach. The key idea is to use a discrete graph to model finite structural variations of human motion and represent nodes and edges in the graph as a statistical model to model continuous style variations. In our work, we use a low cost markerless motion capture system to get recorded motion. We will show how this algorithm work with our low cost capture system compared with some very high quality data in our experiments. And we also tried to modify the algorithm a little bit to make it work better on our data.*

**Key words:** *generative motion synthesis, motion graphs, statistical models*

Content

[Abstract: 1](#_Toc385255629)

[1. Introduction 3](#_Toc385255630)

[1.1 Initial Situation 3](#_Toc385255631)

[1.2 Problem Statement 4](#_Toc385255632)

[1.3 Objective 5](#_Toc385255633)

[2 Background 6](#_Toc385255634)

[2.1 Manual Synthesis 6](#_Toc385255635)

[2.2 Parametric Motion Synthesis 6](#_Toc385255636)

[2.2.1 Physically-based Simulation 6](#_Toc385255637)

[2.2.2 Data-based Simulation 7](#_Toc385255638)

[2.3 Motion Capture Data driven Motion Synthesis 7](#_Toc385255639)

[2.3.1 Motion Blending 7](#_Toc385255640)

[2.3.2 Motion Graphs 8](#_Toc385255641)

[2.3.3 Statistical Model 9](#_Toc385255642)

[3 Motion Graphs++ 10](#_Toc385255643)

[3.1 Preprocessing Motion Capture Data 10](#_Toc385255644)

[3.2 Morphable Graph Modeling 10](#_Toc385255645)

[3.3 Motion Synthesis 11](#_Toc385255646)

[4 Experiments 11](#_Toc385255647)

[4.1 Automatic Motion Decomposition 11](#_Toc385255648)

[4.2 PCA performance on High Quality Data and Low Quality Data 14](#_Toc385255649)

[5 Summary & Future Work 15](#_Toc385255650)

[1.1: (a) Delmia V5, (b) EMA from IMK-automotive GmbH 4](file:///E:\thesis\master%20thesis\master%20seminar\slides%20for%20master%20seminar\Master%20Seminar%20Report.docx#_Toc388605478)

# Introduction

## 1.1 Initial Situation

With the fast development of global economy and technique, the contradiction between the requirement of shortening product life cycle and the long product planning and verification cycle becomes a critical problem for automotive industry. The keen global competition in automotive industry requires car manufacturers increase the variety of products, while maintaining high quality and efficiency at the same time. In order to meet these requirement, product and assembly system are usually developed in parallel today.

The traditional way to build an assembly system for a product is to build a physical prototype for it. A physical prototype defines the requirements for the whole assembly task, for instance how to determine the worker’s walking path, how to place workbench, which components will be used and the time analysis and ergonomic evaluation and so on. However, with the increase of the complexity of products, to make clear answers to these questions is not a trivial task, due to it is very hard to consider all the possible cases on a very early stage of development. In addition, physical simulation and verification for each prototype is time-consuming and costly. And there is no guarantee for the later products because it is nearly impossible to simulate all possible situations by physical simulation at the very early stage. Another problem is that each prototype is specific for one product, which is extremely unsuitable for today’s requirement of large variety of products.

One promising approach which can avoid using physical prototype is to use digital tools for product planning and verification. A lot of digital modeling and simulation tools have been developed for this purpose. Basically, a good digital simulation tool should able to reconstruct the assembly scenario in a very realistic and physically accurate way, especially for realistic human motion simulation. Because today’s procedure of assembling automobiles still needs many manual works. Several different approaches have already been proposed for this target.

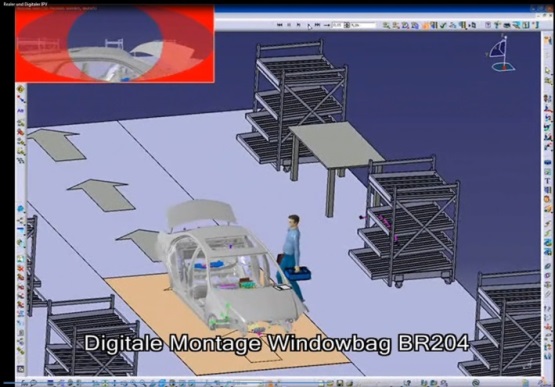
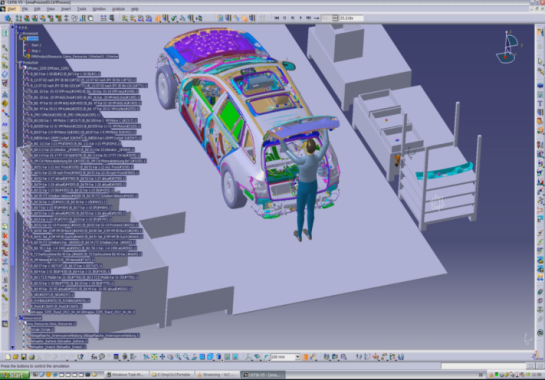
One of the earliest methods is manual geometric modeling [1]. The user needs to explicitly tune the parameters of human skeleton within physical-valid range to generate the simulation. However, the effort for modeling an accurate human motion is very high. For instance, one minute simulation could require several hours modeling. And the synthesized motion looks unnatural, like robots. Figure 1.1(a) shows an example of this approach, Delmia V5, a CATIA plugin tool from Daimler AG.

