SYNTHESIS OF DANCE PERFORMANCE BASED ON ANALYSES OF HUMAN MOTION AND MUSIC

人体動作と音楽の解析に基づく舞踊動作生成

BY

TAKAAKI SHIRATORI

A DOCTORAL DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL OF
THE UNIVERSITY OF TOKYO

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF INFORMATION SCIENCE AND TECHNOLOGY

DECEMBER 2006

Committee:

Hiroshi HARASHIMA (Chair)
Masaru KITSUREGAWA
Mitsuru ISHIZUKA
Yoichi SATO
Shunsuke KAMIJO

Supervisor:

Katsushi IKEUCHI

ABSTRACT

Recently, demands for synthesizing realistic human motions are rapidly increasing in computer graphics (CG) and robotics fields. One of the easy solutions to this issue is to use a motion capture system. However, it still remains difficult to capture the motion data that animators really want, and most prior work aimed to solve this problem by editing motion capture data, seamlessly blending or connecting motion capture data sets, or modifying them according to physical properties.

In most cases, human movements, however, are induced by external signals: people first receive visual information such as environmental obstacles from eyes, or audio information such as speech or music from ears, and then recognize essential information or feel some emotions from the obtained information, and finally perform movements. Considering these aspects makes it possible to automatically synthesize more human-like motion, and, despite this possibility, only a few methods considering these aspects have been developed.

To meet this need, we are focusing on dance performance as an experimental subject. Dance performance strongly depends on musical features such as rhythm, speed, mood, intensity, or genre of played music recognized by dance performers, and is well-suited to the issue. The ultimate goal of our study is to realize dancing-to-music ability for CG characters and humanoid robots.

This dissertation describes three novel studies.

The first study is to analyze the relationship between motion and musical rhythm. According to observation of human dance motion, motion rhythm is represented with stop motion called a *keypose*, at which dancers clearly stop their movements, and the motion rhythm is synchronized with musical rhythm to perform dance performance. The proposed method aims to reveal the relationship and consists of music analysis step that estimates musical rhythm, and motion analysis step that extract keypose candidates. By integrating these information, keyposes that are very similar to dancers' understandings are extracted.

The second study is to model how to modify upper body motion based on the speed of played music. When we observed structured dance motion performed at a normal music playback speed and motion performed at faster music playback speed, we found that the detail of each motion is slightly different while the whole of the dance motion is similar in both cases. This phenomenon is derived from the fact that dancers omit the details and perform the essential part of the dance in order to follow the faster music speed. To prove this, we analyzed the motion differences in the frequency domain, and obtained two insights on the omission of motion details: (1) The keyposes mentioned in the first

study are preserved, and (2) High frequency components are gradually reduced depending on the musical speed. Based on these insights, we modeled the motion modification using musical rhythm and kinematic constraints that humans have. We show the effectiveness of our algorithm through experimental results. Additionally, we also developed some applications for CG character animation and humanoid robot motion generation.

The third study is to automatically synthesize dance performance that is well matched to input music. People feel various emotions depending on musical mood. For example, people feel quiet and relaxed when listening to relaxing music such as a ballad, and they feel excited when listening to intense music such as hard rock music. We observed dance performance, especially original dance, and found that the same is often true for dance performance. Based on this, we designed an algorithm to synthesize new dance performance by assuming the relationship between motion and music rhythm mentioned in the first study, and the relationship between motion and music intensity. As for motion synthesis step, we propose two methods: a globally optimal method and a locally optimal method. Users can select one of them depending on their purposes.

Our studies have many advances over prior work on human motion analysis and synthesis. They contribute to not only entertainment systems of CG animation and humanoid robots, but also applications for digital archive of intangible cultural heritages.

論文要旨

近年コンピュータグラフィクス (CG) やロボティクスの分野では，自然な人体動作を生成することの需要が高まってきている．モーションキャプチャシステムはその解決策の一つであるが，アニメータが本当に必要としている動きを得ることは依然難しく，得られたモーションキャプチャデータをさらに加工，編集しなければならないケースが多い．そのため既存の研究では一つの動きデータを加工する手法や，複数の動きデータを滑らかに連結する手法，力学的拘束を満たすための動作変形手法などが主に提案されてきている．

しかし実際の人間の行動を観察すると，まず環境などの視覚情報や音声・音楽などの音響情報を知覚し，そしてその情報の中からから必要なもののみを抽出したり情報に対する感情が生まれ，その結果行動を起こす場面が多い．このような人間の情報抽出能力や感情などを考慮した人体動作の生成手法が求められてきているにもかかわらず，着手されているものは非常に少ない．

そこで本研究では主に舞踊動作を対象とし，動作と音楽の双方から舞踊動作を観察・解析し，得られた知見を基に新たな舞踊動作を生成する手法について提案する．舞踊においては，演者が演奏されている楽曲からそのテンポ，リズムの早さや曲調，盛り上がり，ジャンルなどの情報を抽出し，それらを基に動作を構成する．そのため舞踊動作は人間の認識能力とそれに基づく動作を解析・生成するのに最も適した研究題材の一つである．

本論文では以下に示す三つの手法について提案する．

一つ目の研究では動きのリズムと音楽のリズムとの関係に関する解析手法を提案する．実際の舞踊を観察してみると，動きのリズムは「留め動作」，すなわち動きが静止している状態によって表されることが多く，演者は留め動作を音楽のリズムに合わせることで舞踊を披露している．本手法では，最初に実際の舞踊動作データから手，足，または重心がほぼ静止している時刻を求めて留め動作の候補点とし，また動作計測時に使われた楽曲データから「音がどのくらいの強さで発音されたか」を示す発音成分を抽出し，その周期性から音楽のリズムを推定する．そして双方の情報を考慮することで，舞踊動作のキーポーズを抽出する手法について述べる．また実験により動きの留め動作と音楽のリズムとの間に強い相関性があることだけでなく，本手法の結果が実際の舞踊演者の理解と近いことを示す．

二つ目の研究は楽曲の速さに応じて生じる動きの変化のモデル化手法を提案する．ある型の決まった舞踊動作を 1.0 倍の音楽再生スピードに合わせて演じた場合と 1.5 倍の再生スピードに合わせて演じた場合とを比較してみると，大局的に見れば同じ舞踊動作をしていても，局所的に見るとわずかではあるが動きの違いが見られる．これは楽曲の速さに追従するために動作の細部を省略し，本質の部分のみを残そうとした結果であると考えられる．そこでこれらの動作列を周波数領域で解析した結果，一つ目の研究で得られた留め動作が保存されること，動きが速くなるにつれて高周波成分から省略されていくこと，の二つの知見が得られた．この観察結

果を基に，実際にリズムの速さに基づく動きが変化する様子をモデル化し，実験によってその有効性を示す．またCGアニメーションやヒューマノイドロボットにおけるアプリケーション例も示す．

三つ目の研究では楽曲の曲調が舞踊動作に与える影響について観察を行い，楽曲の曲調に合った舞踊動作を自動生成する手法を提案する．人は音楽を聞いている間，その楽曲の曲調や激しさなどからさまざまな感情を得る．例えばロックなどの激しい音楽を聴いている場合は感情が高揚することが多く，またバラードなどのゆったりとした音楽を聴いている場合はリラックスした気分になる．実際に創作舞踊を例として観察してみると，楽曲の盛り上がっている部分では舞踊が激しくなり，また落ち着いた曲調の部分では落ち着いた舞踊が披露されている場面が多いことが分かった．そこで，一つ目の研究で得られた音楽リズムと留め動作の相関性に加え，音楽の盛り上がりと動きの盛り上がりの間にも相関があると仮定し，入力した楽曲の特徴と合った舞踊動作を生成する手法を提案する．本研究では，動作生成にはローカルな最適解探索方法とグローバルな最適解探索方法の二種類を用意し，目的に応じた使い分けを可能としている．実験を通して，あたかもCGキャラクタが楽曲に合わせて表現豊かな舞踊動作を演じているかのような結果が得られた．

以上これを要するに，本論文では，舞踊動作を研究対象とし，人間の認識・知覚能力を基にした舞踊動作の解析・生成に関する取り組みがなされており，舞踊動作の肝となる留め動作に関する解析手法，楽曲リズムの変化に伴う動きの変化のモデル化手法，楽曲の特徴に合った舞踊動作の自動生成法が提案されている．また本研究の成果はエンターテイメントシステムとして活かされるだけでなく，失われつつある無形文化財のアーカイブ化への応用なども期待され，社会上・実益上の観点から見ても寄与するところが大きい．

Acknowledgements

First of all, I would like to express my sincere gratitude and appreciation to my supervisor, Prof. Katsushi Ikeuchi, not only for his expert guidance and mentorship, but also for his encouragement and support at all levels. He gave me the freedom to follow my interests and the guidance to keep me on track whenever I confronted difficult problems in my research. Without him, I would never have been able to complete this dissertation.

I am no less grateful to Prof. Atsushi Nakazawa, the former director of the CVL Human Motion Group. He instructed me in CG animation research during my master's studies. Although he has gone to Osaka University, his careful advice regarding my research and presentations were of significant help to me, and I would like to thank him.

I would also like to thank all the members of our laboratory, especially in the CVL Human Motion Group for their kindness, generosity, and expertise. Dr. Shunsuke Kudoh always gave me lots of valuable advice and suggestions on my research during the last year of my Ph.D. studies. Prof. Taku Komura of the University of Edinburgh, Dr. Jun Takamatsu, Dr. Shin'ichiro Nakaoka of National Institute of Advanced Industrial Science and Technology, Miti Ruchanurucks, and Manoj Perera also deserve very special thanks for discussing my field of study with me, giving me valuable advice, and helping me relax with their enjoyable conversation.

I would like to thank Ms. Hisako Yamada, who is an excellent grand master of traditional Japanese folk dances. She generously allowed me to use her dance performances for experiments in my study. In addition, she helped in many presentations that introduced my research achievements. I am also very grateful to the members of her dance group, *Aizu-Gyokusui-Kai*.

I would also like to thank Mr. Takaaki Kaiga of *Warabi-za* for allowing me to use their motion databases.

I wish to express my deepest gratitude to my mentor at Microsoft Research Asia, Dr. Yasuyuki Matsushita, for his wisdom, creativity, and his strong advocacy of my research activities. Although I felt that I lacked knowledge in the field

of computer vision, he generously gave me great suggestions and ideas. Special thanks is also due to Dr. Sing Bing Kang of Microsoft Research and Dr. Xiaoou Tang of Microsoft Research Asia, both of whom gave me helpful comments and suggestions for my research project at Microsoft Research Asia.

I would also like to express my special appreciation to my roommates, Zhou Jian and Zheren Hu. Thanks to them, I survived a tough life in Beijing.

I would like to thank Dr. Joan Knapp, Mr. Robert Knapp and Ms. Marie Elm for proofreading my English. They kindly improved my writing and gave me appropriate suggestions.

I would also like to thank many people outside the laboratory. In particular, I am very grateful to all the members and affiliates of *Espoir Saxophone Orchestra*, to which I have belonged for 5 years, for their contributions to the enjoyment of my student life. Their saxophone performances have always encouraged me a lot.

Finally, I would like to thank my wonderful family for their constant support and encouragement.

December 2006

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Recent demand for realistic-looking human motion is rapidly increasing in the fields of computer graphics (CG) and robotics. One of the easy solutions to this problem is to use a motion capture system, which captures precise position data of markers attached to parts of the human body. Such data always has physical consistency without any false artifacts such as "foot-skating". However, it still remains difficult to capture the motion that animators really want, because of shortcomings in this data; for example, there are differences in body size between characters and actual performers. To solve this problem, most prior work aimed to either edit motion capture data [BW95, WP95, Gle98, LS01], to seamlessly blend [WH97, KG03] or connect [KGP02, AF02] them, or to modify them according to physical properties [TSK00, KG02] or kinematic constraints [PHRA02, RNKI06].

In most cases, however, human movements are induced by external signals: people first receive visual information such as environmental obstacles from eyes, or audio information such as speech or music from ears, recognize essential information or feel some emotion from the obtained information, and finally perform movements. Consideration of these front-end human actions makes it possible to automatically synthesize more human-like motion. Despite this possibility, only a few methods considering these aspects have been developed [PO03, SDO*04, SMK05].

To meet this need, we are focusing on dance performance as an experimen-

tal subject. Dance performance strongly depends on musical features such as rhythm, speed, mood, intensity, or genre of played music recognized by dance performers, and is well-suited to the kind of research. In particular, our subject is Japanese folk dance. Japanese folk dance is very semantic; e.g. a hand trajectory might symbolize the shape of a mountain, and most previous methods have been developed to analyze these semantics based on historical context information. Unfortunately, Japanese folk dance is disappearing. There are fewer trained dancers, and very few methods to analyze the key features of this form of dance. It is becoming increasingly important to archive dance performance alongside analysis of the key features of each dance performance. Our approaches can be very useful for this purpose.

The final goal of our studies is to realize *dancing-to-music* ability for CG characters and humanoid robots. For dance analysis and synthesis, we combine motion capture data and musical wave signals. Our approach consists of three main steps: motion analysis, music analysis, and motion synthesis based on the results of the previous two steps, as shown in Figure 1.1.

In this dissertation, we have proposed and developed three novel methods to analyze and synthesize dance performance.

Analysis of the Relationship between Motion and Musical Rhythm

We propose a method to analyze the relationship between motion and musical rhythm. According to observations of dance performance, motion rhythm is represented with a stop motion called a *keypose*, at which dancers clearly stop their movements; this allows dance movements to become synchronized with a performance's musical rhythm. In other words, motion rhythm has a strong dependence upon musical rhythm. The proposed method aims to reveal this relationship. It consists of a music analysis step that estimates musical rhythm and a motion analysis step that extracts keypose candidates. By integrating the extracted information, keyposes that are very similar to dancers' understandings are extracted.

Motion Synthesis Based on the Relationship between Motion Style and Musical Speed

We propose a method to model how to modify upper body motion based on the speed of played music. When we observed structured dance motion performed at a normal music playback speed versus motion performed at faster

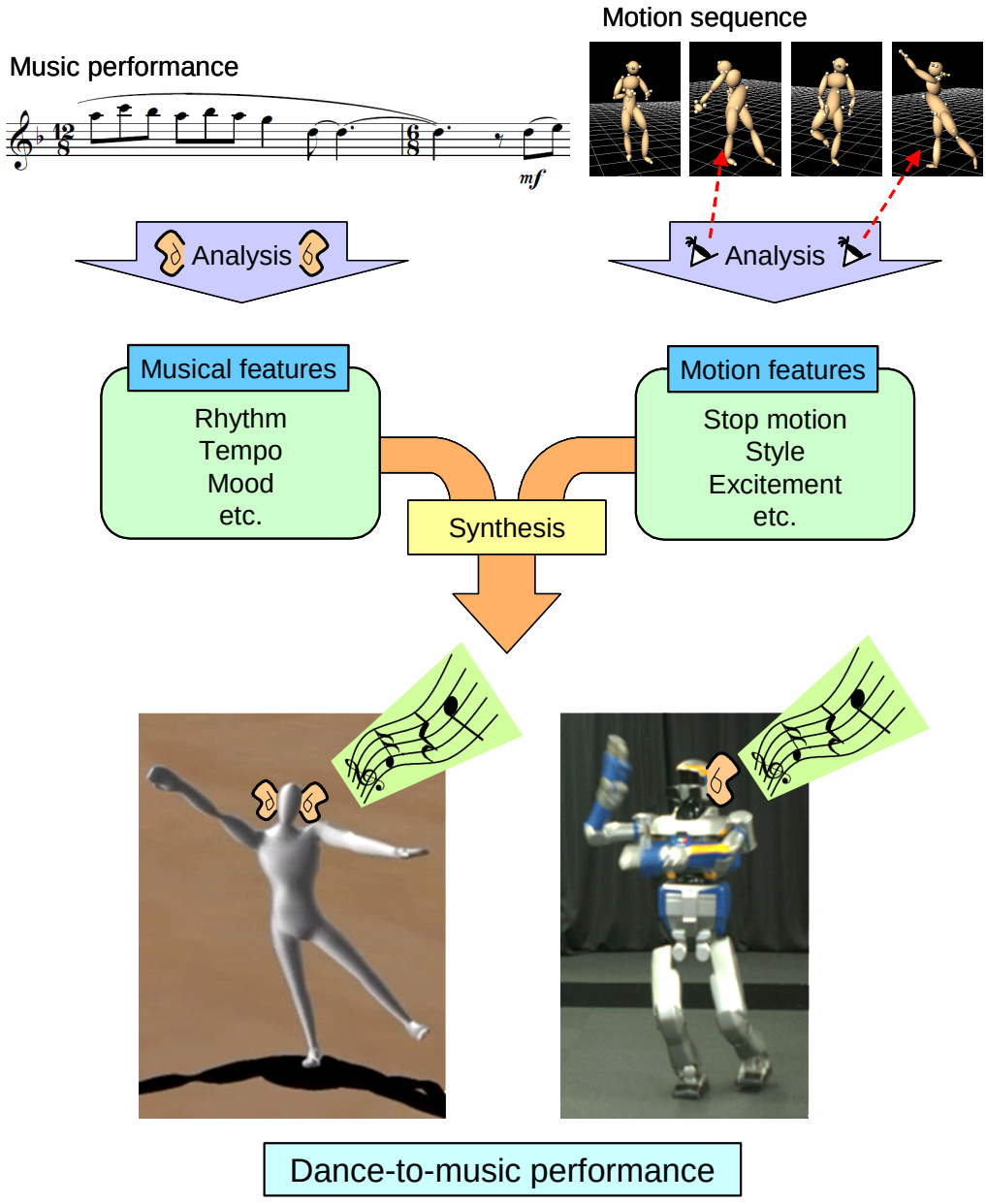


Figure 1.1: Our goal: to realize *dance-to-music* ability for CG characters and humanoid robots. Our approach consists of three main steps: motion analysis, music analysis, and motion synthesis based on the results of the previous two steps.

music playback speed, we found that the details of each set of motions, which is generally called *style* such as individual differences, vary slightly, while the whole of the dance motion are quite similar. This phenomenon arises from the fact that during faster music speed, dancers tend to omit details in order to perform the essential parts of the dance. We characterized motion differences at different speeds in the frequency domain, and thereby obtained insights into real dancers' omission of motion details. Based on these insights, we propose a novel method of modeling this motion modification, and develop some applications useful in CG character animation and humanoid robot motion generation. The experimental results look very natural, indicating the effectiveness of our method.

Motion Synthesis Based on the Relationship between Motion Excitement and Musical Intensity

We propose a method to automatically synthesize dance motion that is well matched to input music. It is based on the fact that people feel various emotions depending on the mood expressed by the music. For example, people feel quiet and relaxed when listening to relaxing music such as a ballad, and they feel agitated or excited when listening to intense music such as hard rock music. We observed dance performances, especially original dance, and found that the same is often true for dance performance. Considering both rhythm and intensity, we design an algorithm to synthesize dance-to-music human motion. The experimental results indicate that our method effectively creates dance performance as if a character were listening and expressively dancing to the music.

1.2 Thesis Overview

Chapter 2 introduces an analysis method to extract the keypose by extracting stop motion and musical rhythm. For musical rhythm estimation, an onset component, which shows how much spectral power has increased from the previous time frame, is calculated, and musical rhythm is then estimated from the onset component sequence. As for motion information, it is detected when end-effectors and center of mass are stopping their movements from their speed sequences. Finally, the keyposes are detected by combining motion and musical information.

In Chapter 3, we first explain a motion decomposition method using a hierarchical B-spline, which is key to accomplishing the modeling of the motion modifications. By using a hierarchical B-spline technique, we observe the differences between motion performed at a normal musical speed and one performed at a faster musical speed. Then we discuss how upper body motion is modified. Based on our obtained insights, we improve the hierarchical B-spline-based method to decompose motion, and propose a new framework to synthesize new upper body motion that satisfies kinematic constraints. We also show some applications based on this method.

In Chapter 4, we first describe our observations on the relationship between motion and musical intensity using original dance. We explain how to extract motion and music intensity features. Motion intensity features are based on the concept of *Effort* proposed by Laban, and musical intensity features are based on the *sound pressure level*. We then explain two types of motion synthesis method incorporating motion and musical features. One is a locally optimal method, and the other is a globally optimal method. Users can select one of the methods depending on their purposes. As for the globally optimal method, we also develop a user interface that enables animators to control the synthesis process by choosing desired motion segments well matched to music segments. For example, animators can set key motions in the motion database for desired music segments, such as setting a jumping motion to the final scene of the song, or a punch motion to a particular sudden sound in the music.

In Chapter 5, we conclude this dissertation by summarizing our research and contributions, and discussing possible future research directions.

Chapter 2

Keypose Extraction for Dance Structure Analysis

2.1 Introduction

Recent improvements in motion capture system have enabled us to deeply understand human motion. Understanding human motion and codifying this understanding into a symbolic representation has been well studied in robotics in order to manipulate a robot more effectively by using the symbolized motion. Some previous methods have actually achieved such symbolization via observations of human motion [OTI*00, TTO*00, JM02, NNK*05, ITTN04]. A symbolic representation of human motion also makes it possible to archive intangible cultural heritage such as Japanese folk dances [NNIY02]. However, most previous method cannot recognize important features of human motion.

This chapter describes a novel method to analyze the relationship between a dance performance's stop motions and its musical rhythms in order to understand the essential features of dance motion. Motion rhythm in dance is represented by stopping movements called *stop motion*; dancers often synchronize their stop motions with musical rhythm. Our method directly analyzes motion capture data for stop motions; our method also analyzes music wave signals for musical rhythm. By integrating these two kinds of information, we extract important and representative instances of dance motion that we call *keyposes*, and reveal that motion rhythm has a strong connection with musical rhythm.

For musical rhythm estimation, an *onset component* which shows how much spectral power increases from the previous time frame is calculated, and musical

rhythm is estimated from the onset component sequence. As for motion information, Flash *et al*. [FH85] found empirically that every motion is represented as a sequence of motion segments, and that these motion segments are connected at the points in time when end-effectors are stopping their motions. Accordingly, our method detects when end-effectors and center of mass are stopping their movements from their speed sequence data. Combining motion and musical information allows the motion's keyposes to be established.

2.2 Prior Work

In this section, we introduce some related work on the structural analysis method of human motion, and musical rhythm tracking method.

2.2.1 Method of Keypose Extraction and Motion Structure Analysis

Generally, keyposes or keyframes show representative instances of animation sequences including video, and human motion. In computer graphics, they are well known as important features to create computer animation. One of the traditional methods for human motion animation is to interpolate transition motion between the keyposes specified at desired times by the animator, and this function has been implemented in some commercial products such as Maya [Auta] and MotionBuilder [Autb]. There are some improved methods to synthesize 2D animation [CON05], or 3D animation including articulated figure animation [CCYL04, IMH05].

These animation synthesis methods are a bottom-up approach: animation is synthesized by specifying keyposes. On the other hand, keypose detection, or motion structure analysis is a top-down approach: keyposes are extracted from a previously existing constructed animation or real movement sequence. The usefulness of detected keyposes depends upon the application; whether they are being used for visualization vs. used by a performer. Assa *et al*. [ACC05] proposed a method to summarize motion capture data by selecting keyposes. Their requirement for selecting a keypose is that it must contain salient and informative human motion for the sequence of interest; poses containing high intensity are selected via their motion analysis method. This method was extended for the summarization of video that contains human activities by Caspi *et al*. [CAMG06]. Sakamoto *et al*. [SKK04] proposed a motion retrieval method with keyposes extracted through a self-organizing map (SOM). While these methods

effectively extract visually salient postures and valuable motion data, they are not sufficient for motion structure analysis: keyposes extracted by these methods can be useful for visualization, but not always useful for performers desiring to reproduce the same motion.

Keypose extraction specifically for motion structure analysis has been well studied both in computer graphics and in robotics research. Li *et al*. [LWS02] proposed a method to synthesize human animation using *motion textons.* Motion textons are repetitive patterns in human motion, vs. the standard texton used in image texture synthesis [MBSL99]. This work extracted motion textons through linear dynamic systems (LDS) and synthesized human motion by modeling the distribution of motion textons. Barbič *et al*. [BSP*04] proposed a method to segment human motion data into distinct behaviors using defined motion and principal component analysis (PCA). These methods requires known motion segments to segment and classify a new motion sequence; it is not applicable to unknown motion sequences.

In robotics research, the goal of motion structure analysis method is to imitate human motions with humanoid robots by symbolizing motion. Codifying human motion into symbols makes it much easier to control a humanoid robots, especially for balance maintenance. According to Flash *et al*. [FH85], every human motion consists of several *motion primitives*, which denote fundamental elements of human motion, and these primitives are segmented by detecting instances when hands and feet stop their movements. Ogawara *et al*. [OTI*00] and Takamatsu *et al*. [TTO*00] proposed a method to extract motion primitives from upper body motion and to imitate human motion using the extracted motion primitives. Nakaoka *et al*. [NNK*05] proposed a method to analyze the structure of lower body motion in order to imitate human dance motion with a humanoid robot. In whole body motion, there are many methods to segment human motion by detecting the local minima of end-effector speed, and classify the motion segment into several clusters by calculating co-occurrence [OSU00, NNIY02], by using Hidden Markov Models (HMM) [ITTN04], or by applying a spatio-temporal isomap for dimensionality reduction [JM02]. Kahol *et al*. [KTP03, KTP06] proposed a motion segmentation method using approximated physical parameters such as force, momentum and kinetic energy. Unfortunately, all these methods share a common problem in that too many keyposes/motion segments are extracted because of the high degree of freedom of an articulated figure.