In order to reduce this high modeling effort, one possible solution is automatic motion synthesis. It regards the motion parameters of each DOF as a single time varying signal [2]. An analytical function is defined for each joint, which will control the change of parameters of each joint. Motion captured data can also be used to train the parameters of analytical function in order to get natural looking result. One example of this approach is EMA (Editor Human Labor), one simulation software from IMK-automotive GmbH, shown by figure 1.1(b). EMA can provide a virtual modeling and simulation environment for assembly task, and saving a lot of efforts and time compared with Delmia V5. However, the evaluation from real users shows that the quality of result is not sufficient. The actions from virtual worker are not very precise, not collision awareness, for example, the hand of virtual worker could go through objects sometimes, unnatural and even unrealistic at times. So, in general, parametric based motion synthesis approaches cannot generate physically-valid, accurate and natural looking results for simulation and verification.

(a)

(b)

Figure 1.1: (a) Delmia V5, (b) EMA from IMK-automotive GmbH [both pictures are from Daimler AG, EE/IPA]



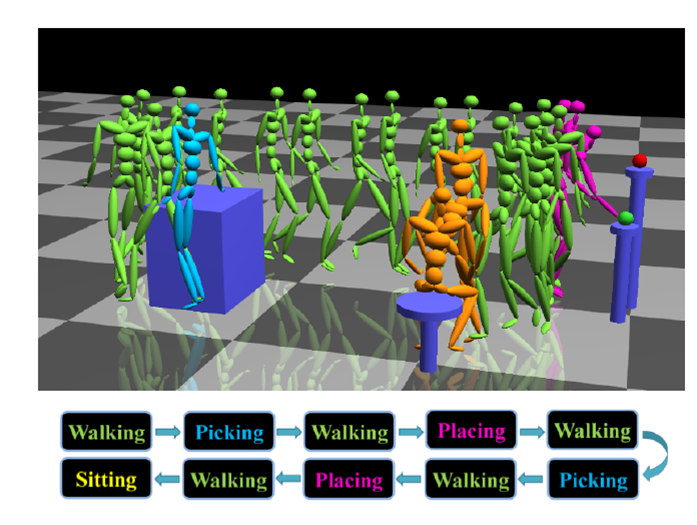
## 1.2 Problem Statement

Because of the deficiencies of analytic motion synthesis, another common solution for motion generation is motion capture. Motion capture is a reliable way to get realistic human motion. However, directly using motion capture data for animation is far away from enough, because motion capture data is proven to be hard to modify. The main problems of using motion capture data for animation are:

* Most motion capture systems are expensive to use, because for each specific task, we need to capture motions which satisfy the requirement. The processing time for motion data is usually high and motion data is not easy to be re-used.
* Some kinds of motion, for instance, special effects in movie and game industry, are very hard to perform by actors.

So some algorithms which can edit, re-use motion capture data to create continuous streams of motion are needed. Motion Graphs [3] takes the idea that decomposes the captured motion data into small chips, then creates a new motion by concatenating these small pieces with some proper methods. Motion graph transforms the motion synthesis problem into a selecting problem, search on the motion capture database. If the captured motion data is large enough, it theoretically can generate arbitrary motions. However, Motion Graphs just uses the motion from prerecorded data, which means that it will not have a rich variation of styles. If we want to enjoy a rich repertoire of poses, which is more natural and realistic, we need a method which can generate infinite motion samples from finite prerecorded motion data. Motion Graphs++ [4] assumes that the high dimensional human motion data has some intrinsic coordination which can be represented in a more compact way and obey some statistical distribution. They also decompose motion capture data into motion segments and classify them into each pre-defined motion primitive. Then they build a statistical model based on these recorded motion samples so that infinite motion samples can be generated just by sampling this statistical model. They also assume that the transition between two motion primitives is followed some statistical distribution, and construct this statistical transition model based on prerecorded motion data. By doing this, this approach should be able to synthesize continuous style variations motion.

Figure 1.2: Semantic motion analysis and synthesis. [1]

The task of our work is to implement Motion Graph++ algorithm on our motion capture system, which uses BVH file format to store motion data. We compare synthesized motion with pre-captured motion, and analyze the performance of this statistical approach under different types of noise.

## 1.3 Objective

Our final objective is to construct an automatic, scalable, semantically interactive and natural-looking motion synthesis model for digital simulation and verification of assembly system. As shown in figure 1.2, the model should be highly automatic. User can interact with it by some high level semantic inputs. Then the modeling and simulation part should be done purely automated, without any manual configuration from users. Good scalability is also necessary. The model should be able to generate motion which contains multi kinds of action. And natural-looking is guaranteed by the statistical model. For the motion segments from the same motion primitive, they are still different because of random sampling. Our work will mainly concern on walking synthesis. We are going to synthesize some random natural-looking walking based on our motion capture database, and analyze the performance of this model under different kinds of noise. A user specified generative motion under environment constraint will not be discussed here, due to the limit of thesis time.

# 2 Background

The study of human motion synthesis has a long history in movie and video game industry. This chapter will give an overview of most popular used motion synthesis methods in past two decades. In general, existing motion synthesis methods can be divided into three categories: manual synthesis, parametric motion synthesis and motion capture data driven motion synthesis.

## 2.1 Manual Synthesis

Manual synthesis is the oldest motion synthesis approach, which is widely used in early cartoon movie and game industries. It manually specifies the parameters of degrees of freedom (DOFs) for a pre-defined human skeleton, which are called as key frames. The frames between these manually drawn key frames are generated through some interpolation methods, for instance, linear interpolation. However, simple interpolation rarely gives realistic results unless the key frames are really dense. This work is tedious and time-consuming, not only because a lot of key frames must be created, but also because it is challenging to draw good key frames which are realistic and smooth when played in sequence. Sometimes it requires some expertise of art. A practical example of this approach is Delmia V5. As we discussed before, the result is not very natural and the simulation requires huge manually modeling work.

## 2.2 Parametric Motion Synthesis

As mentioned above, generating motion simply by interpolating usually give us unrealistic results, because the real and natural human movement is governed by many constraints, e.g. the laws of physics, custom of common behaviors and so on. Parametric motion synthesis embeds these constraints into some controllers which can control the whole skeleton or each joint. The control systems take input from high-level specifications, such as users' constraints or environmental constraints, then automatically generate the lower-level details of the motion. In manual synthesis, the animator must specify many of the lower-level details by hand. In order to achieve this, each controller uses one or several analytical functions to control the parameters of degrees of freedom (DOFs). The analytical functions can either set by physically valid constraints or learned from motion capture data.

### 2.2.1 Physically-based Simulation

The approach to use physical laws to control the movement of body is widely used in robots. Given the mass distribution for each body part and the torques generated by each joint, Newton's laws provide a system of ordinary differential equations that can be integrated to yield joint trajectories. However, while the mass distributions can be obtained from biomechanics literature [5], it is proven considerably hard to find proper joint torques that yield a particular motion. Hodgins et al. [6] proposed handcrafting joint controllers based on finite state machines and they demonstrated this approach on several motions, including running, bicycling and vaulting. Similar approaches [7,8] were used to generate gymnastic motions like flipping or tumbling. Although this method has successfully created certain kinds of motion, a general way to design and select controller for different kinds of motion is still a difficult problem. Each joint in human skeleton is corresponding to one controller, and each controller could have different control strategies for different actions. How to automatically change control strategies when actions change is a nontrivial task. Faloutsos et al. [9] applied support vector machines to learn the conditions under which controllers for different actions can be composed, then switching control strategies to transition between actions. However, physically based simulation can automatically generate physically realistic motions, a motion that obeys physical laws does not mean it is natural.

### 2.2.2 Data-based Simulation

Rather than explicitly representing physical laws as equations to each controller, some other work employed motion capture data to train the parameters of each analytical function. The idea is that as the naturalness and physically valid are implicitly represented in real human motion, the analytical functions which can model the motion capture data well should able to generate realistic human motion while preserving naturalness meanwhile. A real example using this approach is the digital simulation and planning tool EMA, as we discussed before. However, data-based simulation can only provide a relatively narrow range of realistic motion.

Compared with manual synthesis, parametric motion synthesis can save a lot of manual modeling work. It also enables creation of autonomous actors that can react to user or the environment using higher-level control systems. However, despite the potential advantages of parametric motion synthesis, robust control systems which can generate physically-valid and natural-looking motion are difficult to design by hand. And automatic methods are not yet sufficient enough to generate control systems for most human behaviors.

## 2.3 Motion Capture Data driven Motion Synthesis

With the development of motion capture techniques, Motion capture data driven motion synthesis has caused more and more attention over the past two decades. Motion capture technology can provide a source of highly realistic example motions, which can serve as raw material for a variety of data driven synthesis methods. Compared with parametric motion synthesis, data driven approaches have the advantages of natural-looking, realistic and more generative. There is no need to build complex control systems for each joint. Many algorithms have been proposed to synthesize new motion based on motion capture data. In this section, we will go through the state of arts of data driven motion synthesis algorithms.