Regarding to dance motion specification, *Labanotation* [Hut77] proposed by Rudolf Laban is one of the most popular dance notation systems. Figure 2.1 shows an example of Labanotation. Hachimura *et al*. [HN01] and Kojima *et*

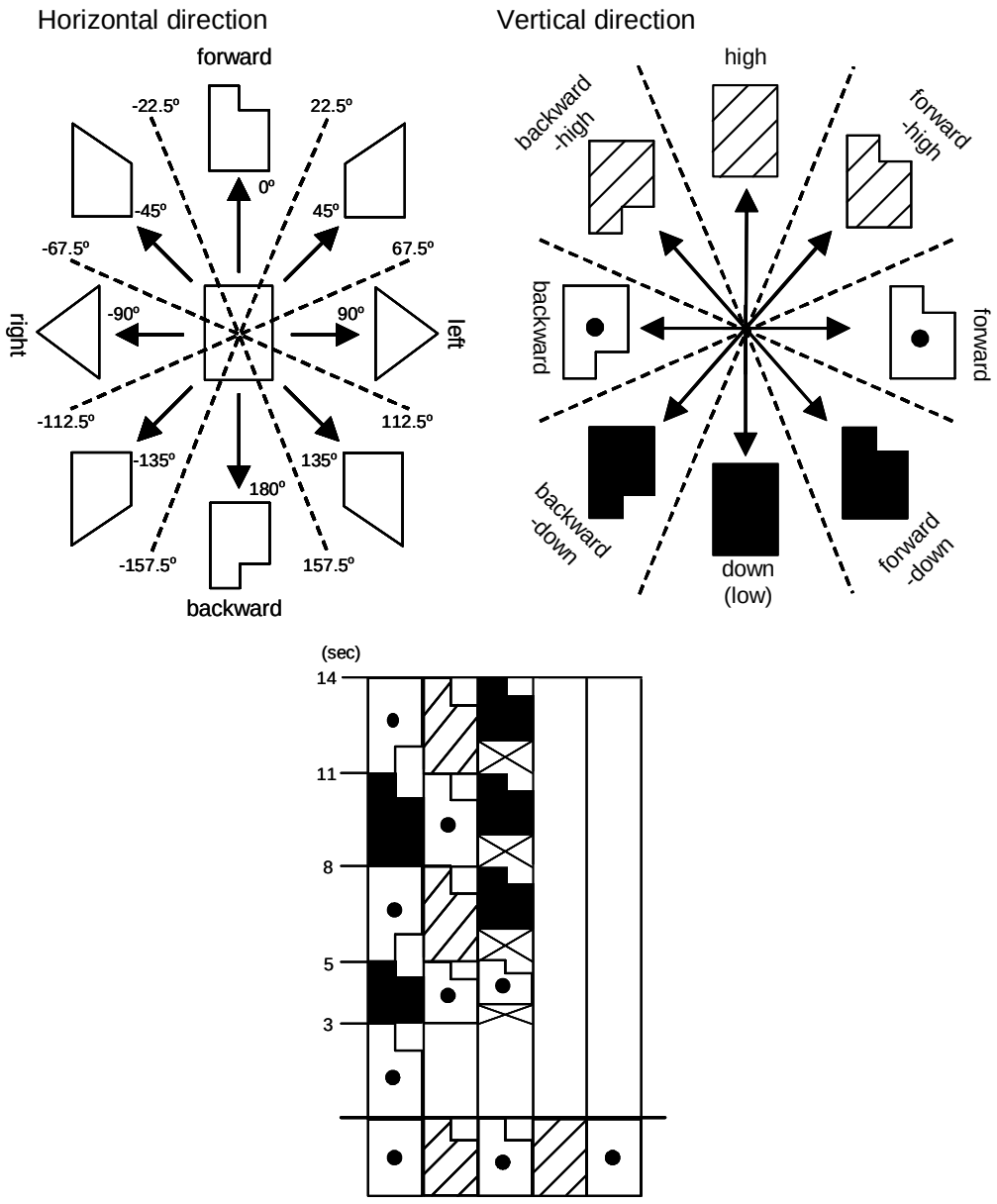


Figure 2.1: Labanotation: a notation method for dance motions. The top two figures represent the notation symbols used for Labanotation; the bottom figure represents an example of a notated dance sequence.

al. [KHN02] proposed a method to convert input motion sequences into a Labanotation score. Unfortunately, the information provided by the Labanotation system is limited in that details of dancer postures are unspecified. Kahol *et al*. [KTP04] extended their method for dance structure analysis. Based on this method, Dyaberi *et al*. [DSJQ04] also proposed a method to analyze dance motion structure using a topological graph structure. However, this method can only be applied to certain forms of dance such as the Rondo: it is not a versatile method.

2.2.2 Musical Rhythm Tracking Method

Most humans have an ability to recognize rhythm and rhythm structure. When people hear music, they will tap their foot, wave their hands in time with the music, and can dance to the music even if they are children or beginners. Thus, rhythm tracking is a fundamental research topic in the music signal processing field, and has attracted many researchers.

In the case of MIDI signals, parameters of various musical features such as onset, pitch and volume are easily obtained and useful in rhythm tracking [DH89, Ros92a, Ros92b, DH94, LK94]. However it is very difficult to extract most of these musical features from audio signals; this problem has been studied by many researchers.

Most rhythm tracking methods for audio signals are based upon knowledge of the onset component [Tod94, LZ03]). In particular, Goto [Got01] proposed a real-time rhythm tracking method based on not only the onset component, but also chord changes and drum sounds for rhythm structure analysis. Scheirer [Sch98] proposed an offline rhythm tracking method for music which has *accel.* and *rit.* and whose rhythm is not constant. There are methods which can predict the musical rhythm by using Kalman filtering [CKDH01], applying image processing techniques to musical spectral components [NT04], and using a Bayesian network [SMS05].

2.3 Approach

An overview of our keypose extraction method is illustrated in Figure 2.2. The inputs to our method are music signal in wave format, and motion capture data that contains dance motion matched to the input music. We estimate musical rhythm components from the onset components in the input music signal, while we extract motion keypose candidates from the speed sequence of

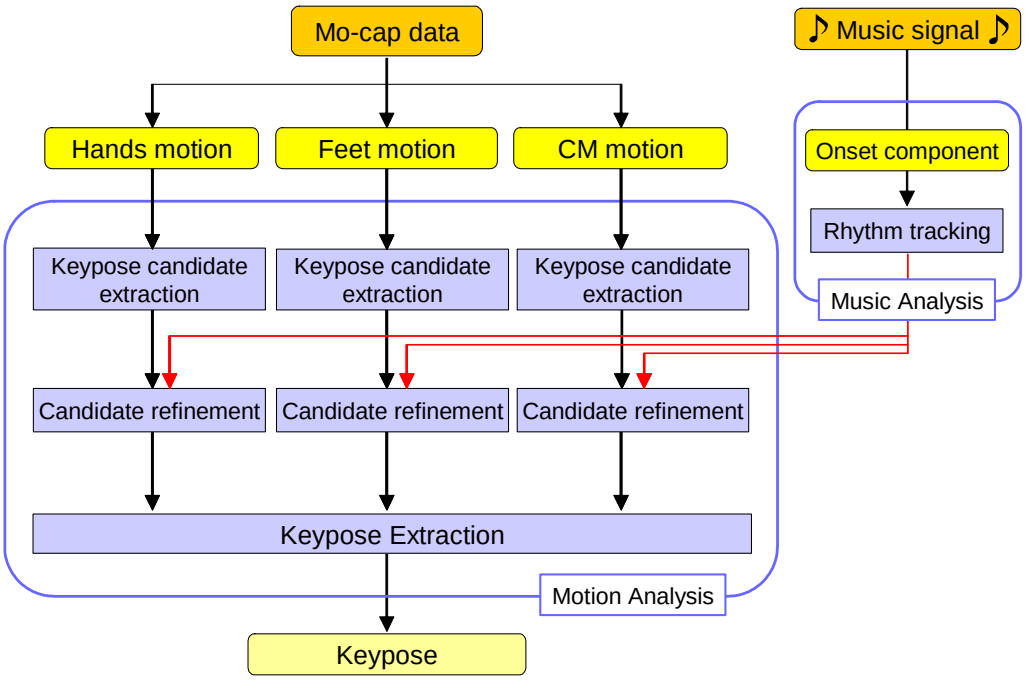


Figure 2.2: Overview of our keypose extraction method. Musical rhythm is estimated from the onset components, and from motion capture data, we extract motion keypose candidates from the speed sequences of end-effectors and the center of mass. Motion keypose candidates are refined by considering the musical rhythm. Finally, dance keyposes are extracted from the refined motion keypose candidates.

hands, feet, and center of mass motion. As there are initially too many motion keypose candidates to be useful, and motion keypose candidates are refined by considering musical rhythm; we extract useful keyposes from a refined subset of motion keypose candidates.

2.4 Rhythm Tracking from Music Sequence

To estimate musical rhythm, we use the following known principles:

Principle 1: A sound is likely to be produced consistent with the timing of the rhythm.

Principle 2: The interval of the onset component is likely to be equal to that of the rhythm.

So we consider the onset component for estimating the musical rhythm. Figure 2.3 illustrates onset component extraction. Here, we denote the spectral power of the k-th note at the t-th temporal frame as $X(t,k)$. Using Principle 1, we calculate the onset component of the k-th note, which is the power increase from the previous temporal frame $t-1$ defined as $d(t,k)$ [Got01].

$$d(t,k) = \begin{cases} \max(X(t,k), X(t+1,k)) - \text{PrevPow} \\ \quad \text{if } \min(X(t,k), X(t+1,k)) \geq \text{PrevPow}, \\ 0 \quad \text{otherwise} \end{cases} \tag{2.1}$$

where

$$\text{PrevPow} = \max(X(t-1,k), X(t-1,k\pm 1)). \tag{2.2}$$

By calculating the sum of the onset components $D(t) = \sum_k d(t,k)$, we can determine the total intensity of the sounds produced at the t-th temporal frame.

Using Principle 2, we calculate the auto-correlation function of $D(t)$ to estimate the average rhythm interval t_{rhythm}:

$$t_{\text{rhythm}} = \arg \max_{\tau \in [T_{\min}, T_{\max}]} \frac{1}{T} \sum_{t=1}^{T} D(t) \cdot D(t+\tau), \tag{2.3}$$

where T is the number of temporal frames and $T_{\min}$ and $T_{\max}$ are defined as the interval of 150 bpm and 60 bpm respectively. Then, the starting time t_{start} is estimated by calculating the cross-correlation function between $D(t)$ and the pulse sequence $P(t)$ whose interval is the estimated rhythm interval t_{rhythm}:

$$t_{\text{start}} = \arg \max_{\tau} \frac{1}{T} \sum_{t=1}^{T} D(t) \cdot P(t+\tau). \tag{2.4}$$

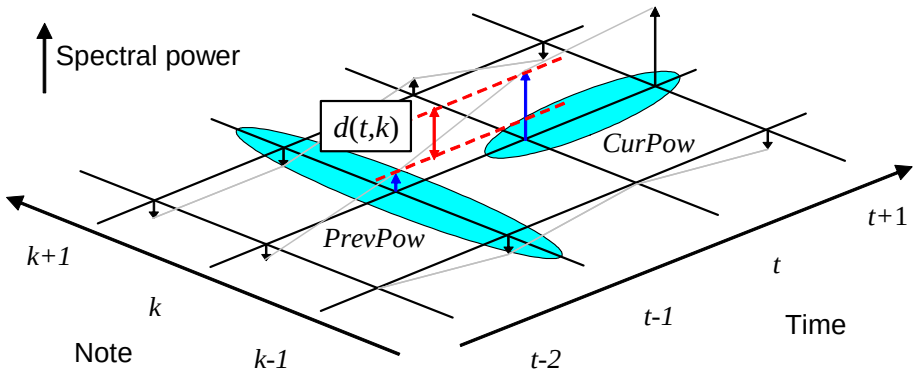


Figure 2.3: Illustration of onset component extraction. First, the maximum among $X(t-1,k)$ and $X(t-1,k\pm1)$ described as *PrevPow*, and the minimum between $X(t,k)$ and $X(t,k+1)$ described as *CurPow* are extracted. Then, the difference between *CurPow* and *PrevPow* is calculated. If the difference is larger than 0, $d(t,k)$ is the difference; otherwise, $d(t,k)$ is 0.

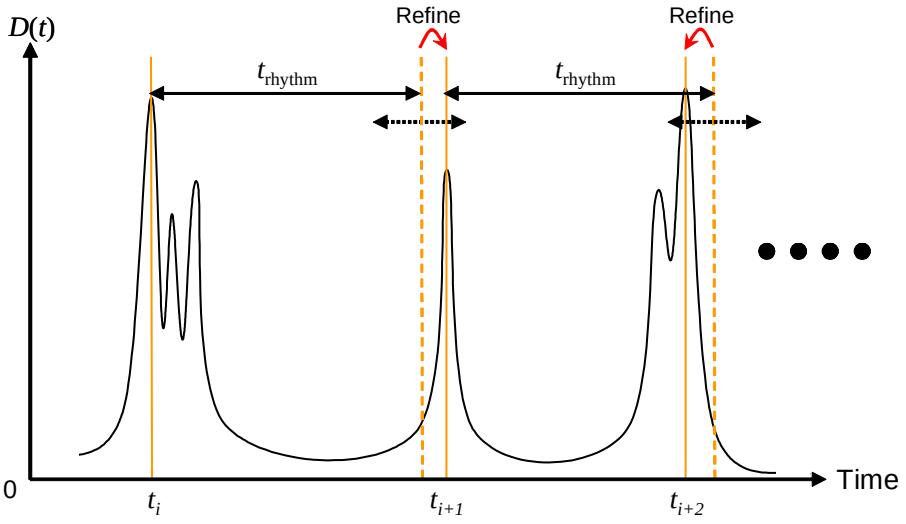


Figure 2.4: Refinement of the estimated musical rhythm. Our method finds the local maximum around the estimated rhythm.

However, in practice, a rhythm interval may change slightly due to the performers' sensibilities, changes in the music itself, etc., and errors caused by these changes make rhythm tracking using rigid timing impossible. To solve this problem, considering Principle 1 again, our method tracks the local maximum around the estimated rhythm. This technique is illustrated in Figure 2.4.

2.5 Keypose Candidate Extraction from Motion Sequence

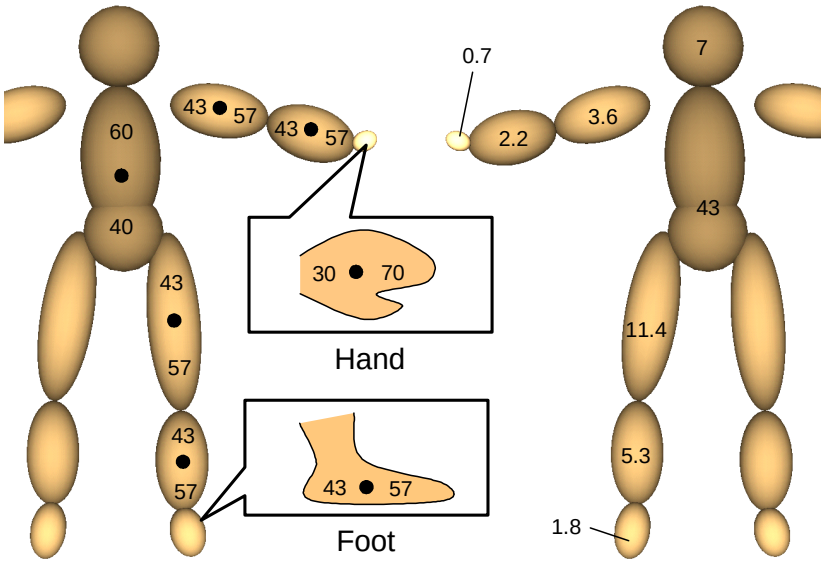
Our motion analysis method is based on the speed of a performer's hands, feet and center of mass (CM). In many forms of dance, including Japanese traditional dance, the movements of hands and feet have a strong relationship with the intended expression of the whole body. Therefore, the speed of the hands and feet are useful for extracting stop motions. However, this is not sufficient for keypose extraction because sometimes the dancer makes rhythm errors, or dances are varied by the preferences or the genders of performers, etc. So in addition to the motion of the hands and feet, our algorithm uses the motion of the body's CM. The motion of the CM represents the motion of the whole body; thus, the effects of missteps and individual differences are less. CM motion is calculated from the standard mass distribution of a human body shown in Figure 2.5. Through this step, we extract motion keypose candidates which satisfy the following criteria:

1. Dancers clearly stop their movements.
2. Dancers clearly move their body parts during neighboring candidates.

2.5.1 Body Center Coordinate System

Captured motion data is recorded in a global coordinate system. But to calculate the speed of hands, we define a local body-centered coordinate system named the *body center coordinate system* as follows. The origin O of this local coordinate system is the middle of the human waist. The three axes of the local coordinate system are shown in Figure 2.6. Note that the x-axis represents a forward direction for the waist, the y-axis points left relative to the waist, and the z-axis is perpendicular to these two axes and points upward. This coordinate system makes it simpler to understand the motion of hands relative to the body.

Hand motions are converted into the body center coordinate system, and then the speed of the hands is calculated. On the other hand, the speed of the



(a) (b)

Figure 2.5: Standard mass distribution of a human body. (a): The position of the CM of each body part, and (b): Relative mass of each body part [%].

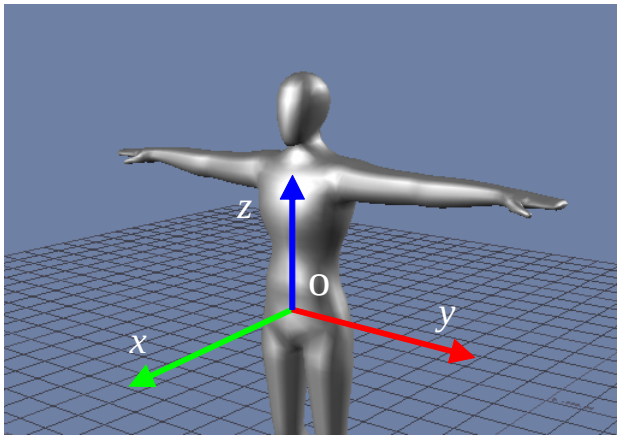


Figure 2.6: Body center coordinate system. The origin O of this local coordinate system is the middle of the human waist. The x-axis represents the forward direction of the waist, the y-axis points left relative to the waist, and the z-axis is perpendicular to these two axes and points upward.

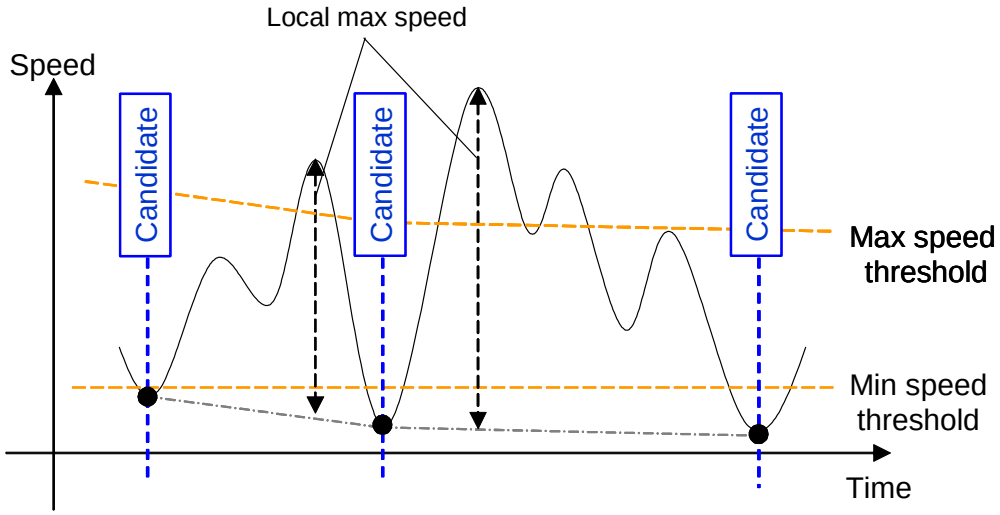


Figure 2.7: Keypose candidate extraction for hand and CM motions. The solid line represents a typical speed sequence of hands and CM, and horizontal dashed lines represent speed threshold values.

feet and of the CM are calculated in the global coordinate system. In the global coordinate system, the speeds of the feet and of the CM is nearly zero when these parts stop, so it is easy to extract the stop motions of these parts.

The effects of noise are reduced by smoothing the motion sequence with a Gaussian filter before extracting candidates.

2.5.2 Keypose Candidate Extraction

After calculating speed, we extract keypose candidates that satisfy the following criteria:

1. Dancers clearly stop the movements of their end-effectors.
2. Dancers clearly move their body parts during neighboring candidates.

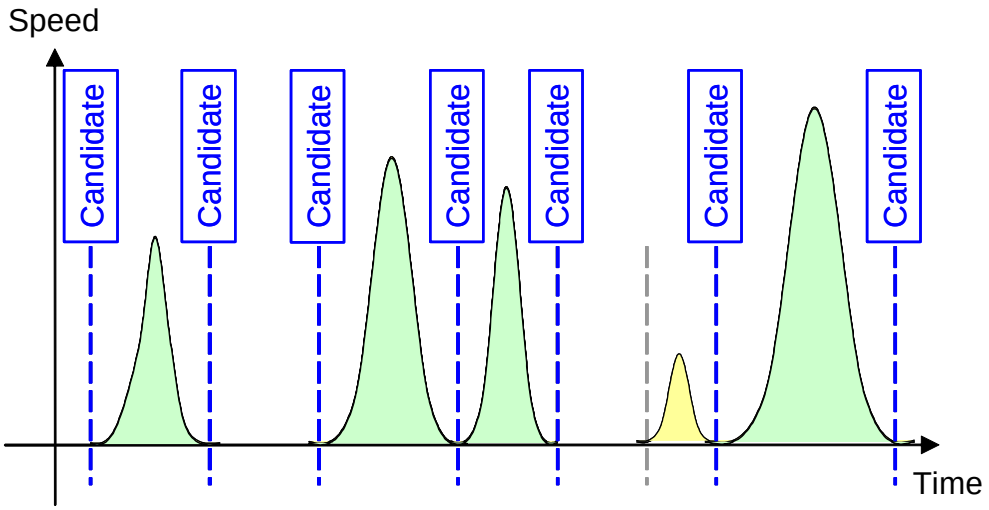


Figure 2.8: Keypose candidate extraction for foot motions. The solid line represents a typical speed sequence of feet, and the green- and yellow-colored areas represent the feet trajectory lengths that satisfy, and do not satisfy, respectively, the thresholding process.

Keypose Candidate Extraction for Hand and CM Motions

In hand and CM motions, the speed sequences demonstrate stop instances, as shown in Figure 2.7. To extract keypose candidates for hands and CM motions, we define the following two criteria which satisfy the keypose candidates criteria described above:

1. Each candidate is a local minimum in the speed sequence, and the local minimum is less than a minimum speed threshold.

2. The local maximum between two successive candidates is larger than a maximum speed threshold.

Keypose Candidate Extraction for Foot Motions

In feet motions, one leg often functions as a supporting leg while the other leg is functioning as a swing sole. Thus, the speed sequence for feet motions

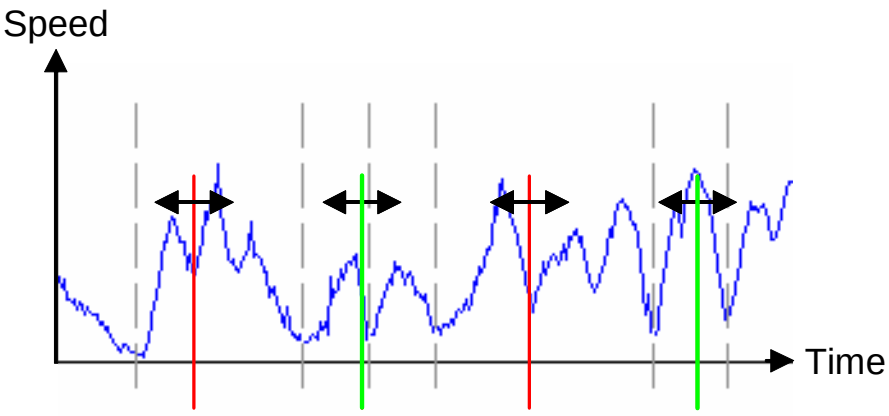


Figure 2.9: Refinement of the keypose candidates using the musical rhythm. Vertical solid lines represent the estimated musical rhythm; vertical broken lines represent the motion keypose candidates. Musical rhythm frames are considered to refine motion keypose candidates. Note keypose candidates relatively closely bracketing the second and fourth musical rhythm, but not the first and third musical rhythm; the first and third keypose candidates are therefore ignored.

often consists of a series of bell-shaped curves, as shown in Figure 2.8 To extract keypose candidates from feet motions, we first extract the rise and decay of the feet speed sequences. Then, the area between the rise and decay, which shows how far each foot moved while it was used as a swing sole, is calculated. If the area is larger than a trajectory length threshold, these rise and decay become candidates.

2.6 Keypose Extraction Using Motion Keypose Candidates and Musical Rhythm

2.6.1 Keypose Candidate Refinement Using Musical Rhythm

The next step is to refine the keypose candidates by considering musical rhythm. For each speed sequence, our method tests whether there are candidates around musical rhythm inflection points t_{rhythm} as detected from the onset

components. If there is a keypose candidate, it is possible that there is a stop point around t_{rhythm}, and if so, this keypose candidate is retained for the final step.

Figure 2.9 illustrates the keypose candidate refinement process. In this figure, a speed sequence is overlaid by vertical broken lines representing the initially extracted motion keypose candidates and solid green or red vertical lines representing the estimated musical rhythm. In this example, there are no keypose candidates immediately surrounding the first and third musical rhythms, which are represented by red vertical lines, so these keypose candidates are not retained in the next phase of the keypose extraction process. On the other hand, there are keypose candidates around second and fourth musical rhythms, represented in the figure by green vertical lines, and these candidates are preserved for the keypose extraction process.