### 2.3.1 Motion Blending

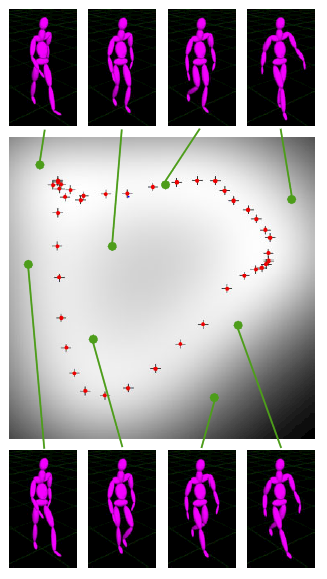
The simplest way to create a new motion from pre-recorded motion is to just concatenate two motion and blend the transition part. Figure 2.1 (a) shows a combination of running and walking. However, simply putting one motion after another could cause two problems. One is the motion discontinuity at transition point, another is two motion could be in different speed scales, for instance, combining a slow running and a fast walking. For the first problem, it is usually addressed by motion blending [6], which can create seamless transitions between motions. The choice of interpolate function depends on the motion capture data. For instance, if the motion capture data is 3D coordinates of joints, then linear interpolation [7] can generate good transition. If the motion capture data is represented by relative orientation angles, like BVH file, radial basis functions [8] can be used to blend two motion examples. For the second problem, motions with different timing may be timewarped such that logically related events occur simultaneously. This mapping between corresponding frame index of motions indices as a timewarp curve. we can use dynamic time warping algorithm to pre-processing motion data, registering one motion to another [14].

(a)

(b)

(c)

Figure 2.1: (a) motion blending between running and walking [14], (b) a walking motion generated to fit a specified path by motion graph [3], (c) random pose synthesis from statistical model [25]

However, for synthesizing some special motions, sometimes, it is not easy to find suitable motion capture. It is always necessary to capture new data for this purpose. So just simply blending and interpolation cannot make use of motion capture data efficiently.

### 2.3.2 Motion Graphs

Motion graphs (sometimes called "move trees") has been used in the video game industry for a long time for the purpose of character control [13]. Most motions more than a few seconds in duration are naturally thought of as sequences of atomic actions. For example, walking consists of an alternating sequence of left stance and right stance. Motion graphs will generate new motions by re-connecting these atomic actions. In the early research, motion graphs have been constructed manually in the sense that users explicitly decided which motion clips would be connected [15, 16]. Given a motion capture data set, an animator identifies places where motions can connect together, then uses tools to adjust these motions so they will join seamlessly at these points, a tedious and time-consuming process.

Automated graph-based motion synthesis has been the subject of a considerable amount of previous work. The crucial point for automatically constructing motion graphs is to automatically identified transition locations. Kovar et al. [3], Lee et al. [18] and Arikan and Forsyth [17] all used a distance metric to find transition locations and then applied search algorithms on the resulting graph to extract motions that satisfied user-defined constraints. After selecting appropriate sequence of motions, to generate transitions is also an important part of building motion graphs. Perlin [19], Kovar et al [3] used a simple linear blending method to concatenate two motion clips. There are also many other transition methods, such as displacement maps [17, 18, 20], the torque-minimization algorithm of Rose et al. [21] or the blending method introduced in previous section. Figure 2.1 (b) shows an example of walking motion followed a user specified path generated by motion graphs.

Motion graphs can not only concatenate two motion samples as motion blending, but generate continuous streams of motion under user specified constrains as well. This approach is more flexible and can make use of motion capture data more efficient than simply blending. It is also possible to combine environment constrains or user specified constraints, for instance, sneak walking in this approach. As motion graphs transfers motion synthesis problem into a directed graph search problem, so all the requirement can be formulated as constraints in an optimization problem.

However, there are some limits of motion graphs. First of all, the synthesized motion is from database of motion capture. So it cannot have a rich variation of pose. Secondly, at the transition point, the thresholds for similarity must be specified by hand, because different kinds of motion have different fidelity requirements. For the motions which are very familiar to people, such as walking, running and so on, people are very sensitive to discontinuity. So a high similarity threshold is required. But for some motions are less familiar for people, maybe a low similarity is enough.