2.6.2 Keypose Extraction

In the next phase, keypose candidates of a dance performance are subjected to two further criteria:

1. Retained keypose candidates must include a match in time between more than two of the following: left hand, right hand, or feet.

2. Retained keypose candidates must include a CM keypose match.

For example, the first criterion would be satisfied by keypose candidates of the left hand, the right hand, and one foot which match in time. It would not be satisfied by keypose candidate time-matches in only one foot and one hand. In other words, the first criterion can extract poses at which dancers stop the movements of their hands and feet even when the stopping instance of each body part is slightly different. These poses are likely to be stop motions.

But this first criterion may extract poses that are not considered to be keyposes. For example, consider *walking* motion. In this motion, a performer's hands nearly stop their movements when his/her feet are on the ground. However, the body keeps moving in the forward direction, and this pose cannot usefully be considered a stop motion. Such translations are common in dance, so we define the second criterion to help eliminate false positives (keypose candidates labeled as valid poses, when in fact they are not); both criteria must be simultaneously satisfied to retain a keypose candidate.

2.7 Experiments

2.7.1 Experimental Data

Our proposed method was evaluated using three dance sequences: the *Aizu-bandaisan* dance performed by a male and a female dance master, and the *Jongara-bushi* dance. The motion data for these three dance performances were captured by an instrument made by Vicon Motion Systems, an optical motion capturing system that recorded the position of 33 markers on each dancer (see Appendix B.1). The sampling rate of the *Aizu-bandaisan* dance and the *Jongara-bushi* dance were 120 fps and 200 fps, respectively. Figure 2.10 and Figure 2.11 show the actual dance performance and the captured motion data of the Aizu-bandaisan dance and the Jongara-bushi dance, respectively.

The music was converted into WAV format with a typical USB-hosted audio input device; 16-bit data was sampled at 32000Hz.

2.7.2 Results of Rhythm Tracking

To extract the music's onset components, its frequency spectrum was calculated using Constant Q Transform (CQT) (see Appendix A) . The windowing function for the CQT was a Hamming function whose size was 1024 samples and which was shifted by 256 samples at each step.

The estimated average rhythm interval of the Aizu-bandaisan and the Jongara-bushi dance music recordings were 0.704 seconds (around 85 beats per minute) and 0.576 seconds (around 104 beats per minute) respectively. Figure 2.12 show the results of our rhythm tracking applied to the Aizu-bandaisan and the Jongara-bushi dances, respectively. The upper window of this application shows the spectrogram for each dance, in which solid red vertical lines indicate the estimated rhythm. The lower window shows the onset component $D(t)$ of each octave-based frequency band. The rhythm appeared at the onset times, which are represented by deep gray in the spectrogram.

2.7.3 Results of Keypose Extraction

To evaluate the effectiveness of our proposed method, for all three dances, we compared the results of our keypose extraction method with the results from Nakazawa *et al.*'s method [NNIY02], which uses only motion capture data to extract keyposes. Additionally, we compared the results of our method with the

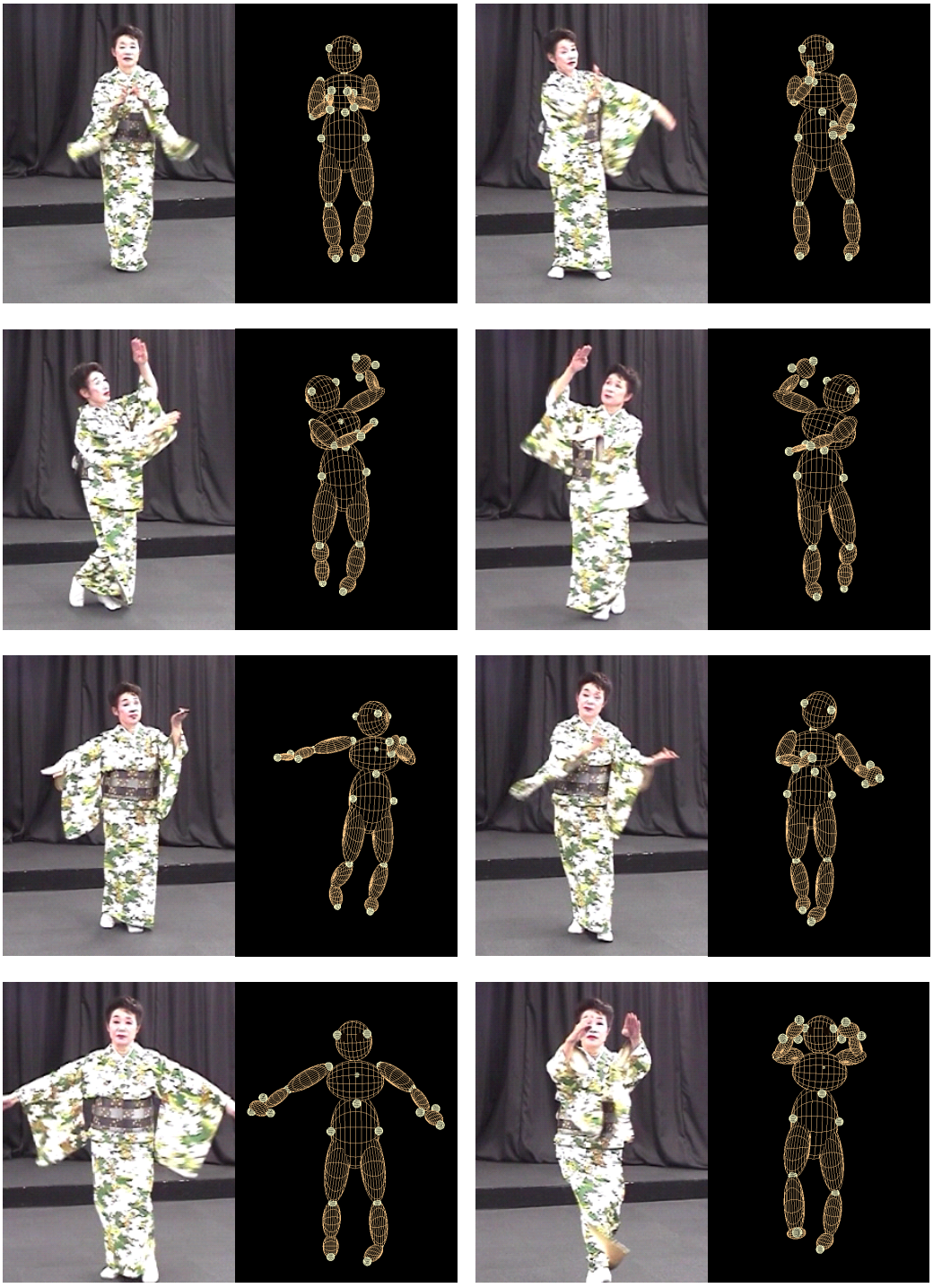


Figure 2.10: The *Aizu-bandaisan* dance.

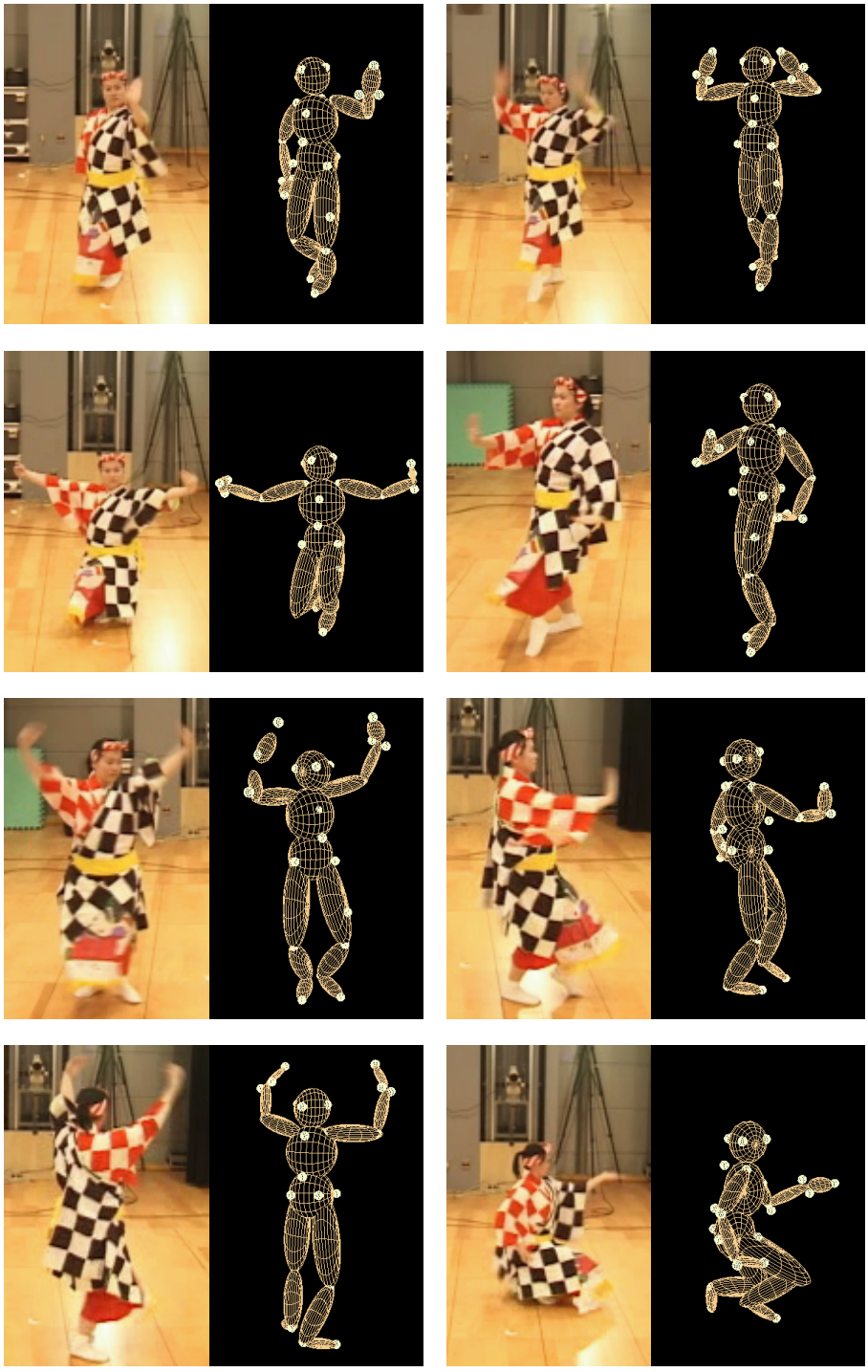
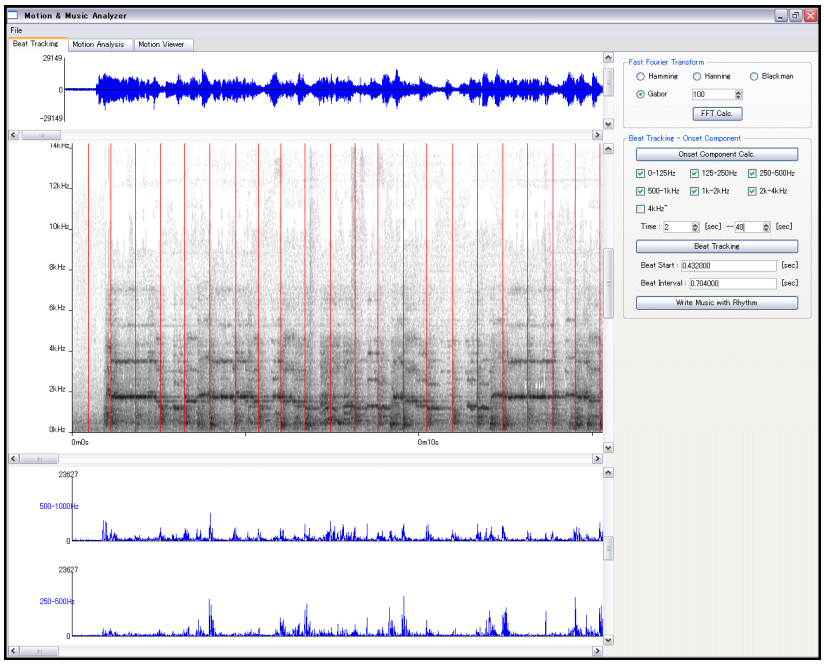
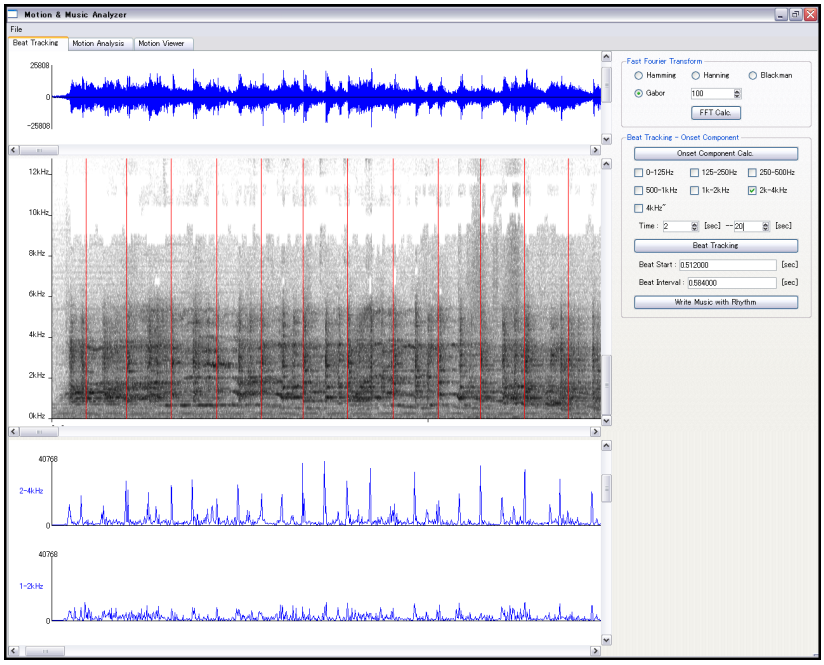


Figure 2.11: The *Jongara-bushi* dance.



(a)



(b)

Figure 2.12: Results of music rhythm tracking of (a) the Aizu-bandaisan dance, and (b) the Jongara-bushi dance.

keyposes manually extracted by dancers. Figure 2.13 shows the keyposes which dance masters recognize as important stop motions in these dances.

Aizu-bandaisan Dance Performed by a Female Dance Master

Our method's analysis of a female dance master performing the Aizu-bandaisan dance is shown in Figure 2.14. Five time-correlated speed graphs are shown for left hand, right hand, left foot, right foot, and CM, from top to bottom, respectively; solid green vertical bars indicate valid keyposes which satisfy all criteria. A 3D computer-generated figure shows each keypose extracted by our method, and below, the desired true keyposes of the dance are shown as drawn by a dance master. (The speed graphs in the figure are not complete transcriptions of the dance from beginning to end, but represent a subset of the data.)

Note that near the right side of this illustration, to the left of the rightmost red vertical line which represents a beat, a CM stop motion was detected and is shown as a light blue vertical bar in the lowermost speed signal graph. This keypose candidate was correctly determined by our method not to be a valid keypose, because although the left hand (shown as the uppermost signal) and right hand (shown immediately below the left hand) are likewise in stop motion, neither of the legs have been identified as keypose candidates. Because our criteria state that at least three of the four end effectors must be keypose candidates as well as the CM, this keypose candidate was correctly rejected as a valid pose. Several correct rejections are apparent in this figure.

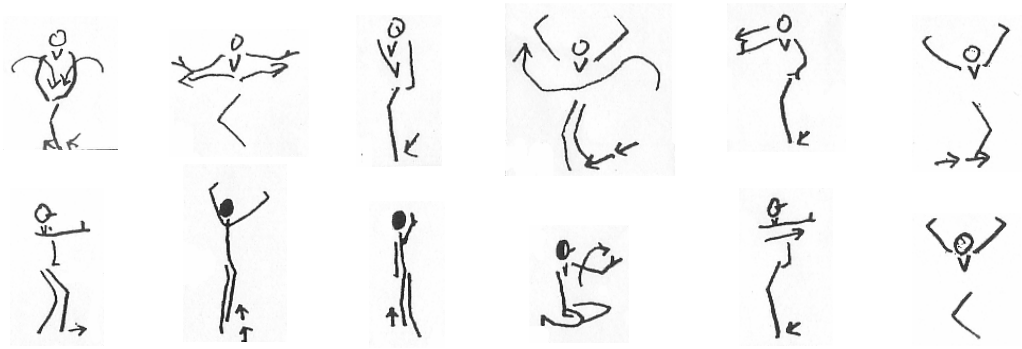
As summarized in Table 2.1, this dance has 9 true keyposes. Our method correctly extracted all of these keyposes with no false positives and no mis-detected errors. A previous method which considers only motion capture data extracted 8 of the 9 true keyposes correctly, but generated 4 false-positives and mis-detected 5 errors.

Aizu-bandaisan Dance Performed by a Male Dance Master

In this example of the dance, the male dancer moved his body parts with unusually large motions, and he did not stop clearly the movements at the dance's prescribed keyposes.

A graphical subset of the data using our method is shown in Figure 2.15. Table 2.2 summarizes our method's analysis in contrast with the previous method.

(a)



(b)

Figure 2.13: Keyposes extracted by dance masters. (a) The Aizu-bandaisan dance, and (b) the Jongara-bushi dance.

Note that the previous method can extract only 3 of the dance's 9 keyposes correctly, and there were 2 undetected errors. In contrast, our proposed method extracted 5 of the dance's 9 keyposes successfully, with 2 mis-detected errors.

These results imply that detection errors caused by individual differences between dancers can be considerably reduced by considering CM motions.

Jongara-bushi Dance

Results for our extraction method for a dancer performing the Jongara-bushi are shown in Figures 2.16 and 2.17, and are summarized in Table 2.3.

This dance has 12 true keyposes. The previous method extracted 6 of these, and had no mis-detected errors. In contrast, our method extracted 9 correct keyposes, with no mis-detected errors. We believe that our method failed to detect 3 keyposes because of the high speed of this dance.

2.8 Discussion

As shown in Section 2.7, the results of our method are much better than those of the previous method. This is derived from the fact that our method considers not only motion capture data but musical information, while the previous method considers only motion capture data. By incorporating an analysis of a dance ' s musical rhythm, we reduce the number of false positives that previous methods have generated due to the high degree of freedom of any articulated figure.

Additionally, through the comparison between our results and the keyposes specified by dance masters, we find that our results are much closer to the dancers' intended keyposes. In this way, our algorithm derives results quite similar to dancers' intended stop motions.

Although we have used traditional Japanese dance motion data in our experiments, we believe that since most dance performances have in common that stop motions are important, and that performers tend to dance to the rhythm of the music, our proposed method should work well for other types of dance such as ballet.

Noting that our method correctly identified all the keyposes of the Aizu-bandaisan dance but failed to identify all the keyposes of the much faster Jongara-bushi, it is possible that our discovery parameters would need adjustment when analyzing fast dance. For example, by using a different windowing technique, higher visual or musical sampling frequencies, or by generally tweaking all

our digital signal processing parameters, we might increase the accuracy of our proposed method in all forms of dance.

2.9 Summary

In this chapter, we have proposed a new analysis method for the motions of human dance. Our method is quite different from previous work in that our method considers not only motion information acquired from motion capture data, but also musical rhythm as estimated from recordings of the accompanying music. Our results prove that musical rhythm is one of the most important factors for dance performances, and that by considering musical rhythm, our method can efficiently acquire keyposes, which are very important in characterizing a particular dance. Our proposed method also considers the speed of the center of mass of a dancer to understand the movement of the body; it can therefore better handle individual variations in dance performances resulting from missteps, gender difference, style, and so forth.

We tested our method and a previous method on three dance sequences and found our method to be quantifiably superior.

	# extracted keyposes	accuracy rate (method / truth)	# mis-detected errors
Prev. method	13	89% (8/9)	5
Our method	9	100% (9/9)	0

Table 2.1: Evaluation of keypose extraction from the Aizu-bandaisan dance performed by a female dance master.

	# extracted keyposes	accuracy rate (method / truth)	# mis-detected errors
Prev. method	3	33% (3/9)	2
Our method	5	56% (5/9)	2

Table 2.2: Evaluation of keypose extraction from the Aizu-bandaisan dance performed by a male dance master.

	# extracted keyposes	accuracy rate (method / truth)	# mis-detected errors
Prev. method	6	50% (6/12)	0
Our method	9	75% (9/12)	0

Table 2.3: Evaluation of keypose extraction from the Jongara-bushi dance.

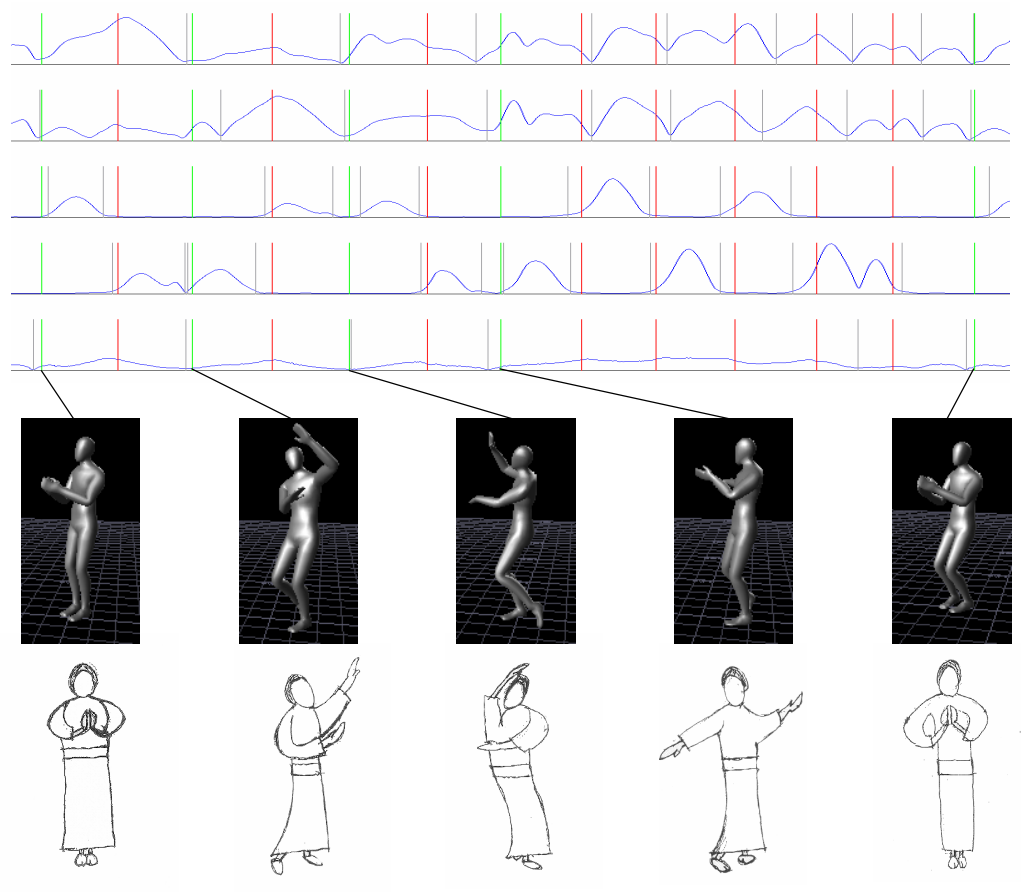


Figure 2.14: Result of keypose extraction from the Aizu-bandaisan dance performed by a female dance master. Five time-correlated speed graphs are shown for left hand, right hand, left foot, right foot, and CM, from top to bottom, respectively; solid green vertical bars indicate valid keyposes which satisfy all criteria. A 3D computer-generated figure shows each keypose derived from the motion capture data, and below, the desired true keyposes of the dance are shown as drawn by a dance master.

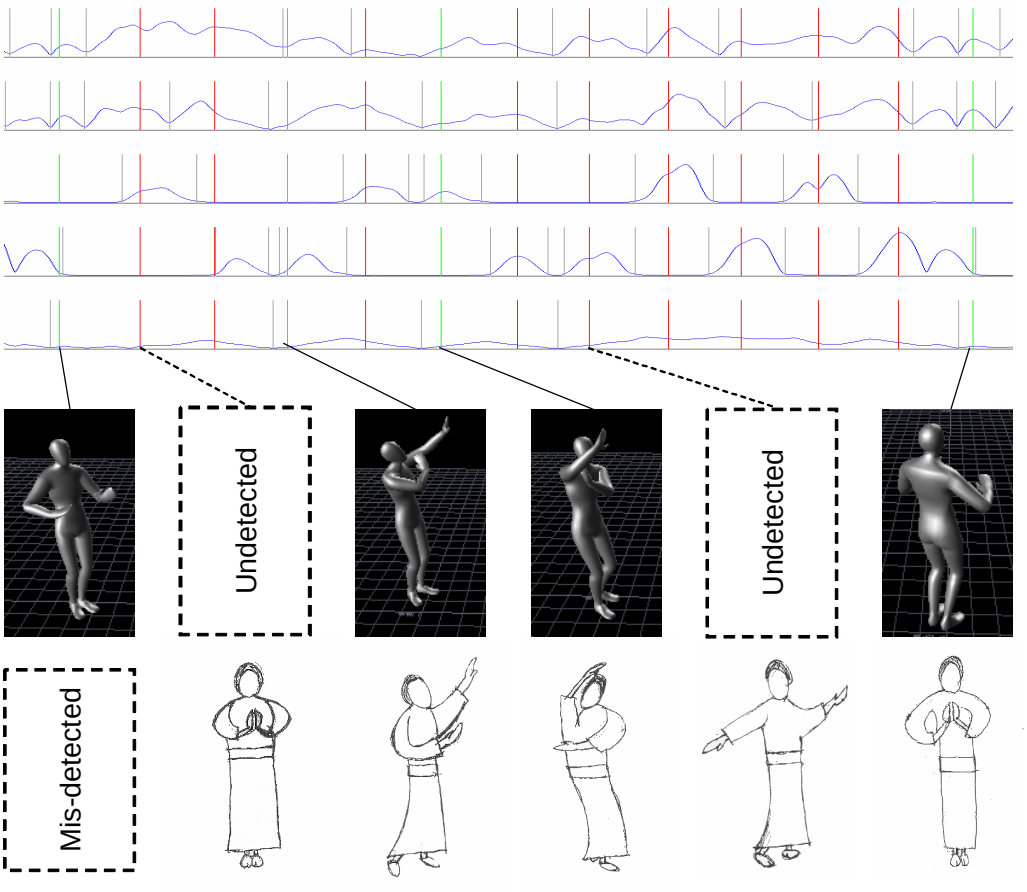


Figure 2.15: Result of keypose extraction from the Aizu-bandaisan dance performed by a male dance master. Five time-correlated speed graphs are shown for left hand, right hand, left foot, right foot, and CM, from top to bottom, respectively; solid green vertical bars indicate valid keyposes which satisfy all criteria. A 3D computer-generated figure shows each keypose derived from the motion capture data, and below, the desired true keyposes of the dance are shown as drawn by a dance master.

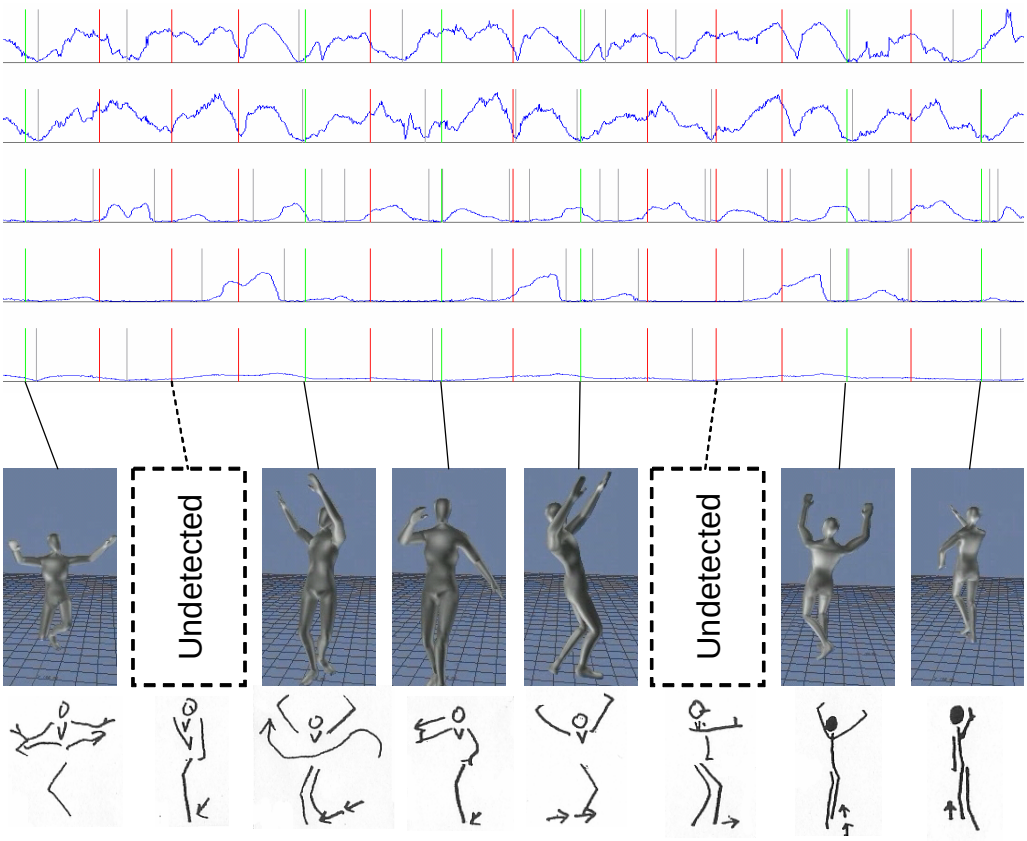


Figure 2.16: Subset of extracted keyposes from the Jongara-bushi dance. Five time-correlated speed graphs are shown for left hand, right hand, left foot, right foot, and CM, from top to bottom, respectively; solid green vertical bars indicate valid keyposes which satisfy all criteria. A 3D computer-generated figure shows each keypose derived from the motion capture data, and below, the desired true keyposes of the dance are shown as drawn by a dance master.

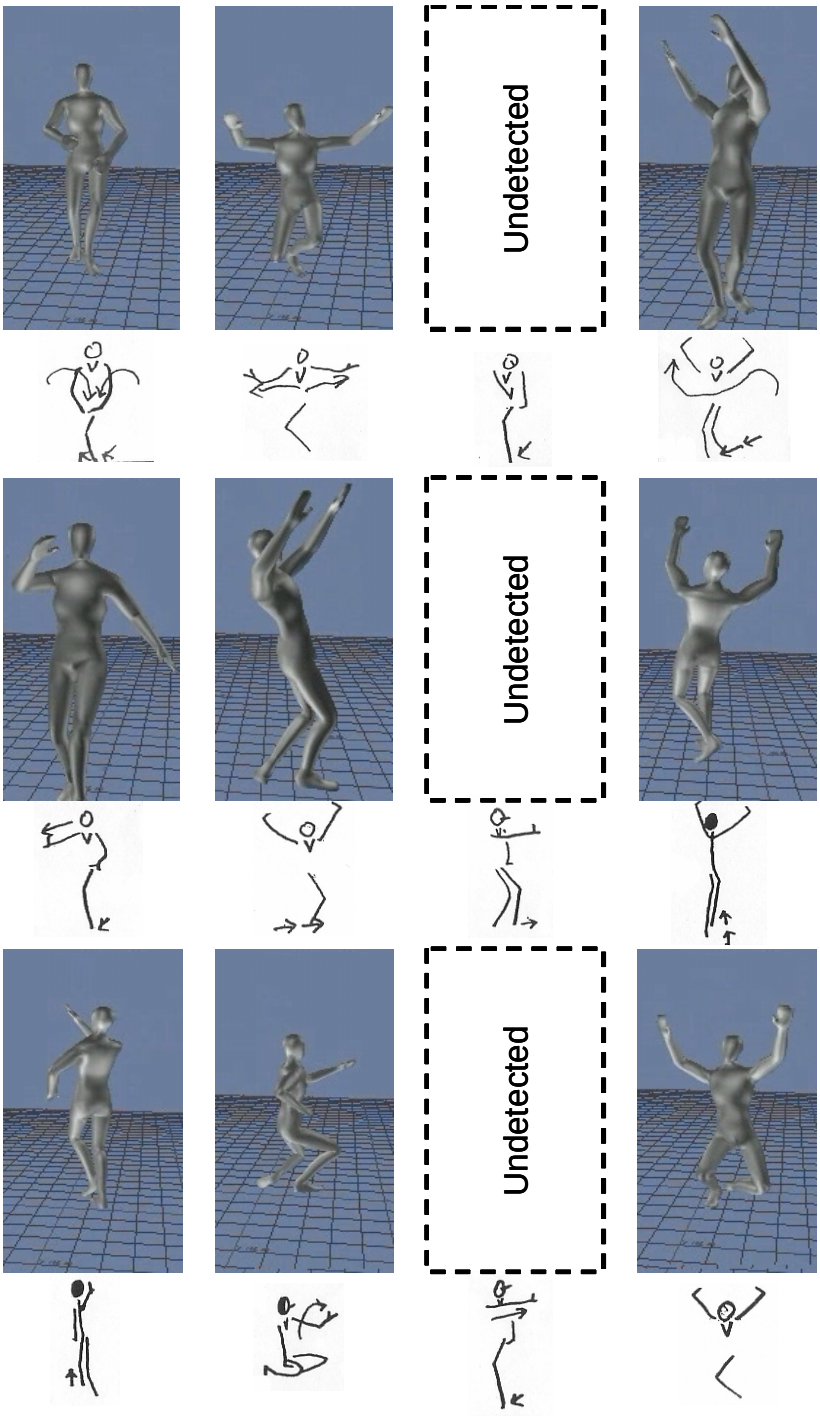


Figure 2.17: All extracted keyposes from the Jongara-bushi dance.

Chapter 3

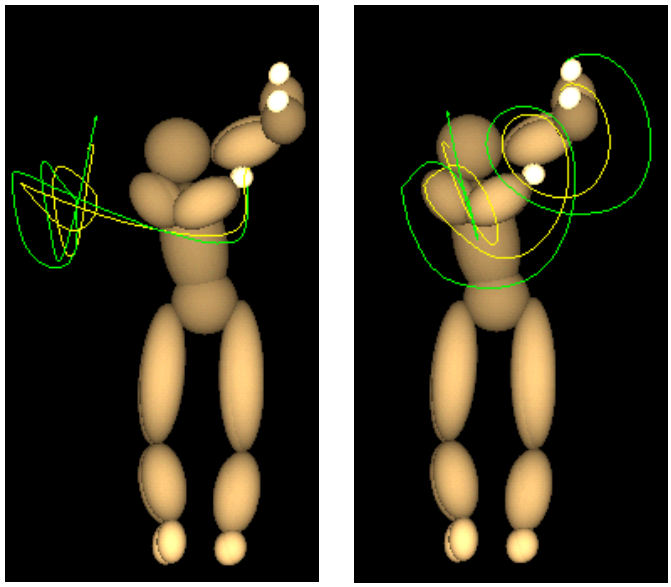
Synthesis of Temporally-Scaled Upper Body Motion Based on Aspects of Human Motion

3.1 Introduction

Synthesizing human motion sequence synchronized with musical rhythm is important for realizing dancing-to-music ability. Toward this, we are developing *sound-feedback system*, in which CG characters and humanoid robots mimic human's improvisational ability for entertainment. In particular, synchronizing recorded human motion data with currently played music plays an important role for the sound feedback system, due to the difference of rhythm. In this chapter, we propose a novel method to temporally scale upper body motion of dance performance for synchronization with music.

To achieve this, we first observe human dance motion and obtain its properties. We then develop a model for the needed modification in synthetic upper body motion based on the speed of corresponding input music, based on insights acquired via this observation.

When we observe structured dance motion performed by humans at normal music playback speeds vs. motion performed using music that is 1.3 times faster, we find that the details of each motion sequence, *style*, differs slightly, though the whole of the dance motion sequence is similar in both cases. An example of this type of motion modification, natural in humans, is shown in Figure 3.1. This phenomenon is derived from the fact that dancers omit details of a dance,



(a) (b)

Figure 3.1: Comparison of hand trajectory differences depending on music playback speed. (a): Comparison of right hand trajectories, and (b): comparison of left hand trajectories. Green lines and yellow lines represent the hand trajectory at normal music playback speed, and 1.3 times faster music playback speeds, respectively.

but retain its essential aspects, if this is necessary to follow faster music. If we therefore observe motion differences in dances performed at different speeds in the frequency domain, we can obtain useful insights on motion detail omission. Based on these insights, we propose a new modeling method and develop some applications useful for CG character animation and humanoid robot motion generation.

First, we describe motion decomposition using a hierarchical B-spline, which is a key technique to accomplish the modeling of motion modification. A B-spline allows us to control frequency resolution by only setting its control points at desired temporal intervals. By using a hierarchical B-spline technique, we observe the differences between motion performed at a normal musical speed vs. motion performed at a faster musical speed. Then we discuss how upper body

motion is modified. Based on our obtained insights, we improve the hierarchical B-spline method to decompose a motion sequence and propose a new framework to modify upper body motion that satisfies kinematic constraints. We also show some applications based on the proposed method.

Our proposed method aims to modify only upper body motion, not whole body motion. However, some methods on motion splicing for CG animation [IF04, HKG06, MZF06] and humanoid robots [NNK*05] have recently been proposed. Motion splicing means that motion of some body parts are transferred to another motion sequence to enrich a motion sequence variation. The combination of these methods and our proposed method could be very useful for whole body motion generation.

3.2 Prior Work

We employ a hierarchical approach to analyze and modify human motion. Previous work related to this approach can be roughly categorized into three domains: a multi-dimensional approach, a parameterization-based approach, and a style-based approach.

Multi-dimensional Approach

A "multi-level" or "multi-dimensional" hierarchical approach has been extensively studied. In computer vision, a *image pyramid*, a multi-level representation of an image [BA83], and its extensions have been used for various purposes such as optical flow estimation [BAHH92] and image and video completion [HB95, WSI04, SMKT06]. In computer graphics, a hierarchical approach has been shown to have potential in efficient point-based rendering [RL00] and scattered data interpolation [LWS97].

Recently, many researchers of human motion animation are focusing on the good potential of a multi-dimensional approach. Bruderlin *et al*. [BW95] and Lee *et al*. [LS01] applied an image pyramid-based approach to motion sequence synthesis; they constructed a hierarchical structure of motion in which each layer contains certain frequency components. By editing a coefficient in each layer, new motion sequences can be generated. This research has influenced us to more deeply analyze the frequency components of human motion.

Safonova *et al*. [SHP04] found that most motion sequences could be represented in a low dimensional space, as few as ten degrees of freedom, because some body portions, e.g. legs and arms are operating in a coordinated way

in most cases. They applied principal component analysis (PCA) to a motion capture database in order to reduce its dimensionality, and thereby synthesized human motion that satisfies both user-specified constraints and physical correctness by solving an optimization in a lower-dimensional space. After this research appeared, many researchers considered dimensionality reduction-based approaches. Chai *et al*. [CH05] and Liu *et al*. [LZWM06] used PCA to reduce the dimensionality of a motion capture database and to estimate full-body marker positions from a commensurately smaller marker set. Forbes *et al*. [FF05] used weighted PCA to convert a high-dimensional motion data set into a low-dimensional space. The resulting low dimensional space converted by PCA has a Euclidean distance metric which makes it simpler to retrieve a desired motion from a motion database. Arikan [Ari06] and Liu *et al*. [LM06] proposed a method to compress a motion database. In their method, first, motion data is segmented and classified. Then, PCA is applied to each cluster to compress similar motions. Mukai *et al*. [MK06] proposed a method to efficiently render human motion animation by converting motion into low-dimensional multilinear spaces. In robotics research, this technique has generated recent interest; there is now a method to control humanoids in PCA spaces [CGM*06].

There are some methods in which the dimensionality of motion data can be reduced by using a motion description method such as Labanotation. Yu *et al*. [YSLG05] proposed a method to retrieve motion from a database using Labanotation descriptors method as query terms. Shen *et al*. [SLY*05] proposed a method to edit motion by adjusting the parameters of Labanotation. Müller *et al*. [MRC05, MR06] proposed a motion retrieval method in which they assumed prior motion initial conditions based on end-effector trajectories.

Parameterization-Based Approach

There are some methods to convert motion into a parameterized representation. Lee *et al*. [LS99] also proposed a method to efficiently resolve spacetime constraints problem using a hierarchical motion decomposition method. For human motion imitation with a humanoid robot, Ruchanurucks *et al*. [RNKI06] proposed a method to decompose motion into a hierarchical B-spline and to optimize the control points of each hierarchical B-spline layer in order to satisfy mechanic constraints of a humanoid robot. Abe *et al*. [ALP06] developed a method to represent momentum curves for motion capture data with a cubic non-uniform B-spline. Considering momentum conservation, their method optimized the control points of B-spline curves in order to satisfy user-specified

constraints.

Style-Based Approach

The concept of *motion style* was introduced by Unuma *et al*. [UAT95]. They analyzed walking motions with some expressions such as happy or sad, and decomposed motion into low frequency components representing basic motion common to various walking motions and high frequency components representing the emotion of a motion, using a Fourier transform. They also synthesized new walking motions by blending the high frequency components. Pullen *et al*. [PB02] employed a similar approach. They observed the differences between motion generated using a keypose-based technique and motion capture data and found that motion capture data looks more natural than keypose-based synthetic motion because of its details, i.e. the high frequency components, of motion capture data that keypose-based synthetic motion does not have. Therefore, their method extracted the high frequency components from motion capture data and superimposed them onto keypose-based motion in order to enable animators to synthesize more human-like animation using the traditional keypose-based synthesis approach. Nakazawa *et al*. [NNI03, NNI04] employed a keypose-based approach to analyze style components of dance motion. First, they detected keyposes in dance motion and extracted *base motions* by interpolating the requisite joint angle trajectories between keyposes. Individual differences, which they defined as motion style, were extracted by calculating the differences between actual motions and base motions; new dance motions were generated by blending motion style components.

Recent style-based approaches have employed on stochastic models. Brand *et al*. [BH00] employed a hidden Markov model (HMM) to analyze style in a motion database. Using an HMM, new motion sequences were synthesized by adjusting a small number of stylistic parameters. Urtasun *et al*. [UGB*04] and Glardin *et al*. [GBT04] used PCA to analyze motion data sets that contained various walking styles. In their experiments, they captured walking motions at varying speeds. Applying PCA to their motion data, principal components containing walking styles were derived as a function of walking speed, and new walking motions were interpolated and extrapolated by adjusting weights of principal components. Shapiro *et al*. [SCF06] used independent component analysis (ICA) to analyze and synthesize stylistic motions. ICA is slightly different from PCA in that PCA space has an orthogonal basis while ICA space has a non-orthogonal basis; their method can effectively represent motion style as a set of independent

components. Faloutsos *et al*. [FvdPT01] proposed a motion controller based on a support vector machine (SVM) learning method that could generate various kinds of motion such as walking, running, falling down and so on. Grochow *et al*. [GMHP04] proposed a method of inverse kinematics computation based on the scaled Gaussian process latent variable model (SGPLVM). Hsu *et al*. [HPP05] proposed a method to transfer motion style using a linear dynamic system (LDS) model and their iterative motion warping method.

The goals of previous methods are quite different from ours in that most previous methods aim to synthesize new motions as efficiently as possible, whereas our goals are to analyze the details of human motion and to synthesize temporally-scaled realistic motion based on our analysis of these properties. MacCann *et al*.'s approach [MPS06] is somewhat similar to our proposed method; they aimed to temporally scale a motion sequence while retaining physical consistency.

3.3 Hierarchical B-Spline

This section describes how we decompose motion sequences using a hierarchical B-spline.

3.3.1 B-Spline

B-spline is one of the best-known parameterized curves; it is a generalized version of a Bèzier curve. Let a vector known as the knot vector be defined as

$$\boldsymbol{T} = (t_0, t_1, t_2, \cdots, t_M)\,. \tag{3.1}$$

Note that all the elements of the knot vector should satisfy the criterion that $t_i \leq t_{i+1}$. Given a control point set of a B-spline curve $\boldsymbol{Q} = \{\boldsymbol{Q}_i | i = 1, \cdots, N\}$ and the knot vector, the degree of the B-spline curve is defined as

$$d \equiv M - N - 1, \tag{3.2}$$

and its basis functions are recursively defined as

$$B_{i,1}(t) = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases}, \tag{3.3}$$

and

$$B_{i,d+1} = \frac{t - t_i}{t_{i+d+1} - t_i} B_{i,d}(t) + \frac{t_{i+d+2} - t}{t_{i+d+2} - t_{i+1}} B_{i+1,d}(t) \quad (d \neq 1) \tag{3.4}$$

for the B-spline whose degree is d. From these parameters and functions, the B-spline curve $\boldsymbol{f}(t)$ is given as

$$\boldsymbol{f}(t) = \sum_{i=0}^{n} B_{i,d}(t) \boldsymbol{Q}_i. \tag{3.5}$$

If each interval of neighboring knots is the same, the B-spline curve is said to be uniformal.

For a uniformal cubic B-spline of degree three ($d = 3$), the basis function is formulated as

$$B_{0,4}(t) = \frac{1}{6}(1 - t)^3, \tag{3.6a}$$

$$B_{1,4}(t) = \frac{3t^3 - 6t^2 + 4}{6}, \tag{3.6b}$$

$$B_{2,4}(t) = \frac{-3t^3 + 3t^2 + 3t + 1}{6}, \tag{3.6c}$$

$$B_{3,4}(t) = \frac{1}{6}t^3, \tag{3.6d}$$

for $0 \leq t < 1$. Here, denoting a control point at knot t as $\boldsymbol{Q}_t$, the cubic B-spline curve is given as

$$\boldsymbol{f}(t) = \sum_{i=0}^{3} B_{i,4}(t - \lfloor t \rfloor) \boldsymbol{Q}_{\lfloor t \rfloor + i - 1}. \tag{3.7}$$

As this formulation is a three dimensional polynomial equation with regard to t, the cubic B-spline is two times differentiable. Therefore, continuous acceleration can be approximated using a B-spline.

3.3.2 Motion Approximation Using a B-Spline

To model motion using a B-spline, let the knot range of a motion sequence be $[0, n]$ and the number of motion frames be m. Assume that the knot of each control point lies within $[0, n]$. In order to approximate motion sequence $\mathcal{D} = \left\{\boldsymbol{d}(t_i) | t_i < t_{i+1}, {}^\forall t_i \in [0, n], i = 1, \cdots, m\right\}$ with a B-spline, a least-squares solution is used to determine the control point set $\hat{\boldsymbol{Q}} = \left\{\hat{\boldsymbol{Q}}_i | i = 0, \cdots, n\right\}$:

$$\hat{\boldsymbol{Q}} = \arg\min_{\boldsymbol{Q}} \sum_i \left| \boldsymbol{f}(t_i) - \boldsymbol{d}(t_i) \right|^2. \tag{3.8}$$

This least-squares problem can be solved using a pseudo-inverse.

To achieve this, we assume that the motion sequence $\mathcal{D}$ can be represented with the B-spline curve:

$$\boldsymbol{d}(t) \simeq \sum_{i=0}^{3} B_{i,4}(t - \lfloor t \rfloor)\hat{\boldsymbol{Q}}_{\lfloor t \rfloor + i - 1}. \tag{3.9}$$

From all the data points, the following linear system of equations is formed:

$$\begin{pmatrix} \boldsymbol{d}(t_1) \\ \boldsymbol{d}(t_2) \\ \vdots \\ \boldsymbol{d}(t_m) \end{pmatrix} \simeq N \begin{pmatrix} \hat{\boldsymbol{Q}}_0 \\ \hat{\boldsymbol{Q}}_1 \\ \vdots \\ \hat{\boldsymbol{Q}}_n \end{pmatrix}, \tag{3.10}$$

where N is the $m \times (n+1)$ matrix whose elements are the basis function of the B-spline represented as

$$N_{ij} = \begin{cases} B_{j+1-\lfloor t_i \rfloor,4}(t_i - \lfloor t_i \rfloor) & \text{if } j \leq t_i < j+1 \\ 0 & \text{otherwise} \end{cases}. \tag{3.11}$$

Matrix N may appear to have a number of elements, but each row vector of N has at most four non-zero values, and N is therefore a sparse matrix.

Since N is not a square matrix in most cases, we use a pseudo-inverse matrix to solve the linear system of equations. The pseudo-inverse matrix N^+ is calculated as follows:

$$N^+ = \begin{cases} N^T \left(NN^T\right)^{-1} & \text{if } (n+1) < m \\ \left(N^T N\right)^{-1} N^T & \text{if } (n+1) > m \end{cases}. \tag{3.12}$$

Using this, the control point set can be estimated as

$$\begin{pmatrix} \hat{\boldsymbol{Q}}_0 \\ \hat{\boldsymbol{Q}}_1 \\ \vdots \\ \hat{\boldsymbol{Q}}_n \end{pmatrix} = N^+ \begin{pmatrix} \boldsymbol{d}(t_1) \\ \boldsymbol{d}(t_2) \\ \vdots \\ \boldsymbol{d}(t_m) \end{pmatrix}, \tag{3.13}$$

and this provides a least-squares solution which satisfies Equation (3.8).

An example of this kind of data approximation is shown in Figure 3.2, which indicates that input data can be roughly approximated by a B-spline curve.

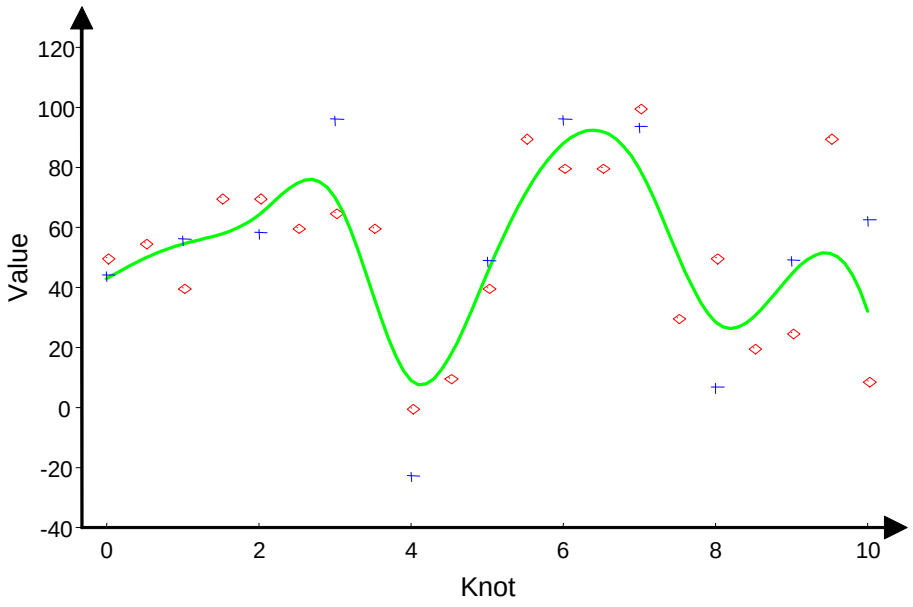


Figure 3.2: B-spline fitting. Red points: input data sequence, blue crosses: B-spline control points estimated from Equation (3.13), and green curve: B-spline curve approximating the input data sequence.