### 2.3.3 Statistical Model

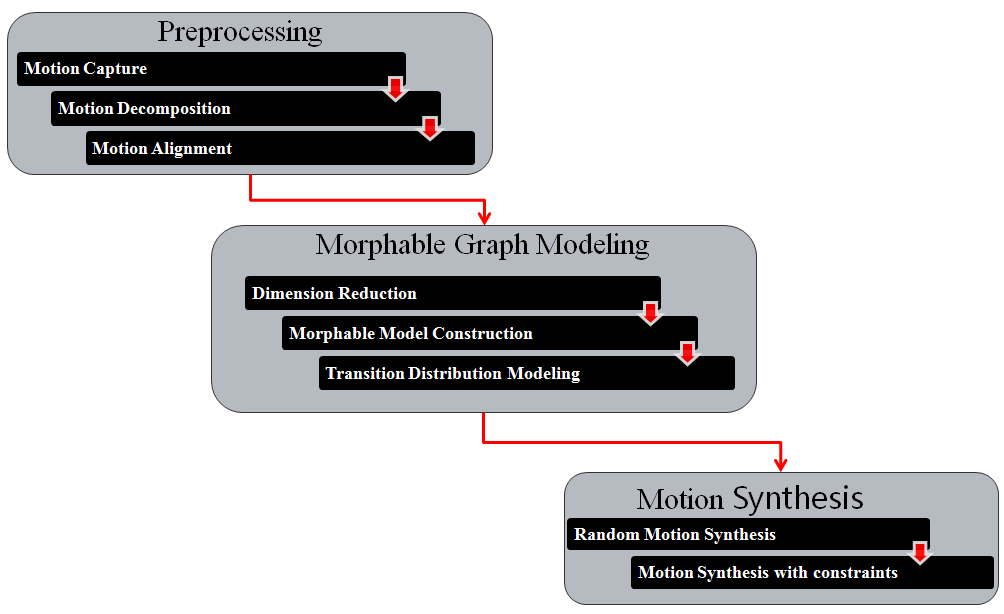
The method is inspired by above mentioned problem, how to generate a variant of different motions from limited number of captured motions. The assumption is although human motion looks quite arbitrary and could have infinite styles, different poses of one motion should have some intrinsic relationship, which can be model by some statistical models. The models are usually represented as a set of mathematical equations or functions that describe human motion using a finite number of parameters and their associated probability distribution. So the task of motion synthesis is converted to construct a proper probability distribution model, which can well fit the finite motion capture data. A new variant motion can be generated by sampling the distribution. Theoretically, this approach can generate any possible poses of motion, but usually the motions which are most similar to the training data are taken. A lot of researches have been invested in this field. Grochow et al. [25] built an inverse kinematics system based on a learned model of different human poses. They stated that this system can generate any pose which satisfies the given constraints. Bowden [12], Brand and Hertzmann [22], Galata et al. [23] and Li et al. [24] combined motion graphs with statistical models. They have built graph-based statistical models. Each node in the graph is a simple generator of poses which is corresponding to a statistical model, and each edge represents possible transitions between different kinds of poses. Figure 2.1 (c) is an example of using statistical model to create different pose of walking. However, a fundamental limitation of statistical 

Figure 3.1: Pipeline of Motion Graphs++

model approach is that it lacks the ability to control motion in a very precise way, and does not consider physical force that causes the motion.

# 3 Motion Graphs++

In our work, we use the approach called Motion Graphs++ [4], which is a combination of Motion Graphs and statistical models. Similar to Motion Graphs, Motion Graphs++ represents motion synthesis problem as directed graph search problem. However, the difference is in Motion Graphs++, node and edge are both corresponding to a statistical model. The procedure of Motion Graphs++ is shown in figure 3.1. Here, we will briefly introduce the outline of modeling process, the details will be given in my thesis.

## 3.1 Preprocessing Motion Capture Data

For captured motion data, we first look for important keyframes (e.g. left foot strike for walking) in each motion data and use these keyframes to decompose each motion sequence into distinctive motion segments. Then motion segments will be automatically placed into the same motion primitives if they share the same starting and ending keyframes.

For motion segments in each primitive, they could have different number of frames, different orientation and position. So we register all motion segments in each primitive to a selected reference motion segment. Registered motion segments in each primitive should more similar to each other and suitable for generative statistical modeling.

## 3.2 Morphable Graph Modeling

For the purpose of generative statistical modeling, each motion segment is represented as a long vector by sequentially concatenating the parameters of each frame. So firstly, we apply principal component analysis to reduce the linear redundancy, represent motion segment in a compact way.

(3.1)

where is the raw motion segment, is the mean motion, *P* is a matrix of first n eigenvectors and is the low dimensional vector for the i-th motion primitive.

Then we organize all the motion primitives into a finite directed graph . Each node in the graph stores one motion primitive with a statistical model of a prior distribution function , and each edge *(i,j*) represents a valid transition from motion primitive *i* to primitive *j.* And a transition distribution is constructed using Gaussian Processes to model possible transitions.

## 3.3 Motion Synthesis

For random motion synthesis, the algorithm firstly random generate a graph walk in motion graph, then generate first motion segment by random sampling the morphable model of first motion primitive. For the following nodes, the corresponding motion segments will be generated by transition models.