3.3.3 Motion Decomposition Using a Hierarchical B-Spline

As shown in Figure 3.2, it can be difficult to approximate high frequency components of the original input sequence with only one B-spline curve; approximation using a hierarchical B-spline can solve this problem. Hierarchical B-spline consists of using a series of B-spline curves with different knot spacings; higher layers of a hierarchical B-spline are based on finer knot spacing which can preserve the higher frequency components of the original sequence.

Hierarchical B-spline construction is illustrated in Figure 3.3 First, the input data sequence is approximated with a B-spline curve f_0 using Equation (3.13), which serves as a smoothed initial approximation. There is a point-by-point difference between the input data and the estimated B-spline curve, however, expressed by

$$\Delta_1(2t_i) = d(t_i) - f_0(t_i). \tag{3.14}$$

A "finer" layer which preserves higher frequency content can be constructed by

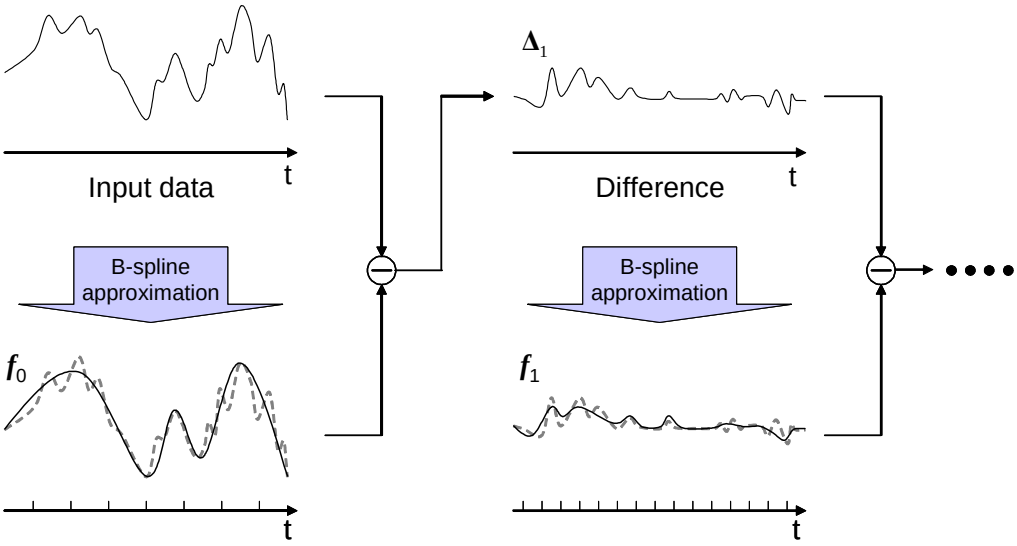


Figure 3.3: Illustration of hierarchical B-spline construction. First layer f_0 roughly approximates original input sequence, while other "finer" layers f_k approximate difference sequence $\mathcal{D}_k = \left\{\Delta_k\left(2^k t_i\right) \middle| \Delta_k\left(2^k t_i\right) = d(t_i) - \sum_{l=0}^{k-1} f_l\left(2^l t_i\right)\right\}$.

fitting a B-spline curve f_1 to the difference sequence $\mathcal{D}_1 = \{\Delta_1(2t_i) | i = 1, \cdots, m\}$. To preserve high frequency components, the knot spacing of f_1 is chosen to be half the interval of f_0.

The same process can be used to construct the next layer f_2: the next finer B-spline curve f_2 is obtained from the difference sequence $\mathcal{D}_2$ whose elements Δ_2 are given as

$$\Delta_2\left(2^2 t_i\right) = d(t_i) - \left(f_0(t_i) + f_1(2t_i)\right). \tag{3.15}$$

This can be continued as desired; the k-th layer of a hierarchical B-spline is recursively estimated from the difference sequence

$$\mathcal{D}_k = \left\{\Delta_k\left(2^k t_i\right) \middle| \Delta_k\left(2^k t_i\right) = d(t_i) - \sum_{l=0}^{k-1} f_l\left(2^l t_i\right)\right\}. \tag{3.16}$$

The approximated data sequence f is therefore given as

$$f = \sum_{l=0}^{L} f_l\left(2^l t_i\right), \tag{3.17}$$

in which L is the number of hierarchical B-spline layers.

The use of a hierarchical B-spline decomposition offers these advantages:

- One can attain any desired frequency resolution by adjusting the knot spacing.
- The optimization of a generated motion sequence is relatively easy.

In our proposed method, the desired frequency resolution for an input motion sequence depends on the musical rhythm. We set the knot spacing to correspond to the frequencies of the musical rhythm so that we can compare the same frequency components of motions captured at different speeds.

Optimization is likewise simpler when using a hierarchical B-spline because the input sequences are converted into sets of control points. Thus, the optimization procedure for a motion sequence merely manipulate the estimated control points, not the sequence itself, frame-by-frame; the method's computational cost is therefore not high.

3.4 Observations of Human Motion

Using a hierarchical B-spline-based decomposition technique, we observed human dance motion. This section describes how we observed motion and the acquired relationship between human motion and music playback speed.

Mean and Variance of Joint Angles

Using a motion capture system, we captured the Aizu-bandaisan dance, a classical Japanese folk dance, at its normal speed as performed by a dance master. We then asked the dance master to perform the dance with input music played 1.2 times faster than this speed, and 1.5 times faster. Motion sequences at each of these three speed were captured five times in order to investigate motion variance, so a total of 15 datasets were considered in this experiment.

We obtained the marker position of each joint angle through an optical motion capture system (see Appendix B), and then converted the marker position data to joint angles (see Appendix C). Using quaternion algebra, the j-th joint angle $\boldsymbol{q}_j$ can be represented as follows:

$$\begin{aligned} \boldsymbol{q}_j &= \exp(\boldsymbol{v}_j) \\ &= \cos|\boldsymbol{v}_j| + \frac{\boldsymbol{v}_j}{|\boldsymbol{v}_j|}\sin|\boldsymbol{v}_j|, \end{aligned} \tag{3.18}$$

where $\boldsymbol{q}_j$ is a unit quaternion, and $\boldsymbol{v}_j$ is a 3-dimensional vector whose unit vector $\frac{\boldsymbol{v}_j}{|\boldsymbol{v}_j|}$ represents a rotation axis, and whose norm represents half the joint rotation (see Appendix D for details).

The mean joint angle $\bar{\boldsymbol{v}}$ is calculated as

$$\bar{\boldsymbol{v}}_j(t) = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{v}_j^i(t), \tag{3.19}$$

where $\boldsymbol{v}_j^i$ represents the j-th joint angle of the i-th motion sequence, and N represents the number of input motion sequences (in our case, $N = 5$ for a given music playback speed). Using our computed sequence of mean motion, we also calculate the variation var of j-th joint angle at t-th temporal frame:

$$var_j(t) = \frac{1}{N-1}\sum_{i=1}^{N} \left(1 - \boldsymbol{u}_j^i(t) \cdot \bar{\boldsymbol{u}}_j(t)\right), \tag{3.20}$$

where

$$\boldsymbol{u}_j^i \equiv \frac{\left({\boldsymbol{v}_j^i}^T, 1\right)^T}{|{\boldsymbol{v}_j^i}^T, 1|}. \tag{3.21}$$

The variance is calculated in a 4D homogeneous coordinate system which accounts for both magnitude and direction differences [BFB94, SMKT06].

We set the knot spacing to the estimated musical rhythm mentioned in Chapter 2, and then apply a hierarchical B-spline decomposition technique. We used up to five layers in our motion decomposition and observed the mean and variance of each reconstructed motion. Our choice of five layers was arbitrary, but empirically found to be enough to reconstruct high frequency components of human motion.

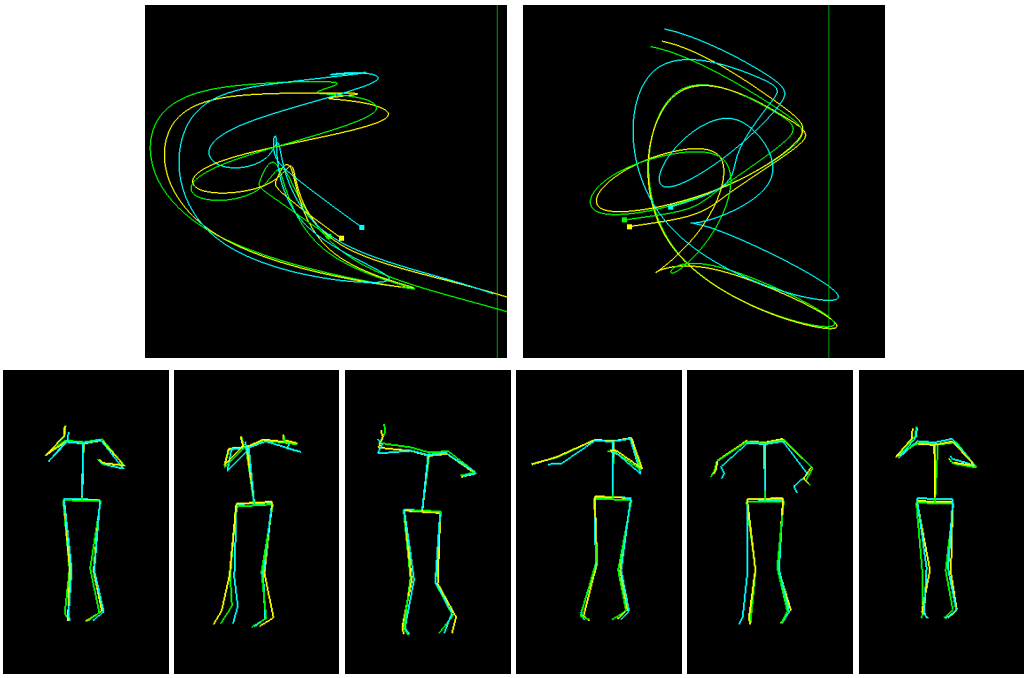


Figure 3.4: Comparison of mean motion reconstructed using a single-layer B-spline. Top left: joint angle trajectories of left shoulder, top right: joint angle trajectories of right shoulder, and bottom row: reconstructed motion sequences. Green, yellow, and light blue lines represent the motion sequences at normal, 1.2 times faster, and 1.5 times faster music playback speeds, respectively.

Motion Comparison Using a Single-Layer B-Spline

The mean angle trajectories of left and right shoulders are shown in Figure 3.4. From this result, we obtained the following insights:

- The shape of the joint angle trajectory at a normal music playback speed is similar to the trajectory at music playback which is 1.2 times faster.
- The shape of the joint angle trajectory at a normal music playback speed is also similar to the trajectory of music playback which is 1.5 times faster music playback speed, but their details, such as curvature, differ visibly from each other.

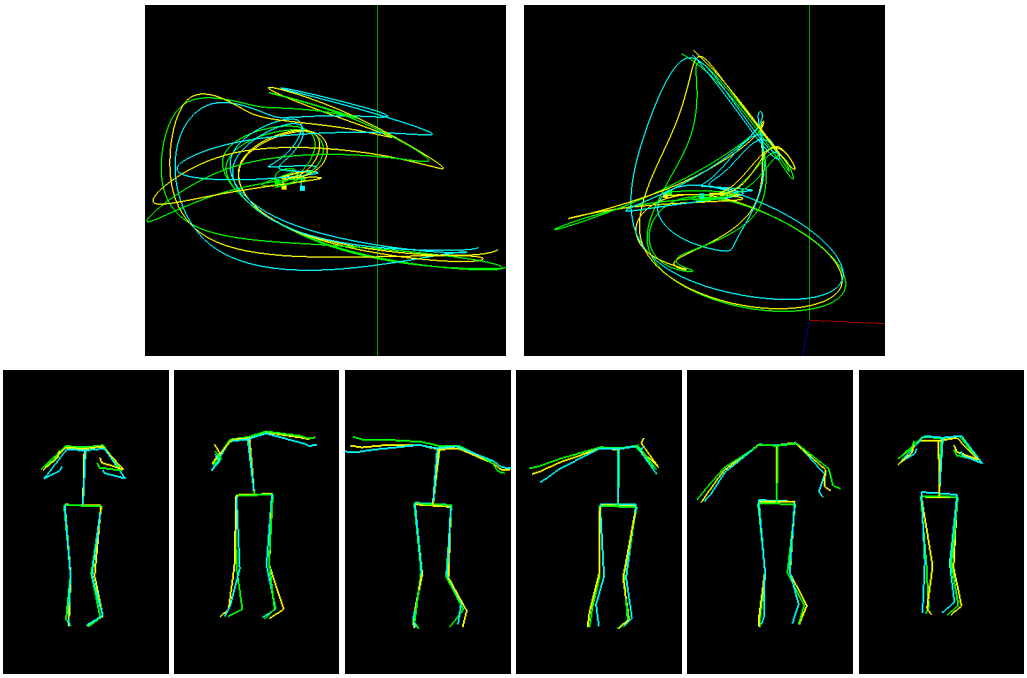


Figure 3.5: Comparison of mean motion reconstructed using a two-layer hierarchical B-spline. Top left: joint angle trajectories of left shoulder, top right: joint angle trajectories of right shoulder, and bottom row: reconstructed motion sequences. Green, yellow, and light blue lines represent the motion sequences at normal, 1.2 times faster, and 1.5 times faster music playback speeds, respectively.

Motion Comparison Using a Two-Layer Hierarchical B-Spline

The mean angle trajectories of left and right shoulders are shown in Figure 3.5. From this result, we obtained the following insights:

- The shape of the joint angle trajectory at a normal music playback speed slightly differs from that of 1.2 times faster music playback speed, especially in the trajectory's sharpest curves.
- The shape of the joint angle trajectory using 1.5 times faster music playback speed appears to be a smoothed version of the normal music playback speed trajectory.

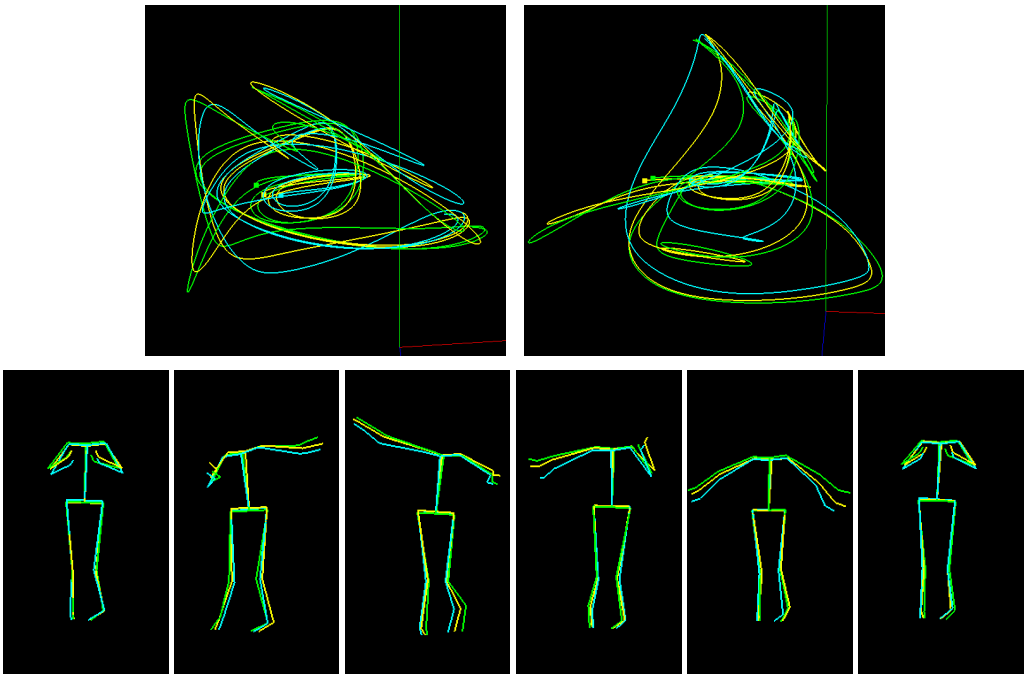


Figure 3.6: Comparison of mean motion reconstructed using a three-layer hierarchical B-spline. Top left: reconstructed joint angle trajectories of left shoulder, top right: joint angle trajectories of right shoulder, and bottom row: reconstructed motion sequences. Green, yellow, and light blue lines represent the motion sequences at normal, 1.2 times faster, and 1.5 times faster music playback speeds, respectively.

Motion Comparison Using a Three-Layer Hierarchical B-Spline

The mean angle trajectories of left and right shoulders are shown in Figure 3.6. From this result, we obtained the following insights:

- The differences among each joint angle trajectory are becoming noticeable.
- The shape of the joint angle trajectory at 1.2 times faster music playback speed appears to be a slightly smoothed version of the trajectory at a normal music playback speed.

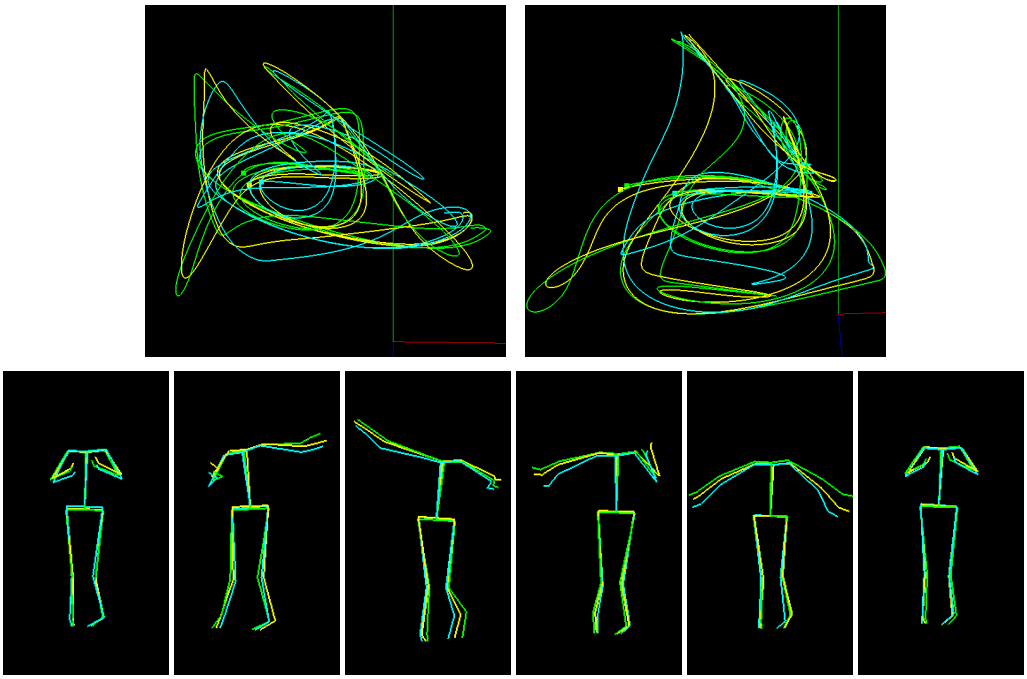


Figure 3.7: Comparison of mean motion reconstructed using a four-layer hierarchical B-spline. Top left: joint angle trajectories of left shoulder, top right: joint angle trajectories of right shoulder, and bottom row: reconstructed motion sequences. Green, yellow, and light blue lines represent the motion sequences at normal, 1.2 times faster, and 1.5 times faster music playback speeds, respectively.

Motion Comparison Using a Four-Layer Hierarchical B-Spline

The mean angle trajectories of left and right shoulders are shown in Figure 3.7. From this result, we obtained the following insights:

- The differences among each joint angle trajectory get clearly noticeable.
- The shape of the joint angle trajectory at 1.5 times faster music playback speed seems to be a smoothed version of the trajectory at 1.2 times faster music playback speed.
- The shape of the joint angle trajectory at 1.2 times faster music playback speed seems to be a smoothed version of the trajectory at normal music playback speed.

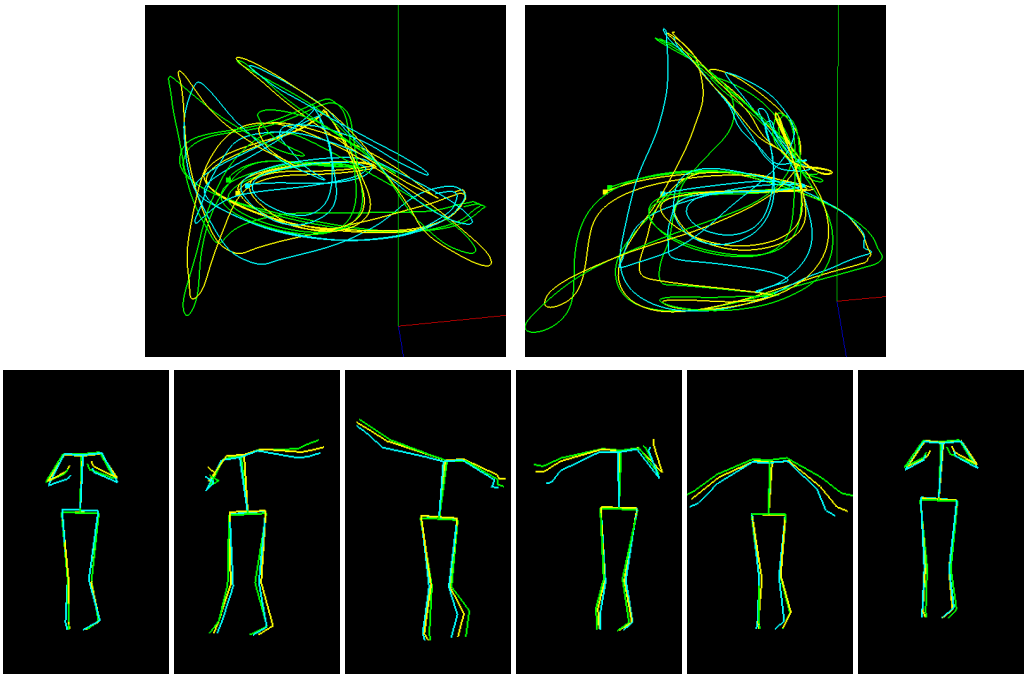


Figure 3.8: Comparison of mean motion reconstructed using a five-layer hierarchical B-spline. Top left: joint angle trajectories of left shoulder, top right: joint angle trajectories of right shoulder, and bottom row: reconstructed motion sequences. Green, yellow, and light blue lines represent the motion sequences at normal, 1.2 times faster, and 1.5 times faster music playback speeds, respectively.

Motion Comparison Using a Five-Layer Hierarchical B-Spline

The mean angle trajectories of left and right shoulders are shown in Figure 3.8. In this case, we could not find any other noticeable difference between each joint angle trajectory.

Comparison of Variances of Joint Angle Trajectories

Figure 3.9 shows the variance of left shoulder angle trajectories reconstructed using a five-layer hierarchical B-spline. According to this observation, it is confirmed that there are some valleys where each variance sequence is locally

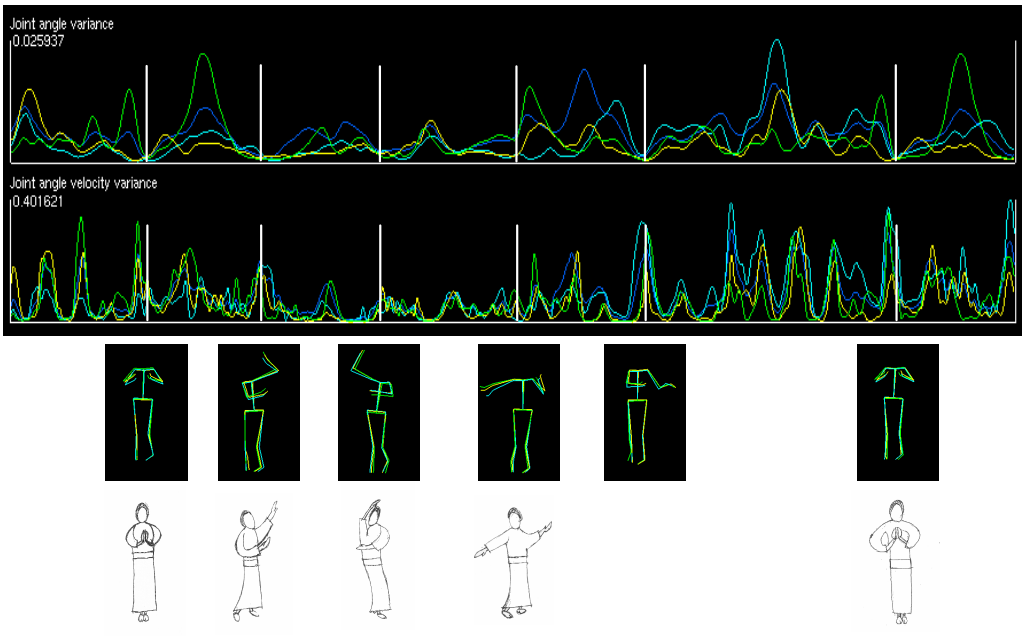


Figure 3.9: Variance graph of joint angle and angular velocity sequences of left shoulder reconstructed using a five-layer hierarchical B-spline. Top row: anglular variances, second row: angular velocity variances, third row: postures whose joint angle variance is local minimum, and bottom row: keyposes specified by the dance masters. Green, yellow, light blue, and blue lines represent the variances of motion sequences at normal, 1.2 times faster, 1.5 times faster music playback speeds, and the variance of all sequences, respectively.

minimum, and that most valleys represent the keyposes specified by the dance masters.

Figure 3.10 shows all the variance sequences. It is also confirmed that most of the keypose instances have a low variance, even if we used fewer layers to reconstruct the motions. However, using more number of B-spline layers reduces the variance at the keypose instances. Additionally, each variance of joint angular velocity has a local maximum value around each keypose. This is because dancers naturally attempt to match their keyposes with the musical rhythm.

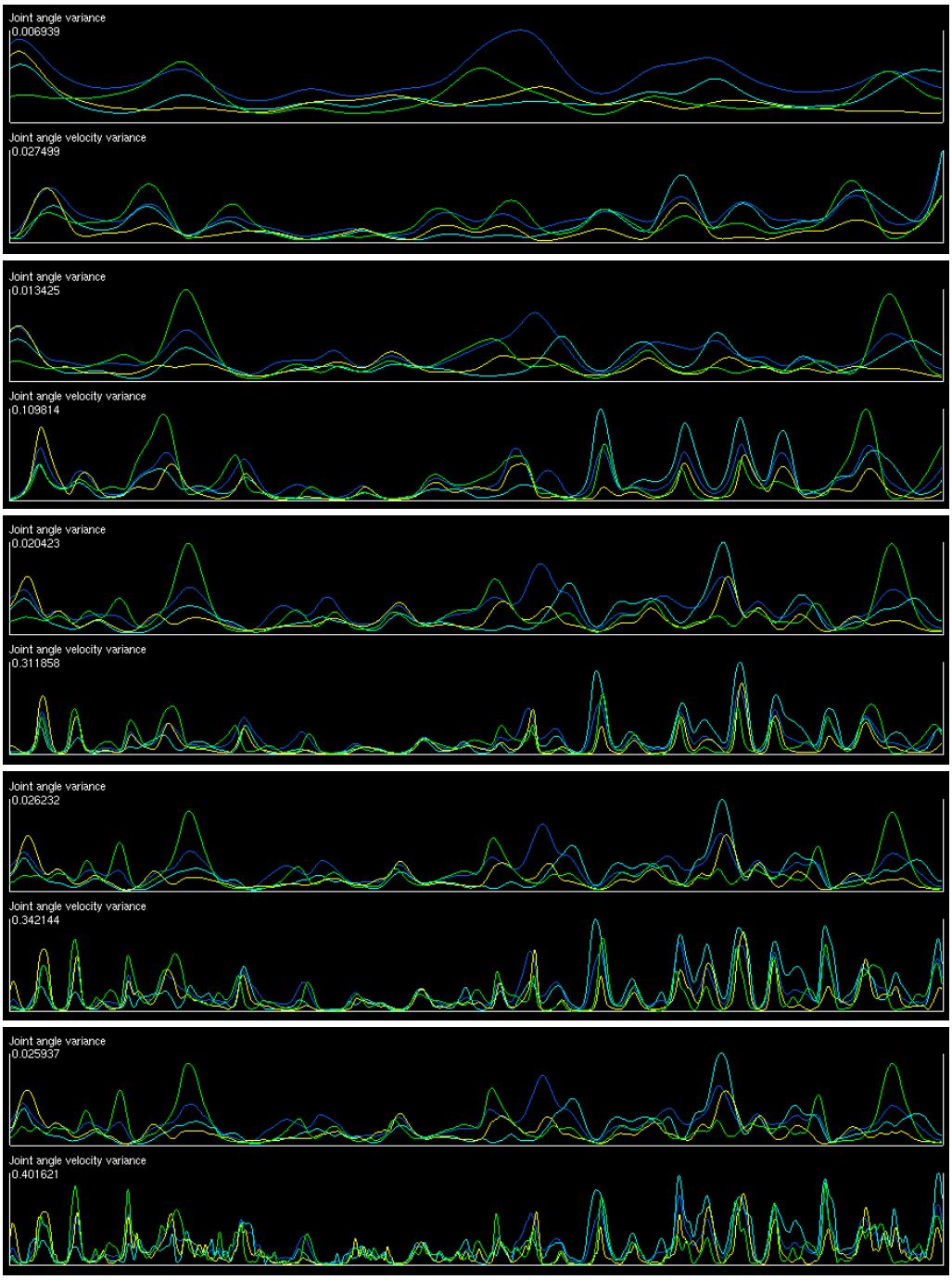


Figure 3.10: Variance graphs of left shoulder angle and angular velocity sequences. Green: using normal playback speed music, yellow: using 1.2 times faster music, light blue: using 1.5 times faster music, and blue: variance calculated from all motions. From top to bottom: the number of B-spline layers is 1, 2, 3, 4, and 5, respectively.

Summary of Motion Observations

According to the observations of the mean motion, the motion performed at faster music playback speed has fewer high frequency components. Additionally, according to the observations of the variance sequences, the keyposes mentioned in Chapter 2 are preserved in all cases. In the above figures, we show the trajectories of left and right shoulders, but the same is true for other joint angle trajectories and the motions performed by other dancers. Considering these insights, we assume the following aspects of human motion in order to model modifications of upper body motion:

1. High frequency components of human motion should be attenuated when the music playback speed is too quick to follow.

2. Keyposes will be preserved even if high frequency components are attenuated.

3.5 Motion Modification Based on Kinematic Constraints

Based on these observations, in this section, we design an algorithm to modify upper body motion to follow input music playback speed. Our motion modification method consists of two steps: hierarchical motion decomposition considering keypose information, and motion reconstruction that satisfies kinematic constraints. A summarized algorithm of our motion modification method is described in Algorithm 3.1.

3.5.1 Hierarchical Motion Decomposition Using Keypose Information

Remembering our insight that keyposes will be preserved even if high frequency components become attenuated, we can improve the method of motion decomposition described in Equation (3.13). To achieve this, our motion decomposition method should consider the posture and velocity information of the keyposes.

To consider posture information, we could apply an optimization method which uses hard constraints such as the SVD-based optimization method [LH74]. However, in our case, most control points must satisfy these kinds of hard constraints if this method is applied. Therefore, a better solution is that we can densely sample input motion sequence around keyposes and sparsely sample it in other parts, and then use these samples to form a linear system of equations.

Algorithm 3.1 Motion modification

---

input: motion sequence $\boldsymbol{q}$
input: musical rhythm information M
input: keypose information K
input: kinematic constraints C
output: modified motion sequence $\boldsymbol{q}'$
local: motion segment s
local: hierarchical B-spline H
local: the number of hierarchical B-spline layers L
local: weighting factor w

```
1  for each joint of upper body
2    H ← DecomposeMotion(q, M, K)
3    s ← SegmentMotion(H, M)
4    for each motion segment
5      L, w ← Initialize()
6      while
7        L, w ← UpdateParameters(L, w)
8      until ComposeMotion(s, L, w) satisfies C
9      s ← ComposeMotion(s, L, w)
10   end for
11   q' ← InterpolateMotionSegments(s)
12 end for
```

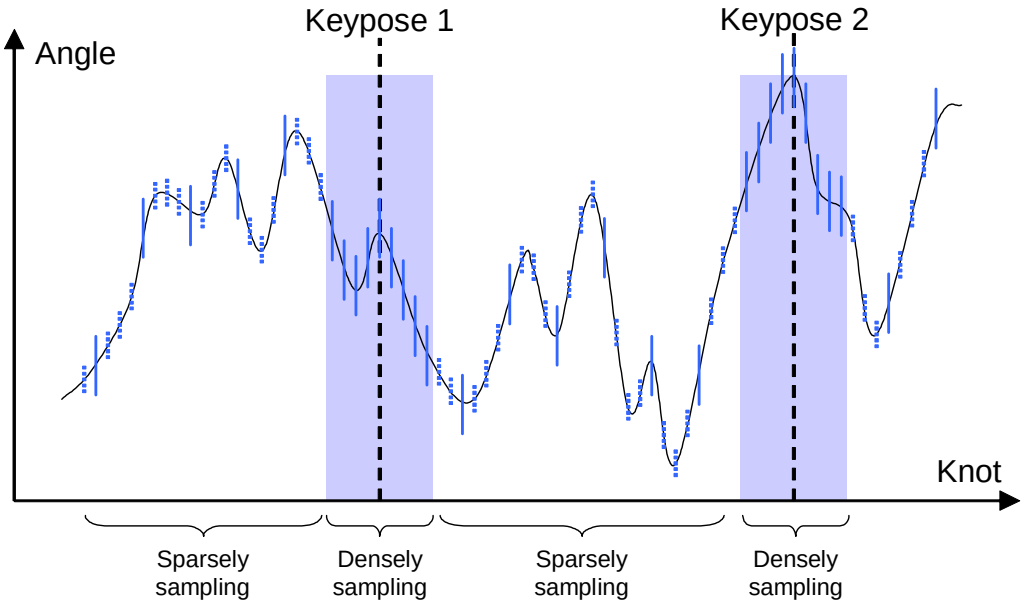


Figure 3.11: Illustration of our sampling method for motion decomposition. For hierarchical B-spline construction, our method densely samples motion sequence around the keyposes, while sparsely sampling motion sequence in other parts. In this example, our method considers only the data represented by the vertical solid lines, and ignores the data represented by the vertical broken lines, to estimate control points of hierarchical B-spline.

Figure 3.11 provides an illustration of our data-sampling method for motion decomposition. All vertical lines in this illustration represent originally sampled data, and our method uses only the solid lines shown among them.

With regard to velocity information, the movements of a dancer's arms and hands are stopping around keyposes; velocity of the hands and arms are approximately zero at keyposes. We exploit this useful property of keyposes as velocity information in our motion decomposition method. The derivation of B-spline curve is represented as

$$\frac{d\boldsymbol{f}}{dt} = \sum_{i=0}^{3} \frac{d}{dt} B_{i,4}(t - \lfloor t \rfloor) \boldsymbol{Q}_{\lfloor t \rfloor + i - 1}, \tag{3.22}$$

where

$$\frac{d}{dt}B_{0,4}(t) = -\frac{1}{2}(1-t)^2, \tag{3.23a}$$

$$\frac{d}{dt}B_{1,4}(t) = \frac{3t^2-4t}{2}, \tag{3.23b}$$

$$\frac{d}{dt}B_{2,4}(t) = \frac{-3t^2+2t+1}{2}, \tag{3.23c}$$

$$\frac{d}{dt}B_{3,4}(t) = \frac{1}{2}t^2. \tag{3.23d}$$

From all the keyposes, we form a linear system of equations to satisfy the velocity constraints:

$$\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix} \simeq N^{vel} \begin{pmatrix} \hat{\mathbf{Q}}_0 \\ \hat{\mathbf{Q}}_1 \\ \vdots \\ \hat{\mathbf{Q}}_n \end{pmatrix}, \tag{3.24}$$

where N^{vel} represents a (the number of keyposes)$\times(n+1)$ matrix whose elements are given as

$$N^{vel}_{ij} = \begin{cases} \frac{d}{dt}B_{j+1-\lfloor t_i \rfloor,4}(t_i - \lfloor t_i \rfloor) & \text{if } j \le t_i < j+1 \\ 0 & \text{otherwise} \end{cases}. \tag{3.25}$$

Considering both the posture and velocity constraints, the motion decomposition method is modified as

$$\begin{pmatrix} \boldsymbol{d}(t_1) \\ \boldsymbol{d}(t_2) \\ \vdots \\ \boldsymbol{d}(t_m) \\ \hline \\ 0 \\ \\ \end{pmatrix} \simeq N^{keypose} \begin{pmatrix} \hat{\mathbf{Q}}_0 \\ \hat{\mathbf{Q}}_1 \\ \vdots \\ \hat{\mathbf{Q}}_n \end{pmatrix}$$

$$= \begin{pmatrix} \\ N^{pos} \\ \\ \hline \\ N^{vel} \\ \\ \end{pmatrix} \begin{pmatrix} \hat{\mathbf{Q}}_0 \\ \hat{\mathbf{Q}}_1 \\ \vdots \\ \hat{\mathbf{Q}}_n \end{pmatrix}, \tag{3.26}$$

where N^{pos} represents a coefficient matrix of B-spline basis functions based on our densely/sparsely sampling method. For each layer of hierarchical B-spline, we can estimate the control points by solving Equation (3.26)

$$\begin{pmatrix} \hat{Q}_0 \\ \hat{Q}_1 \\ \vdots \\ \hat{Q}_n \end{pmatrix} = \left(\begin{array}{c} N^{pos} \\ \hline N^{vel} \end{array} \right)^{+} \left(\begin{array}{c} \boldsymbol{d}(t_1) \\ \boldsymbol{d}(t_2) \\ \vdots \\ \boldsymbol{d}(t_m) \\ \hline 0 \end{array} \right), \tag{3.27}$$

and decompose the input motion sequence.

3.5.2 Motion Modification Based on Kinematic Constraints

Every body part has kinematic constraints: the joint angles and angular speed of arms have natural limitations. Our motion modification method reconstructs motion based on the kinematic constraints from our constructed hierarchical B-spline.

In this step, we first segment the motion sequence to correspond to music rhythm frames, and then optimize each motion segment such that a resulting motion sequence must satisfy certain kinematic constraints. A resulting joint angle trajectory $\boldsymbol{q}'$ of motion segment $\mathcal{M}$ can be represented as

$$\boldsymbol{q}'(t;\mathcal{M}) = \exp\left(\sum_{l=1}^{L^{\mathcal{M}}} w_l^{\mathcal{M}} \boldsymbol{f}_l\left(2^l t\right)\right), \tag{3.28}$$

where $L^{\mathcal{M}}$ is the number of the hierarchical B-spline layers to be used for motion reconstruction, and $w_l^{\mathcal{M}}$ is a weighting factor for each layer in the motion segment $\mathcal{M}$ within the range $[0, 1]$ for the currently considered joint. Our optimization process involves determining the $L^{\mathcal{M}}$ and $w_l^{\mathcal{M}}$ that can satisfy certain kinematic constraints:

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{v}_{\text{original}}(t) - \sum_{l=1}^{L^{\mathcal{M}}} w_l^{\mathcal{M}} \boldsymbol{f}_l(2^l t)\| \\ \text{subject to} & \theta_{\min} \le |\boldsymbol{v}'| \le \theta_{\max}, \\ & \dot{\theta}_{\min} \le |\dot{\boldsymbol{v}}'| \le \dot{\theta}_{\max}, \\ & \text{etc.}, \end{array} \tag{3.29}$$

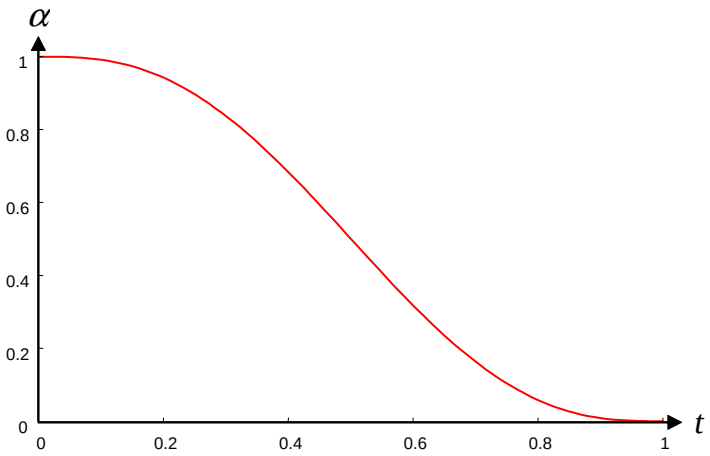


Figure 3.12: Quintic polynomial equation $\alpha(t) = -6t^5 + 15t^4 - 10t^3 + 1$ for motion interpolation. Using this polynomial equation, a resulting motion satisfies the C^2 continuity.

where $v_{\text{original}} = \log\left(q_{\text{original}}\right)$, $v' = \log(q')$, and $\theta_{\min}$, $\theta_{\max}$, $\dot{\theta}_{\min}$, $\dot{\theta}_{\max}$, etc. are predefined kinematic constraints.

First, $L^{\mathcal{M}}$ and $w_l^{\mathcal{M}}$ are initialized with the maximum number of desired hierarchical B-spline layers to consider and 1, respectively. The first step of the optimization process is to determine the optimum number of layers $L^{\mathcal{M}}$. According to our insights obtained through the observations, when musical rhythm gets faster, the higher frequency components of a joint angle trajectory will be attenuated to catch up with the musical rhythm. If a joint angle or joint angular velocity of a body part is beyond a natural limit, the numbers of hierarchical B-spline layers to be used for motion reconstruction are gradually reduced, in successive passes, in order to satisfy all the constraints. In this process, a discontinuity might develop between neighboring motion segments if there ends up being a difference in the number of used layers. So we apply motion blending around the discontinuities. Let $\mathcal{A}$ and $\mathcal{B}$ be neighboring motion segments. To accomplish blending, we interpolate the joint angle sequence of neighboring motion segments using SLERP interpolation (see Appendix D):

$$q_j(t) = SLERP\left(q_j^{\mathcal{A}}(t),\ q_j^{\mathcal{B}}(t);\ \alpha\left(\frac{t - t_{st}}{L}\right)\right), \tag{3.30}$$

where t_{st} represents a starting frame of interpolation, L represents the duration of interpolation, and $\alpha(t)$ is a quintic polynomial equation given as

$$\alpha(t) = -6t^5 + 15t^4 - 10t^3 + 1. \tag{3.31}$$

This quintic polynomial equation is a C^2 continuous function such that

$$\alpha(0) = 1, \tag{3.32a}$$
$$\alpha(1) = 0, \tag{3.32b}$$
$$\frac{d}{dt}\alpha(0) = 0, \tag{3.32c}$$
$$\frac{d}{dt}\alpha(1) = 0, \tag{3.32d}$$
$$\frac{d^2}{dt^2}\alpha(0) = 0, \tag{3.32e}$$
$$\frac{d^2}{dt^2}\alpha(1) = 0. \tag{3.32f}$$

The shape of $\alpha(t)$ is represented in Figure 3.12.

In the next phase of the algorithm, the weighting factors $w_l^{\mathcal{M}}$ are determined. We have already determined $L^{\mathcal{M}}$, and a motion sequence reconstructed using $L^{\mathcal{M}}$ layers of the hierarchical B-spline always satisfies the kinematic constraints. Thus, $w_l^{\mathcal{M}}$ for these layers should be 1. Then we update $L^{\mathcal{M}}$ as

$$L^{\mathcal{M}} \leftarrow L^{\mathcal{M}} + 1, \tag{3.33}$$

and try to determine the weighting factor for $L^{\mathcal{M}}$-th layer in order to reconstruct joint angle trajectories as closely as possible to the original joint angle trajectories. Through this step, there might also arise a discontinuity between neighboring motion segments because of the different weighting factors. To solve this problem, we re-apply the same interpolation technique described in Equation (3.30).

It might seem that there is a possibility that some keyposes are violated by this motion interpolation. However, the keypose constraint is mostly satisfied through our motion decomposition method, and this proves not to be a serious problem.

3.6 Experiments

3.6.1 Experimental Data

We tested our algorithm by modifying Aizu-bandaisan dance data performed at a normal musical speed. Each motion data is captured at 120 fps

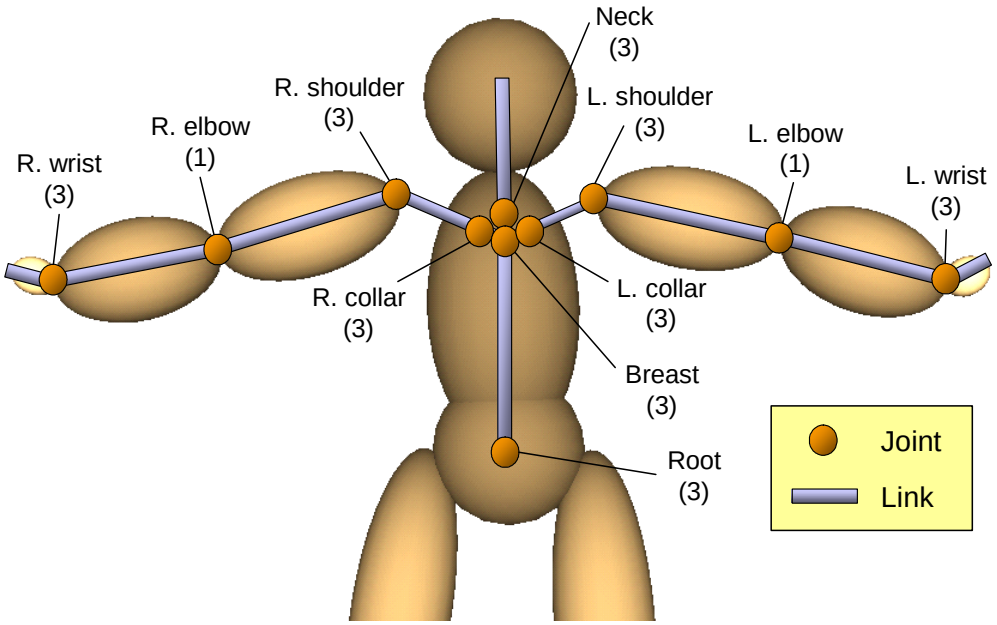


Figure 3.13: Degrees of freedom of the upper body joints.

by an optical motion capture system produced by Vicon. The format of the input motion data is VPM, in which each marker position is recorded, and we convert the position data into the joint angle. The degrees of freedom of each joint is shown in Figure 3.13.

For motion decomposition, we set the maximum number of hierarchical B-spline layers to 5.

3.6.2 Results of Motion Decomposition

We first show results of the motion decomposition method described in Section 3.5.1. Figure 3.14 (a) and Figure 3.14 (b) show comparisons of reconstructed dance keyposes using a single-layer B-spline and a two-layer hierarchical B-spline respectively. In these figures, top and bottom rows represent reconstructed keyposes and joint angular velocity trajectories, respectively. Green, and yellow lines represent motions reconstructed by our motion decomposition method (Equation (3.27)), and the traditional hierarchical B-spline fitting method

Joint	Dancer1 [degree / sec]	Dancer2 [degree / sec]
Shoulder	312.3	433.7
Elbow	244.7	304.8
Wrist	568.4	624.5

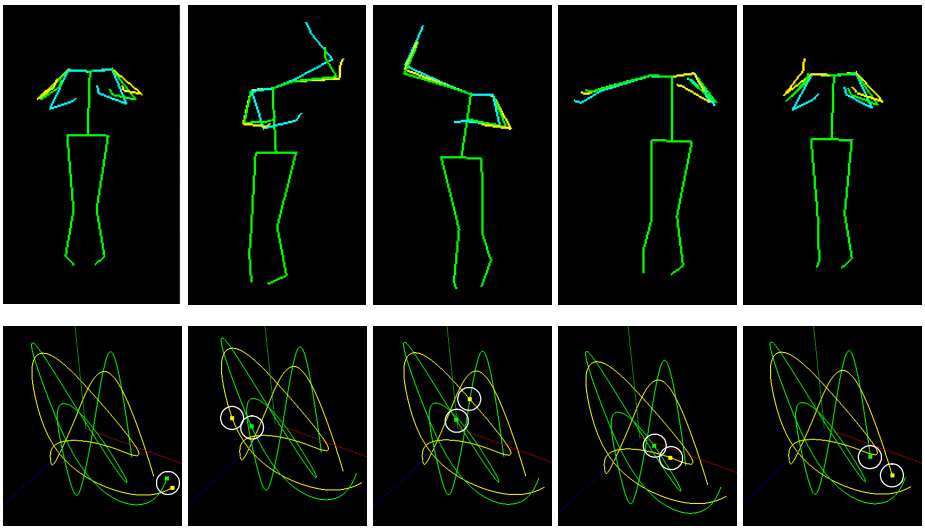
Table 3.1: Extracted joint angular speed limitations. The speed limitations of Dancer1 and Dancer2 are calculated from motion data performed at 1.5 times and 1.3 times faster musical playback speeds, respectively.

(Equation (3.13)) respectively, and the light blue articulated figure represents the original motion after noise removal. As shown, the keyposes reconstructed by our hierarchical B-spline method are more similar to the original ones than those reconstructed by the traditional hierarchical B-spline. With regard to joint angular velocity, ours are closer to zero than the results of the traditional methods. Altogether, these results validate the effectiveness of our proposed method.

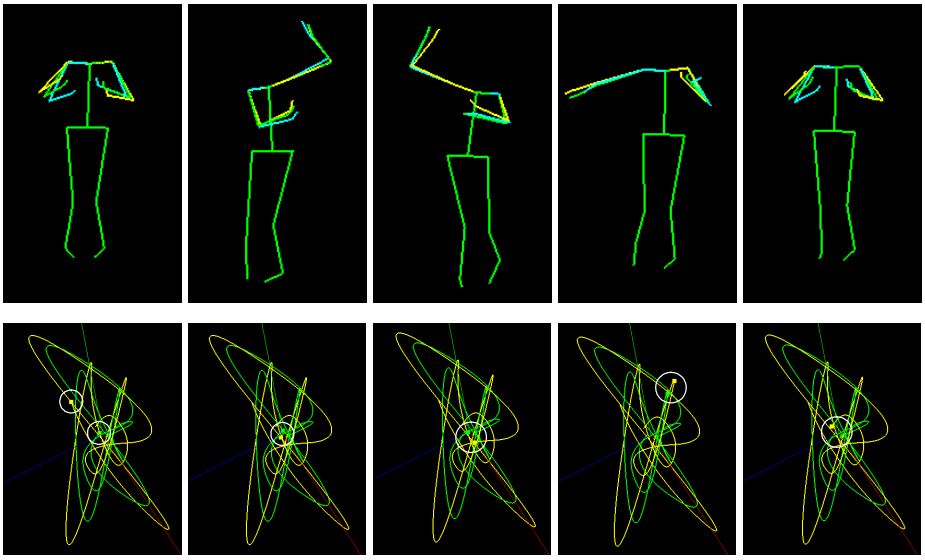
3.6.3 Results of Motion Modification

We considered only the maximum speed of each joint angle extracted from the original dance data performed at the faster musical playback speed as kinematic constraints. The speed limitation extraction was done after smoothing the motion data in order to reduce noise. The extracted speed limitations are shown in Table 3.1.

For comparison, we also synthesized motion from the motion capture data at a normal musical playback speed by applying temporal scaling of the motion and then a scaling of joint angle magnitude in order to satisfy the extracted speed limitations. In the following, we denote motion synthesized by our method, original motion performed at the faster musical playback speed, and motion synthesized via the simple scaling as *Synthesized Motion*, *Original Motion*, and *Scaled Motion*, respectively.