For constrained motion synthesis, basically, there are two kinds of constraints: user specified constraints and environment constraints ***e***. Then the motion synthesis problem can be formulated as an optimization problem in a Maximum A Posterior (MAP) framework. In general, we want an optimal path of the morphable nodes as well as the optimal values for corresponding morphable parameters by solving the following MAP problem:

(3.2)

# 4 Experiments

In this chapter, some initial experiments we have implemented based on our motion capture data set will be discussed. The motion capture system we use is a low cost markerless camera motion capture system from MPII. And the mocap data is stored in BioVision (BVH) file format, which stores absolute root position and relative rotation angles for each joint in a hierarchical structure. Our test scenario is 4 meter's straight line walking in a workshop. There are 145 files recorded, 23467 frames in total. And we also have a very high quality data from Wicon motion capture system for the same scenario, which can be regarded as ground truth. There are 30993 frames in total.

## 4.1 Automatic Motion Decomposition

As we mentioned before, the first step to process mocap data is to decompose them into atomic motion segments. This is achieved by identifying pre-defined keyframes in the frame sequence. Then the coming question is how to extract these keyframes from frame sequence. The intuitive way to do this is manually labeling keyframes. However, although the result is good, the work is tedious and time-consuming. In our case, it took 8 hours to annotate 23467 test frames. So extracting keyframes by some automatic way is needed.

In our work, we define two kinds of keyframes:

* Contact state transitions occur

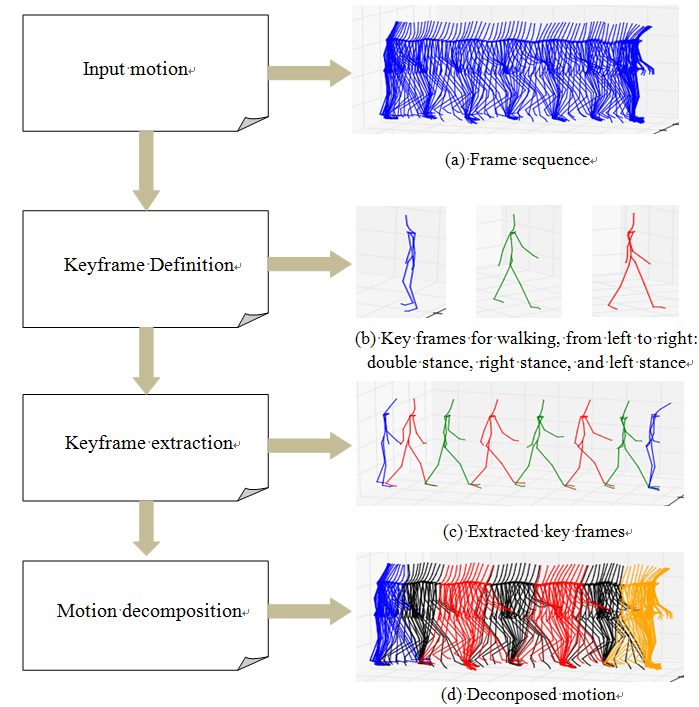


Figure 4.1: Work-flow of motion decomposition

* Highest visual content changes

For contact state transitions, it is quite clear to tell when it happens. It can be automatically found by using some physical sensors to detect these contact points or using some dominant features in visualization.

For highest visual content changes, it is not easy to give clear definition of highest visual content. Different people could have different opinion about that. So for this kind of keyframes, we can manually annotate keyframes in a training set, then using some distance measure to find the most similar frames in remaining motion segments.

In our experiment, we only consider a very simple motion -- walking. As walking can simply be regarded as a sequence of alternating double and single-stance phases, so we only define three keyframes for walking, as shown in figure 4.1, and use a dominant visual feature feet distance to automatically extract keyframes. Figure 4.2 shows the feet distance measure of an example walking motion. The local maximum within stepsize above the given threshold will be regarded as a keyframe. In our implementation, the threshold will be automatically decided, so only parameter needed to manually set is stepsize. Here, for quality measure, we take similar measure terms in information retrieval shown in table 4.1. And we define two criteria to measure how good our approach in low quality data and high quality data. Accuracy measures how accurate the extraction result is and distortion measures the difference between

Figure 4.2: Feet distance measure of an example motion

|  |  |  |
| --- | --- | --- |
|  | keyframe | not keyframe |
| extracted | true positive (tp) | false positive (fp) |
| not extracted | false negative (fn) | true negative (tn) |

Table 4.1: Extraction measure

Accuracy: (4.1)

Distortion rate: (4.2)

where is the i-th correctly automatically extracted keyframe, is the i-th corresponding manually extracted keyframe, and *N* is the total number of correctly extract keyframes

manually extraction and automatically extraction. A good extraction result would has high accuracy and low distortion.