(a)



(b)

Figure 3.14: Results of motion decomposition using Dancer1's motion. (a): Result using a single-layer B-spline, and (b): result using a two-layer hierarchical B-spline. Top: reconstructed keyposes, and bottom: joint angular velocity trajectories of the left shoulder. Green and yellow lines represent motions reconstructed via our motion decomposition method and the traditional hierarchical B-spline fitting method, respectively. The light blue articulated figure represents the original motion after noise removal. Time markers at keyposes are shown inside white cycles inside the bottom figures.

Results of Motion Modification for Dancer1

First, we show the results of the experiment in which Dancer1's motion at 1.5 times faster musical playback speed was synthesized using the motion performed at the normal musical playback speed. The results are shown in Figure 3.15, in which green, yellow, and light blue lines represent the hand trajectories of the Synthesized Motion, Original Motion (performed at 1.5 times faster musical playback speed), and Scaled Motion in the body center coordinate system, respectively. Figure 3.16 shows a visualization of the layers and weighting factors for motion modification estimated in order to satisfy the given speed limitations. The high frequency components of the shoulder and wrist angles are more attenuated than those of elbows. This is because shoulders and wrists have more DOFs than elbows, and therefore, the movements of shoulders and wrists contain more complex motion than those of elbows.

Figure 3.17 shows the shoulder angle trajectories of Dancer1's motion. Figure 3.18 represents the frame-by-frame distance of the hand position in the body center coordinate system between the Original Motion and the Synthesized/Scaled Motion. From these results, it is confirmed that the trajectories of the Synthesized Motion are much closer to those of the Original Motion than those of the Scaled Motion.

Results of Motion Modification for Dancer2

Dancer2's motion at 1.3 times faster musical playback speed was synthesized using the motion performed at the normal musical playback speed.

The results are shown in Figure 3.19, in which green, yellow, and light blue lines represent the hand trajectories of the Synthesized Motion, Original Motion (performed at 1.3 times faster musical playback speed), and Scaled Motion in the body center coordinate system, respectively. Figure 3.21 shows the shoulder angle trajectories of Dancer2's motion. Figure 3.22 represents the frame-by-frame distance of the hand position in the body center coordinate system between the Original Motion and the Synthesized/Scaled Motion. It is confirmed that the trajectories of the Synthesized Motion are much closer to those of the Original Motion than those of the Scaled Motion.

The differences between the Original Motion and Synthesized Motion of Dancer2 is much larger than those of Dancer1. This is because Dancer2's hand motions are more complex than Dancer1's. Figure 3.20 and Figure 3.16 also enable us to conclude that when body parts have complex movements, their hierarchical B-spline layers are greatly attenuated. This is derived from the fact

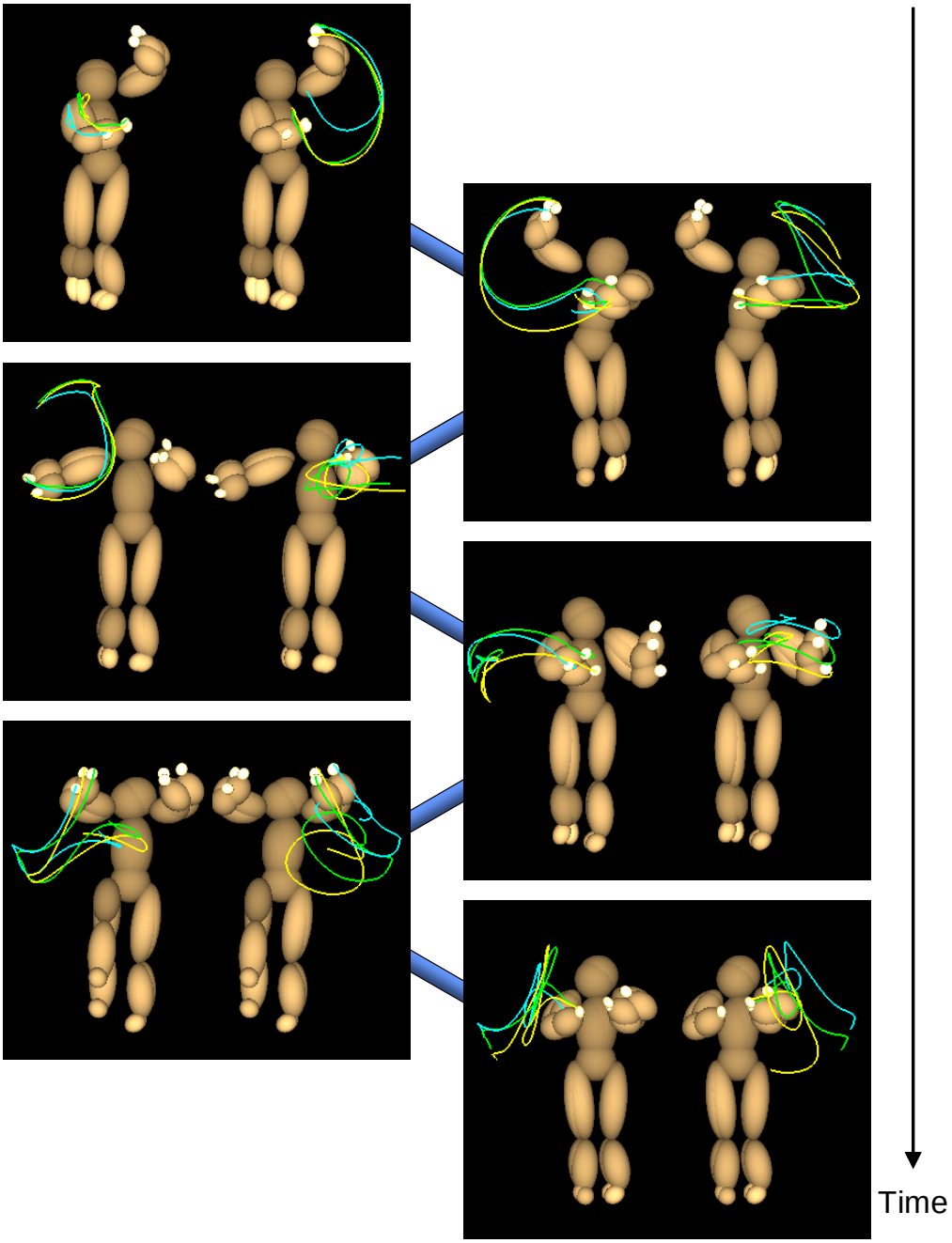


Figure 3.15: Result of the motion modification method using motion performed by Dancer1. Figures on the left side and right side show the right and left hand trajectories of the Synthesized Motion (green), Original Motion (yellow line) and Scaled Motion (light blue line).

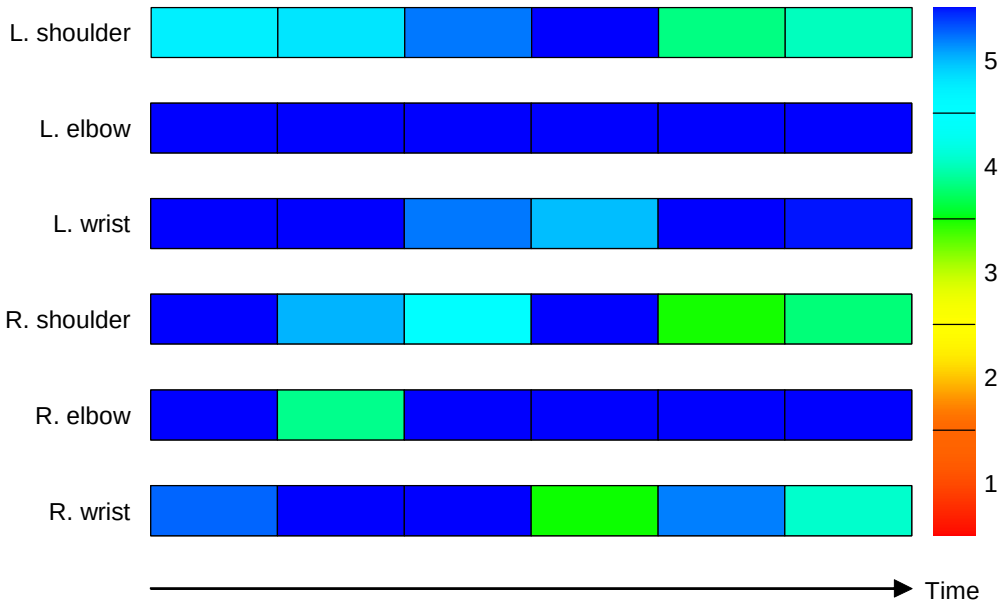


Figure 3.16: Layers and weighting factors for motion modification of Dancer1.

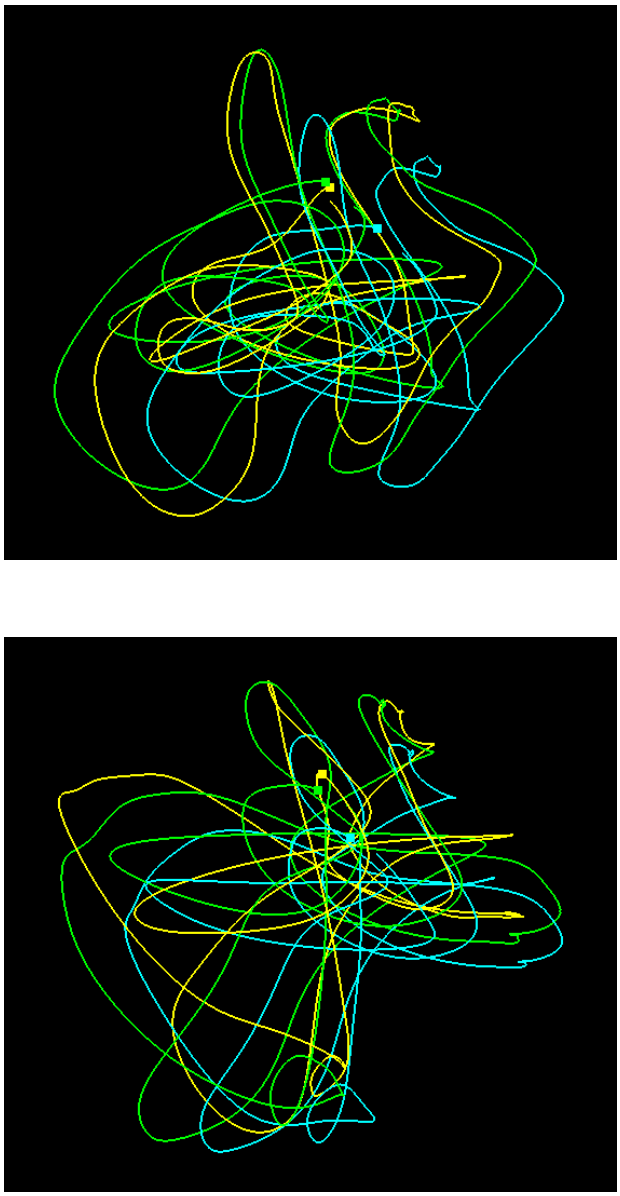


Figure 3.17: Result of the modified shoulder angle trajectories of Dancer1. Top: left shoulder angle trajectories, and bottom: right shoulder angle trajectories. Green, yellow, and light blue lines represent the shoulder angle trajectories of the Synthesized Motion, Original motion, and Scaled Motion, respectively.

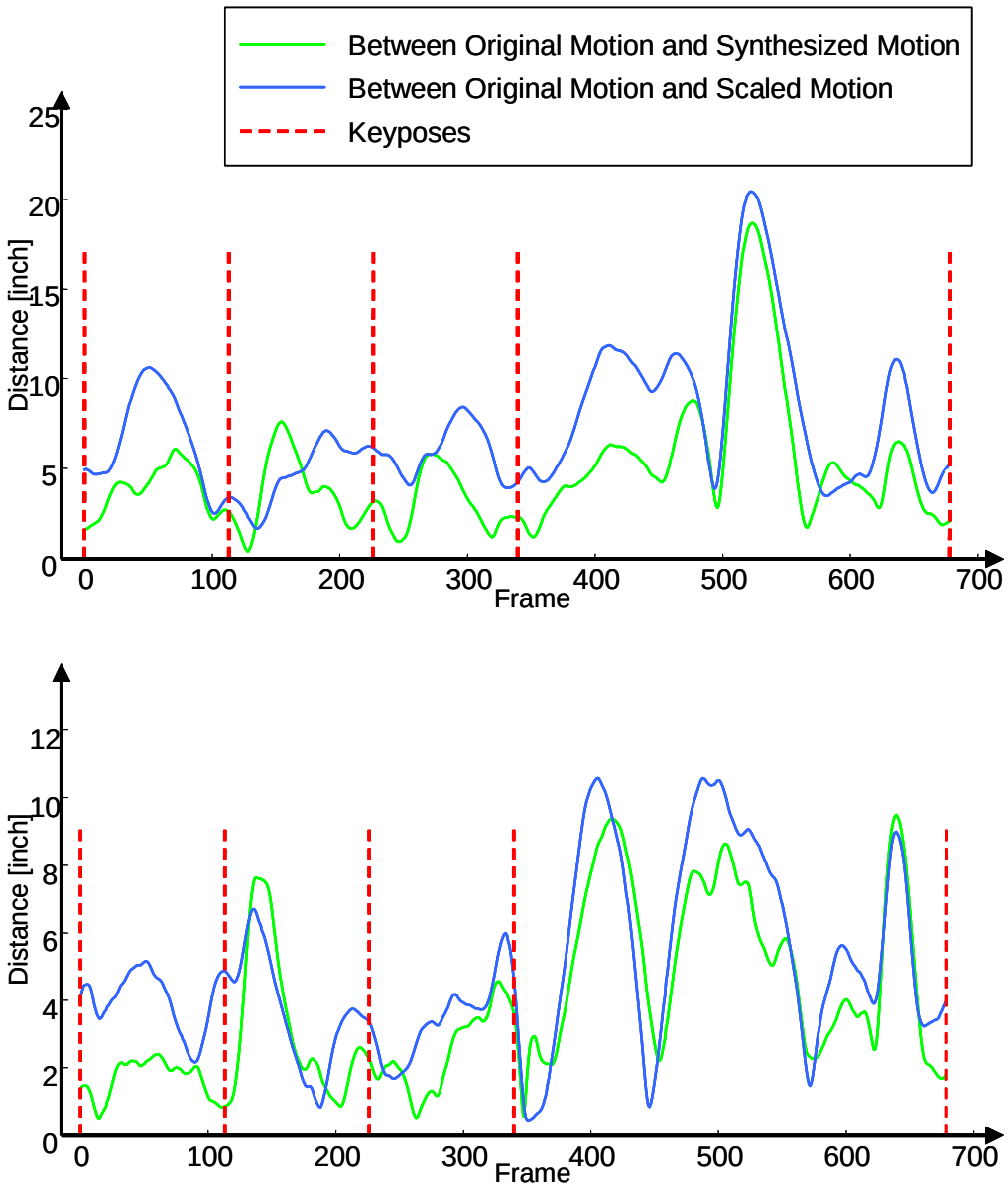


Figure 3.18: Frame-by-frame distance of hand position in the body center coordinate system of Dancer1. Top: difference of the left hand position, and bottom: difference of the right hand position. The green and blue lines show the difference between the Original Motion and the Synthesized Motion, and the difference between the Original Motion and the Scaled Motion. The vertical red broken lines represent keypose frames.

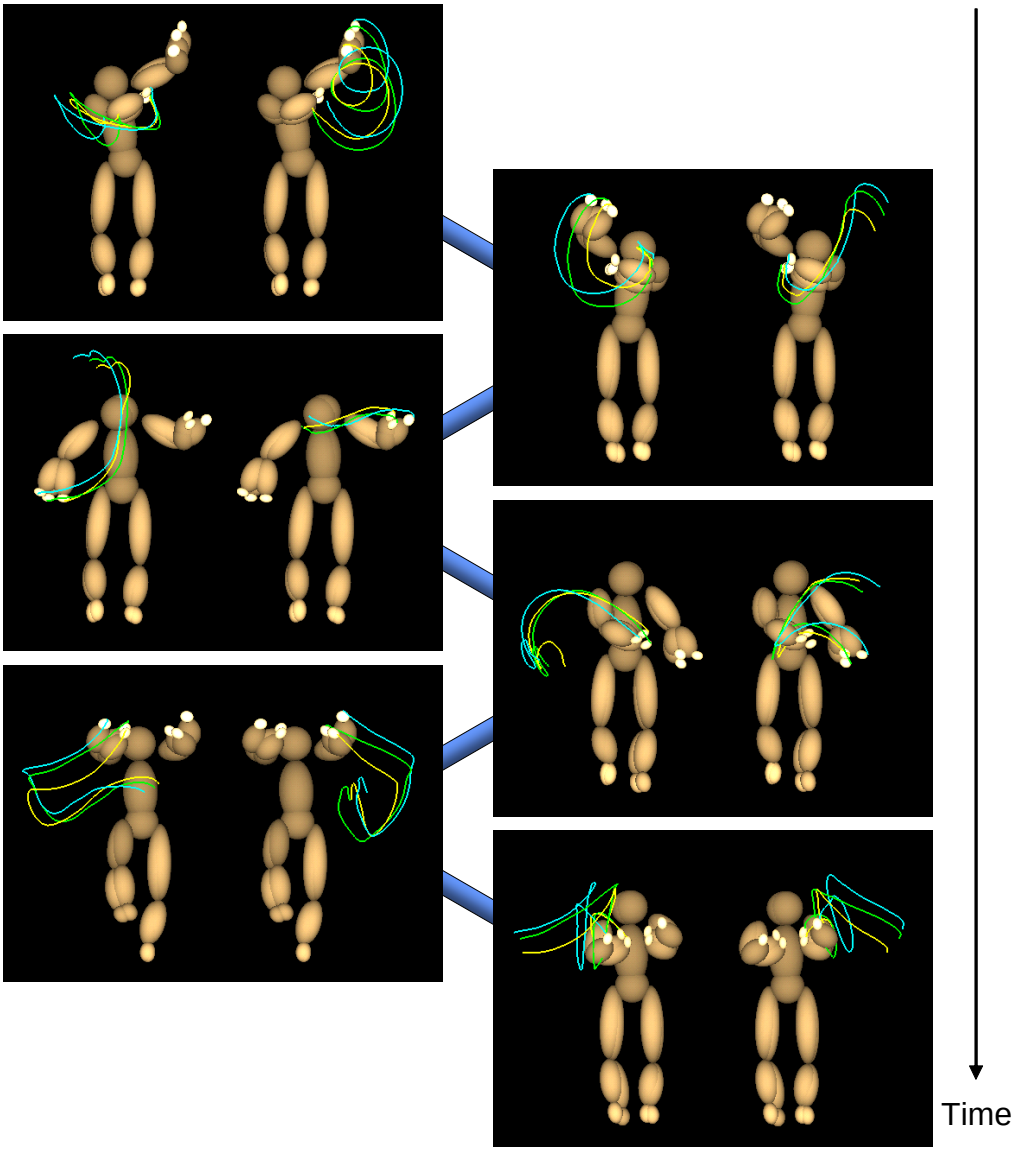


Figure 3.19: Result of the motion modification method using motion performed by Dancer2. Figures on the left side and right side show the right and left hand trajectories of the Synthesized Motion (green), the Original Motion (yellow line) and the Scaled Motion (light blue line).

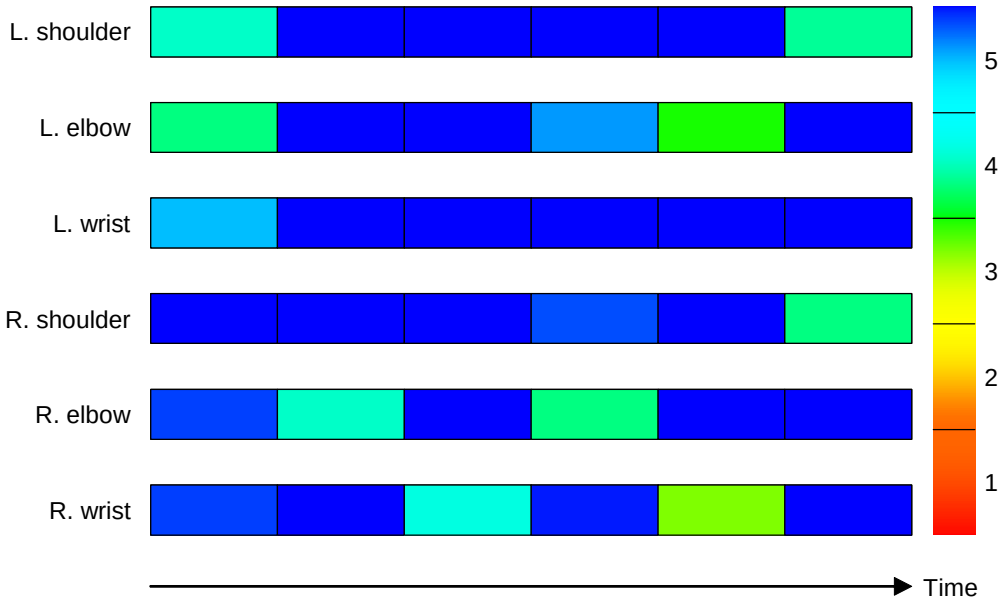


Figure 3.20: Layers and weighting factors for motion modification of Dancer2. Each horizontal block represents a motion segment based on the estimated musical rhythm.

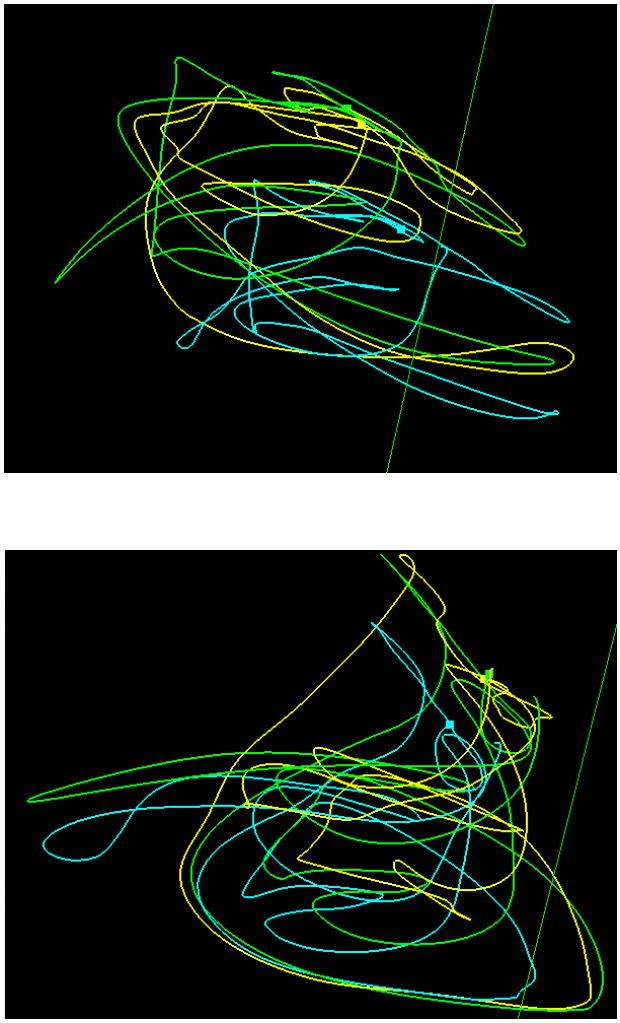


Figure 3.21: Result of modified shoulder angle trajectories of Dancer2. Top: left shoulder angle trajectories, and bottom: right shoulder angle trajectories. Green, yellow, and light blue lines represent the shoulder angle trajectories of the Synthesized Motion, Original motion, and Scaled Motion, respectively.