Figure 4.3 illustrates the influence of different choices of stepsize for keyframe extraction and the optimal stepsize for each. Form the figures we can see that this feature based keyframe extraction approach works well for both low quality data and high quality data. Table 4.2 shows that for walking, by taking correct stepsize, both can achieve 100% accuracy. However, high quality data has better performance in distortion than low quality data due to high quality data is more accurate and stable than low quality data.

|  |  |  |
| --- | --- | --- |
|  | Low quality data | High quality data |
| Accuracy | 100% | 100% |
| distortion | 2.3 | 0.9 |

Table 4.2: Accuracy and distortion measures for two datasets.

Based on the experiment results, we can conclude that for simple motions, e.g. walking, running, using one or two dominant features to detecting contact transitions keyframes works well. And this approach works better for high quality data because these dominant features are more stable and accurate in high quality data.

(a) markerless mocap system

(b) wicon mocap system

Figure 4.3: Extraction results with different stepsize for low quality data and high quality data.

## 4.2 PCA performance on High Quality Data and Low Quality Data

From real synthesis results, which will not presented in this report due to limit of space, we find out that the result from low quality data is much worse than high quality data. So we want to define a metric to measure the quality of motion capture system and find out which kind of quality is good enough for our approach.

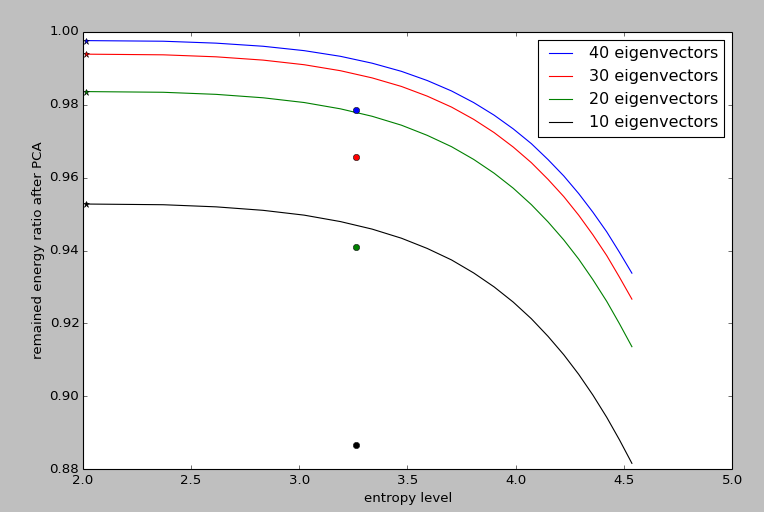
As dimension reduction is very important step for our approach, in this section we will 

Figure 4.4: PCA performance of high quality data and low quality data. Each curve represents the high quality data with increasing noise level. With the increase of noise, the entropy of the motion data increases and remained energy after PCA decreases. Each dot represents low quality data with different number of eigenvectors, the color has same meaning as curves.

estimate the performance of PCA with respect to high quality data and low quality data. We assume that the low quality data is a noisy data from the view of high quality data. The experiment is that we select a set of high quality motion data, add random Gaussian noise with different variance level. Then compute the average entropy of motion signal and average remained energy ratio after PCA. And we run the test with different number of eigenvectors. For the low quality data, the same procedure is taken except for adding artifical noise. Figure 4.4 plots the results. From the results we can conclude that PCA works better with high quality data than low quality data. The less eigenvector used, the larger difference. And random Gaussian noise seems not a realistic noise for the low quality data. The performance of adding random Gaussian noise is quite different from the performance of low quality data. This may because the motion capture system already smoothed random Gaussian noise, and the low quality result is caused by some other factors.

# 5 Summary & Future Work

In this report, we mainly force on the motivation of our work and give an overview of related works. The methodology for our work and some experiments are briefly introduced due to the limit of space. The Motion Graphs++ algorithm was tried on a simple scenario -- walking, and we found out that this approach was quite generative. Standard statistical modeling approach can be used to model the motion distribution, without any knowledge about kinematic constraints or physical constraints. And natural-looking motion can be generated by simply sampling the distribution, rather than manually setting any parameters. However, these statistical approach has a rigid requirement for quality of training data. From the experiments, we found out that the model from low quality data has a high probability to give us a unrealistic motion. So finding a good metric to tell which kinds of quality of motion capture data is enough for this statistical approach is worth to investigate.

Until to now, this approach has been proven useful to build a statistical model based on high quality motion capture data, which can well represent walking. And we test the model by generating some random walking and the results are realistic and natural-looking. The next step that we are going to do is to embed user's specified constrains and environmental constraints into our statistical motion synthesis model, which can make the synthesized motions more useful. And as we mentioned in our goal, the model should be scalable. So more kinds of motion data will be captured and tested by this statistical modeling approach. In addition, we are constructing GUI framework for users to interact with our statistical motion synthesis model now. Hopeful, we could have some real application in the future.

# Reference:

[1] PERLIN, K., AND GOLDBERG, A. 1996. Improv: A system for scripting interactive actors in virtual worlds. In ACM SIGGRAPH.

[2] Bruderlin A. and Williams L.: ‘Motion signal processing’, in Proceedings of the 22nd annual conference on Computer Graphics and Interactive Techniques, pp. 97-104, 1995.

[3] KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. In ACM Transactions on Graphics. 21(3):473–482.

[4] MIN, J., and CHAI, J. 2012. [Motion Graphs++: A Compact Generative Model for Semantic Motion Analysis and Synthesis](http://students.cse.tamu.edu/jianyuan/project_semantic.html). In ACM Transactions on Graphics, 31(6)

[5] D.Winter. Biomechanics and Motor Control of Human Movement. Wiley, New York, 1990.

[6] J. Hodgins,W.Wooten, D. Brogan, and J. O’Brien. Animating human athletics. In Proceedings of ACM SIGGRAPH 95, Annual Conference Series, pages 71–78. ACM SIGGRAPH,August 1995.

[7] W.Wooten and J. Hodgins. Dynamic simulation of human diving. In Proceedings of Graphics Interface (GI’95), pages 1–9, May 1995.

[8] W. Wooten and J. Hodgins. Simulating leaping, tumbling, landing, and balancing humans. In IEEE International Conference on Robotics and Animation, volume 1, pages 656–662, 2000.

[9] BOWDEN, R. Learning non-linear Models of Shape and Motion , PhD Thesis, Dept Systems Engineering, Brunel University, Uxbridge, Middlesex, UB8 3PH, UK. December 99.

[10] S. Theodore. Understanding animation blending. Game Developer, pages 30–35, May 2002.

[11] WILEY, D., AND HAHN, J. 1997. Interpolation synthesis of articulated figure motion. IEEE Computer Graphics and Application 17, 6, 39.45.

[12] R. Bowden. Learning statistical models of human motion. In IEEE Workshop on Human Modelling, Analysis, and Synthesis, CVPR 2000. IEEE Computer Society, June 2000.

[13] M. Mizuguchi, J. Buchanan, and T. Calvert. Data driven motion transitions for interactive games. In Eurographics 2001 Short Presentations, September 2001.

[14] KOVAR, L., AND GLEICHER, M. 2003. Flexible automatic motion blending with registration curves. In ACM SIGGRAPH/EUROGRAPH Symposium on Computer Animation. 214–224.

[15] K. Perlin. Real time responsive animation with personality. IEEE Transactions on Visualization and Computer Graphics, 1(1):5–15, March 1995.

[16] C. Rose, M. Cohen, and B. Bodenheimer. Verbs and adverbs: multidimensional motion interpolation. IEEE Computer Graphics and Application, 18(5):32–40, 1998. [17] O. Arikan and D. A. Forsyth. Interactive motion generation from examples. ACM Transactions on Graphics, 21(3):483–490, 2002.

[18] J. Lee, J. Chai, P. Reitsma, J. Hodgins, and N. Pollard. Interactive control of avatars animated with human motion data. ACM Transactions on Graphics, 21(3):491–500, 2002.

[19] K. Perlin. Real time responsive animation with personality. IEEE Transactions on Visualization and Computer Graphics, 1(1):5–15, March 1995.

[20] K. Pullen and C. Bregler. Motion capture assisted animation: texturing and synthesis. ACM Transactions on Graphics, 22(3):501–508, 2003.

[21] C. Rose, B. Guenter, B. Bodenheimer, and M. Cohen. Efficient generation of motion transitions using spacetime constraints. In Proceedings of ACM SIGGRAPH 1996, Annual Conference Series, pages 147–154. ACM SIGGRAPH, August 1996.

[22] M. Brand and A. Hertzmann. Style machines. In Proceedings of ACM SIGGRAPH 2000, Annual Conference Series, pages 183–192. ACM SIGGRAPH, July 2000.

[23] A. Galata, N. Jognson, and D. Hogg. Learning variable-length markov models of behavior. Computer Vision and Image Understanding Journal, 81(3):398–413, 2001.

[24] Y. Li, T.Wang, and H.-Y. Shum. Motion texture: A two-level statistical model for character motion synthesis. ACM Transactions on Graphics, 21(3):465–472, 2002

[25] K. GROCHOW, S. L. MARTIN, A. HERTZMANN, AND Z. POPOVI´C ,2004. Style-based inverse kinematics. In ACM Transactions on Graphics. 23(3):522–531.