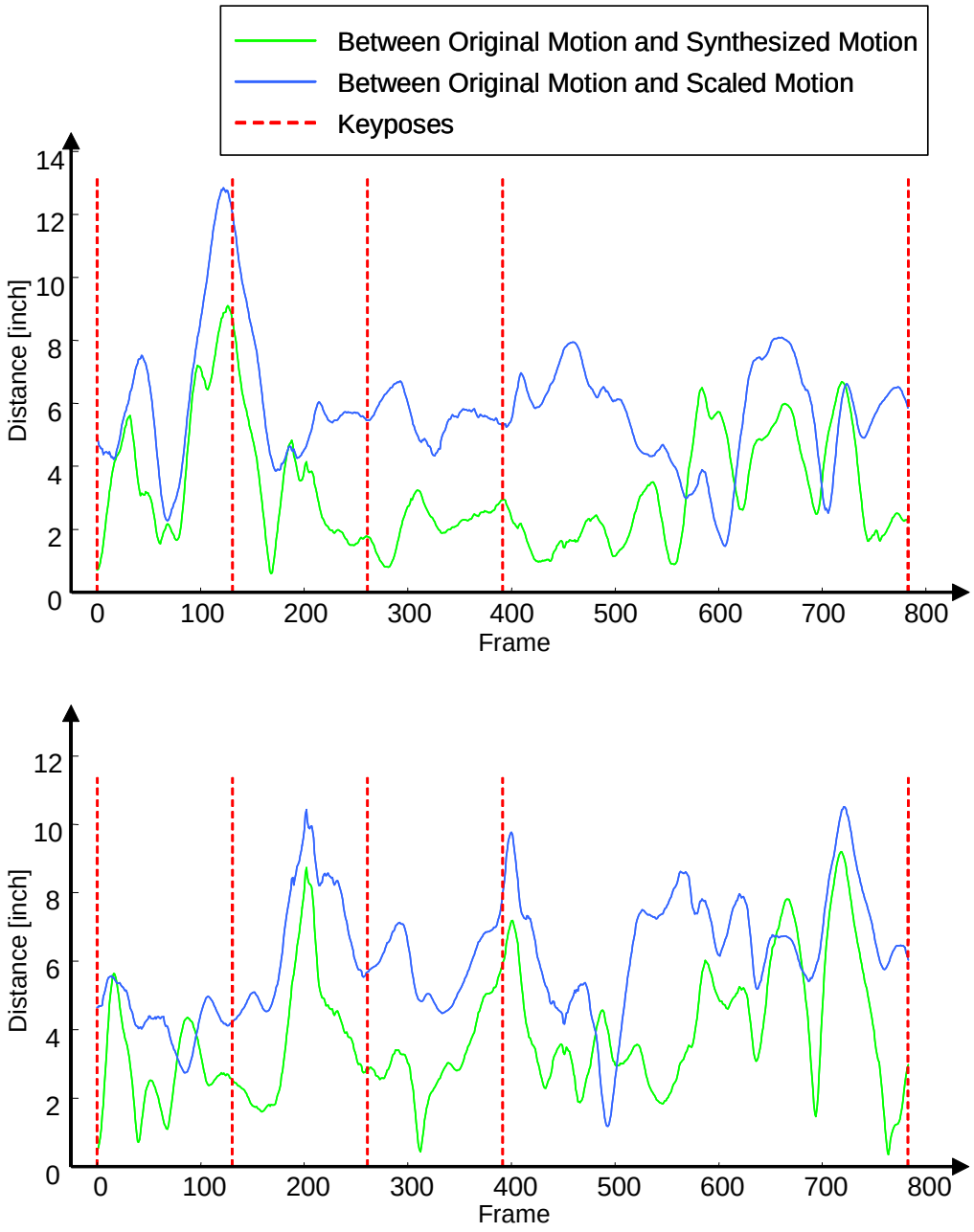


Figure 3.22: Frame-by-frame distance of hand position in the body center coordinate system of Dancer2. Top: difference of the left hand position, and bottom: difference of the right hand position. The green and blue lines show the difference between the Original Motion and the Synthesized Motion, and the difference between the Original Motion and the Scaled Motion. The vertical red broken lines represent keypose frames.

that complex trajectories make it difficult for a dancer to follow a faster music speed. Therefore, our method works more effectively in the case of complex motion trajectories.

3.6.4 Application for Humanoid Robot Motion Generation

Because a humanoid robot has kinematic constraints, humanoid robot motion generation is a suggested applications of our method. Here, we show simulation results which apply the dance motions of Dancer1 and Dancer2 performed at the normal musical speed to a biped humanoid robot. To evaluate our method, we compare our method with Pollard *et al.*'s method [PHRA02], which modifies joint angle and angular speed trajectories using a PD filter to satisfy kinematic constraints. For lower body motion, we apply Nakaoka *et al.*'s method [NNK*05], which analyzes lower body motion, classifies the motion into predefined four states: STAND, SQUAT, L-STEP, and R-STEP, and generates balance-maintained lower body motion based on this classification.

Experimental Platform

Our experimental platform is an *HRP-2* developed by Kaneko *et al.* [KKK*02]. The HRP-2 consists of a whole body with 30-DOF joints; we are focusing on the 12 DOFs of HRP-2's arms as shown in Figure 3.23. The height and weight of the HRP-2 are 1.54 [m] and 54 [kg], which is quite similar to those of humans. We use these limitations as kinematic constraints.

Simulation Results for Dancer1's Motion

Figure 3.24 shows the simulation results for Dancer1's motion. The red sphere represents the Zero Moment Point (ZMP) proposed by Vukobratović *et al.* [VJ69, VBSS90]. If there is a ZMP inside supporting area, a humanoid robot will maintain its balance; otherwise the robot will fall down. Our simulated motion satisfies this criterion for balance maintenance, and the humanoid robot successfully performed the dance. Figure 3.25 illustrates the hierarchical B-spline layers and weighting factors used to modify the original joint trajectories. Figure 3.26 and Figure 3.27 show the angle and angular speed trajectories of the left shoulder roll and the left wrist pitch joint, respectively. In these figures, red, green and blue solid lines denote the original trajectory, the trajectory modified

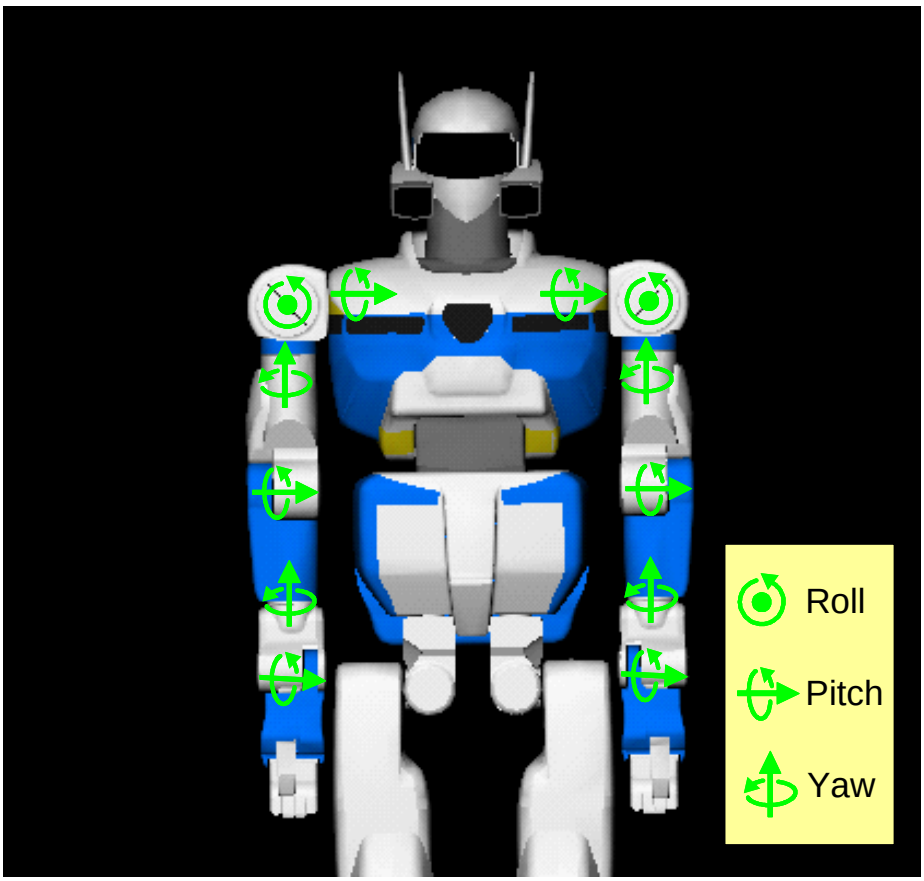


Figure 3.23: Our humanoid robot experimental platform: HRP-2. Each arm of the HRP-2 consists of 6-DOF joints.

by Pollard *et al.*'s method, and the trajectory modified by our method, respectively. Gray broken lines denote the maximum possible angle and angular speed limitations of the HRP-2.

It is easily confirmed that both methods can modify the angle and angular trajectories in order to satisfy kinematic constraints. In the case of left shoulder roll, the original angle and angular speed trajectories mostly satisfy the kinematic constraints, and do not need to be modified. While our method can re-generate motion quite similar to the original one, the motion modified by Pollard *et al.*'s method lacks high frequency components. This is because the original motion was smoothed via PD control.

In the case of left wrist pitch, the original angular speed trajectory sometimes violates the kinematic constraints. The trajectories resulting from our method lack high frequency components around constraint-violating motion frames, but keep their high frequency components in other frames. On the other hand, the trajectories resulting from Pollard *et al.*'s method always lack high frequency components, due to the PD control. Note, also, in Pollard *et al.*'s method, that the speed around constraint-violating motion frames is a constant value. This can create two problems. One is that the humanoid robot cannot clearly reproduce a keypose if the posture and angular speed around the keypose violate kinematic constraints. The other is that the humanoid robot may fall because of the rapid changes in acceleration. Therefore, our method works better for humanoid robot motion generation than Pollard *et al.*'s method.

Simulation Results for Dancer2's Motion

Figure 3.28 shows other simulation results with Dancer2's motion. This simulated motion also satisfies balance maintenance requirements. Figure 3.29 illustrates the layers and weighting factors used to modify the original motion of Dancer2. It is easily confirmed that Dancer1's modified motion is quite different from Dancer2's modified motion, and that essential characteristics are preserved despite high frequency component attenuation.

3.7 Discussion

Keyposes

Through the observations of human motion analyzed using a hierarchical B-spline, we obtained the insight that keyposes are preserved even when the music

playback speed is faster. This is shown by the fact that variances calculated from Equation (3.20) around the keyposes are locally minimal.

Noting that Kang *et al*. [KI93] and Ogawara [Oga01] proposed a method to extract the primitive characteristics of motion by detecting the local minima of motion variance sequences, we can also validate definition of keyposes that instants where motion variance gets locally minimal are likely to be keyposes. In turn, the existence of keyposes depends upon the skill of dance performers. Indeed, keyposes do not appear clearly in motion sequences performed by untrained dancers, while motions performed by Dancer1 and Dancer2, who are masters of the Aizu-bandaisan dance, present very clear keyposes. Therefore, we believe that variance around the keyposes enables us to recognize whether a performer is well-skilled or not.

Applicability to Lower Body Motion

Generally, upper body motion performed by a single dancer is not constrained by the environment, such as contact with objects. In contrast, lower body motion is often constrained. For example, lower body parts, especially feet, experience impacts with the floor. When feet touch the ground, their high frequency components are suddenly much larger [Ari06]. But these high frequency components are not derived from a performer's style. The violation of high frequency components in synthesized foot motion produces an unnaturalness called *foot-skating*. Nakaoka *et al*. [NNK*05] proposed a method to recognize the states of lower body motion and to extract the style components of lower body motion. We believe that our method is applicable to lower body motion if Nakaoka *et al*.'s method is applied so that possible states "swing sole" for a leg are properly recognized.

3.8 Summary

In this chapter, we proposed a method to modify upper body motion considering aspects of human motion and kinematic constraints. We analyzed motion data captured at varying musical speeds by using a hierarchical motion decomposition technique. Through this observation, we obtained the following two insights:

1. High frequency components of human motion should be attenuated when music playback speed is too fast to follow.

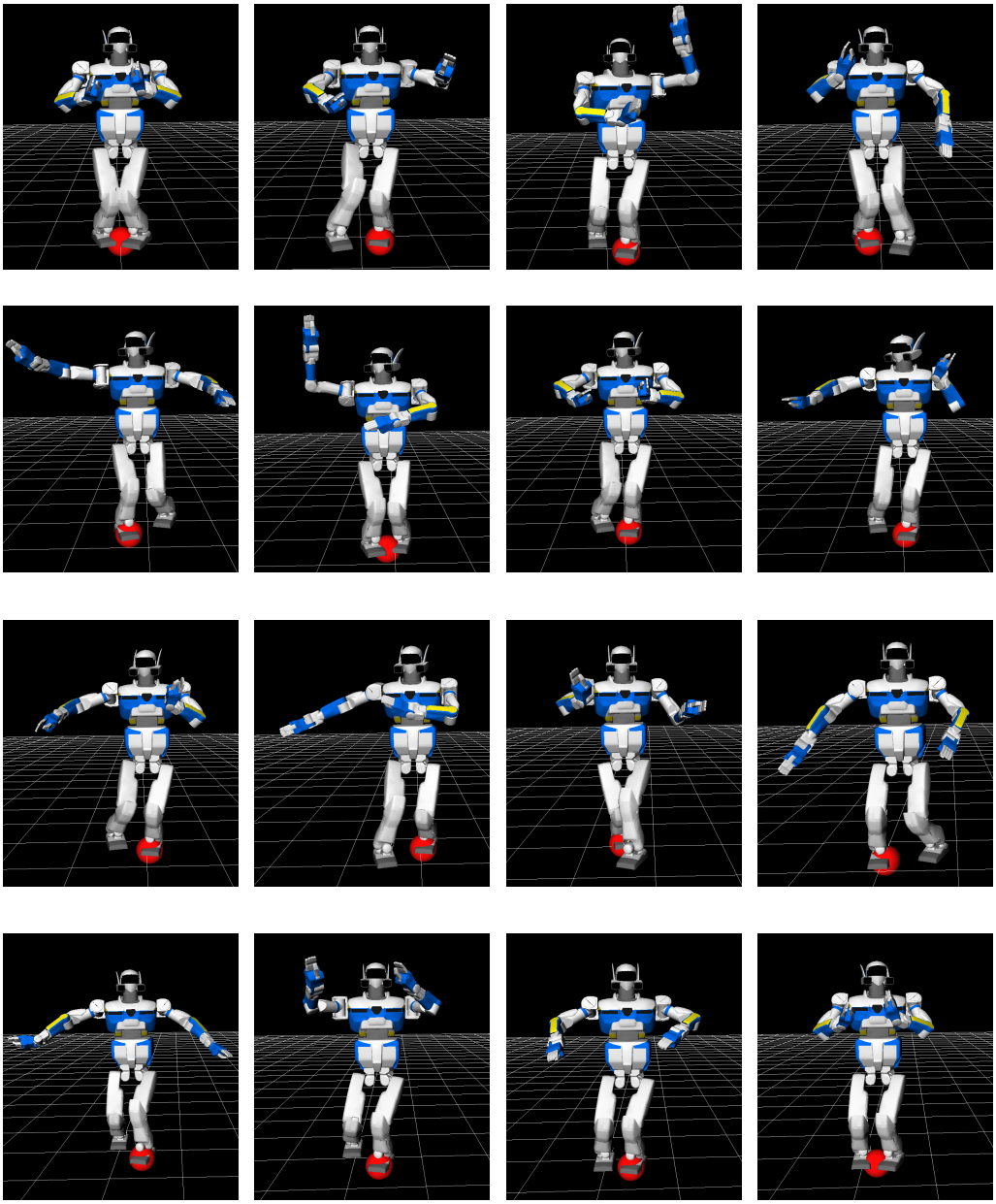


Figure 3.24: Simulation result for Dancer1's motion. Red sphere represents the ZMP of the generated motion.

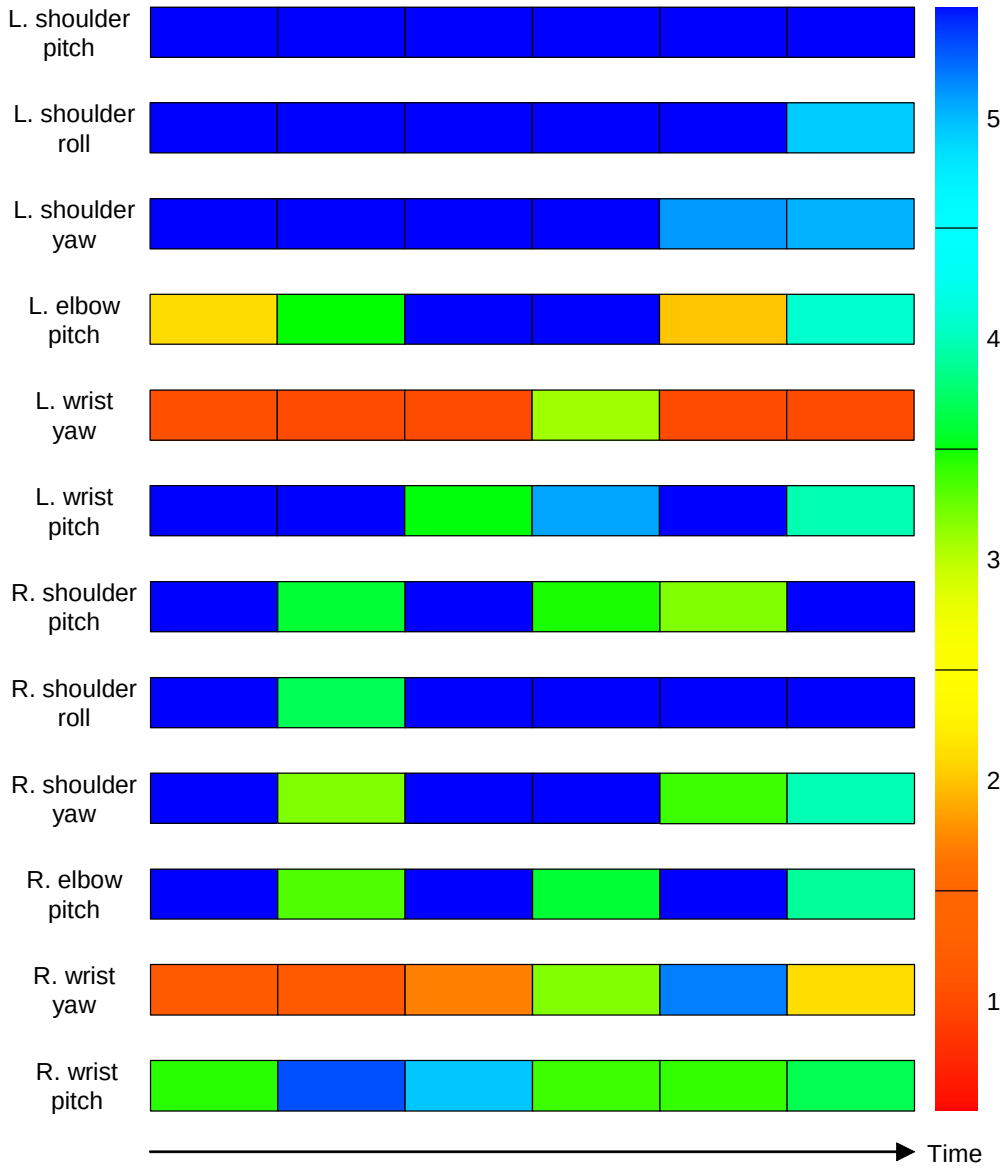


Figure 3.25: Layers and weighting factors used to generate humanoid robot motion from Dancer1's motion.

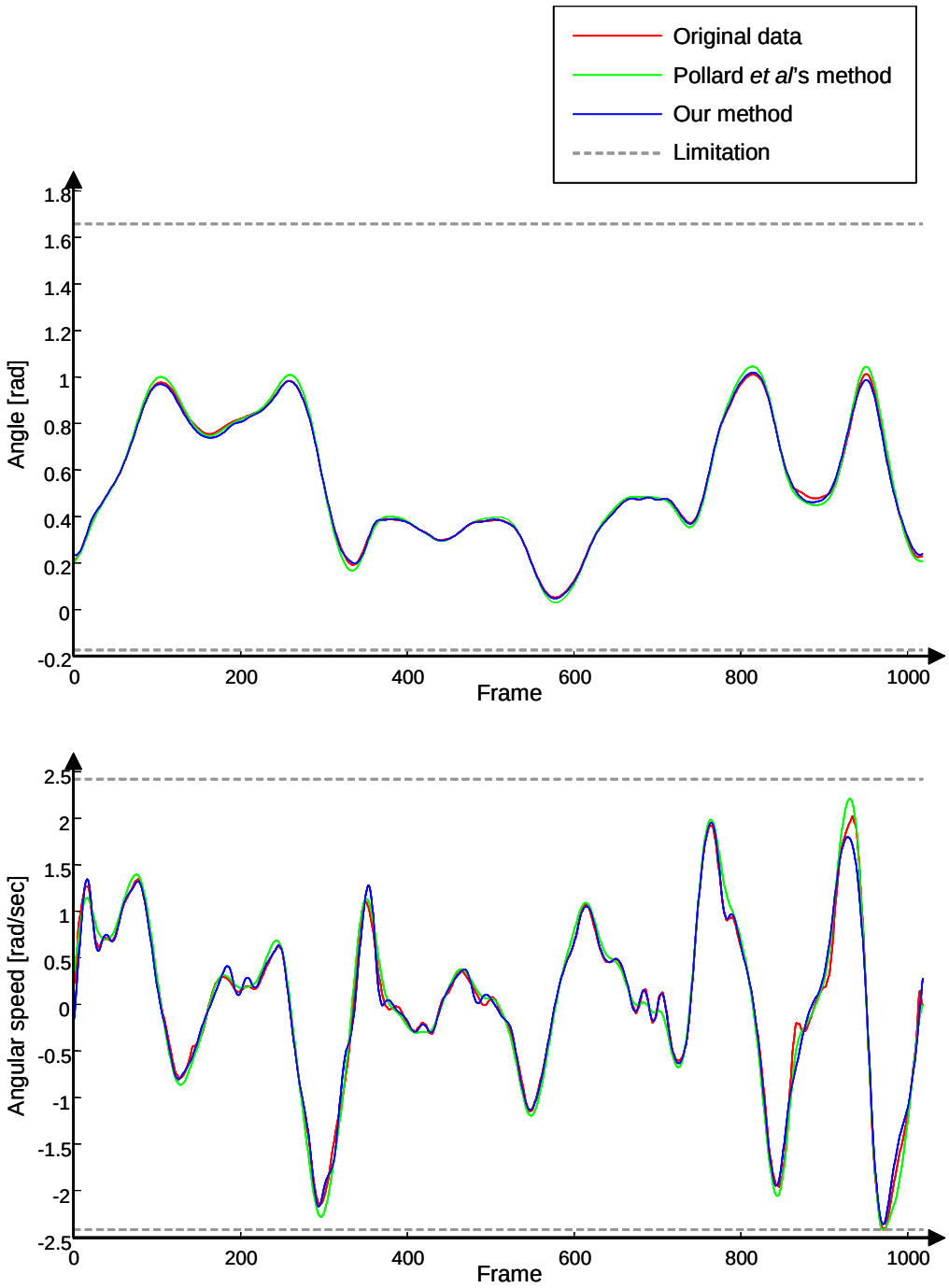


Figure 3.26: Comparisons of angle (top) and angular speed (bottom) of left shoulder roll. Red, green, and blue solid lines represent the trajectories of the original motion, the result of Pollard *et al.*'s method, and the result of our method, respectively. Horizontal broken lines denote the limitations of the HRP-2.

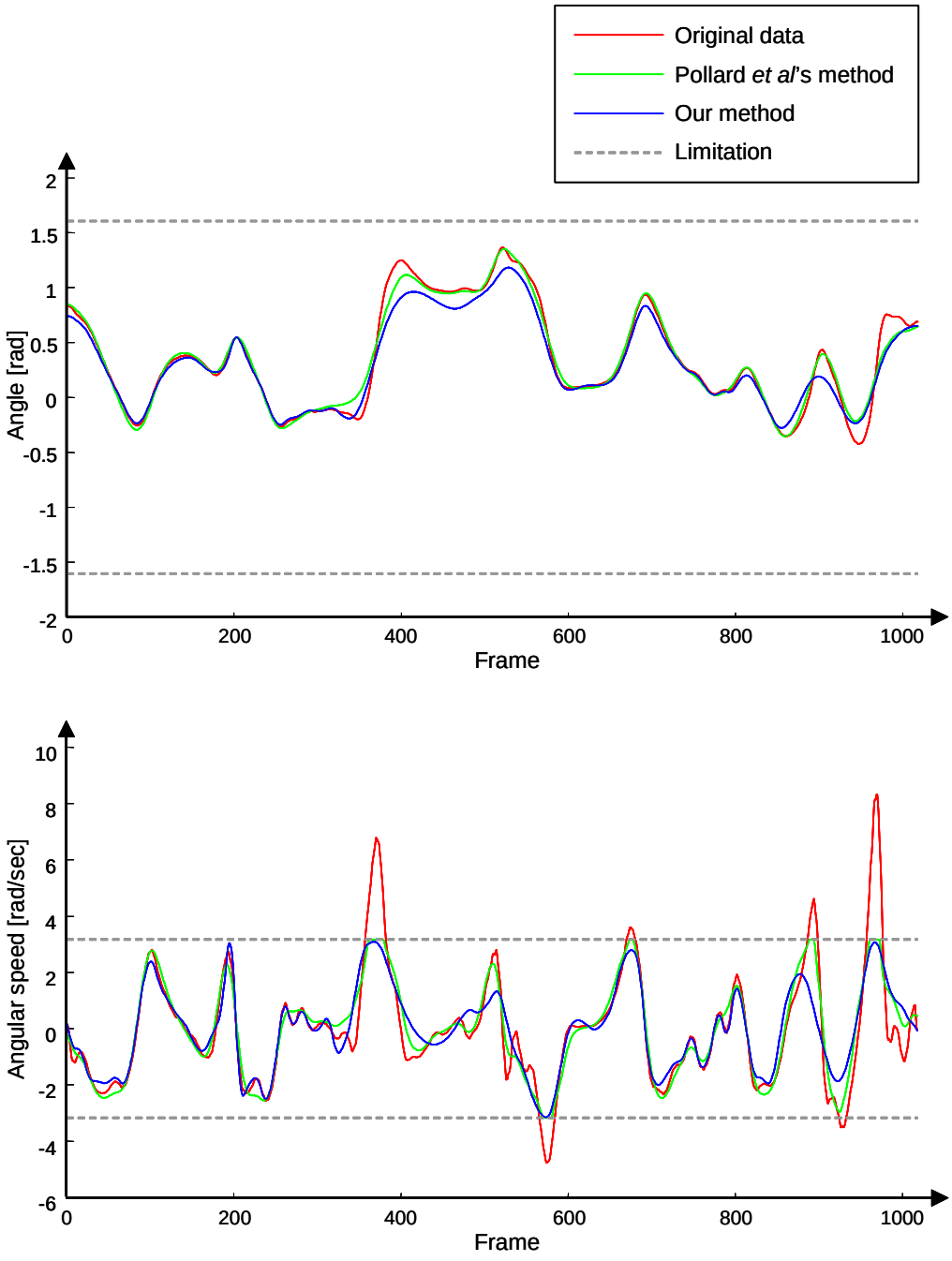


Figure 3.27: Comparisons of angle (top) and angular speed (bottom) of left wrist pitch. Red, green, and blue solid lines represent the trajectories of the original motion, the result of Pollard *et al.*'s method, and the result of our method, respectively. Horizontal broken lines denote the limitations of the HRP-2.

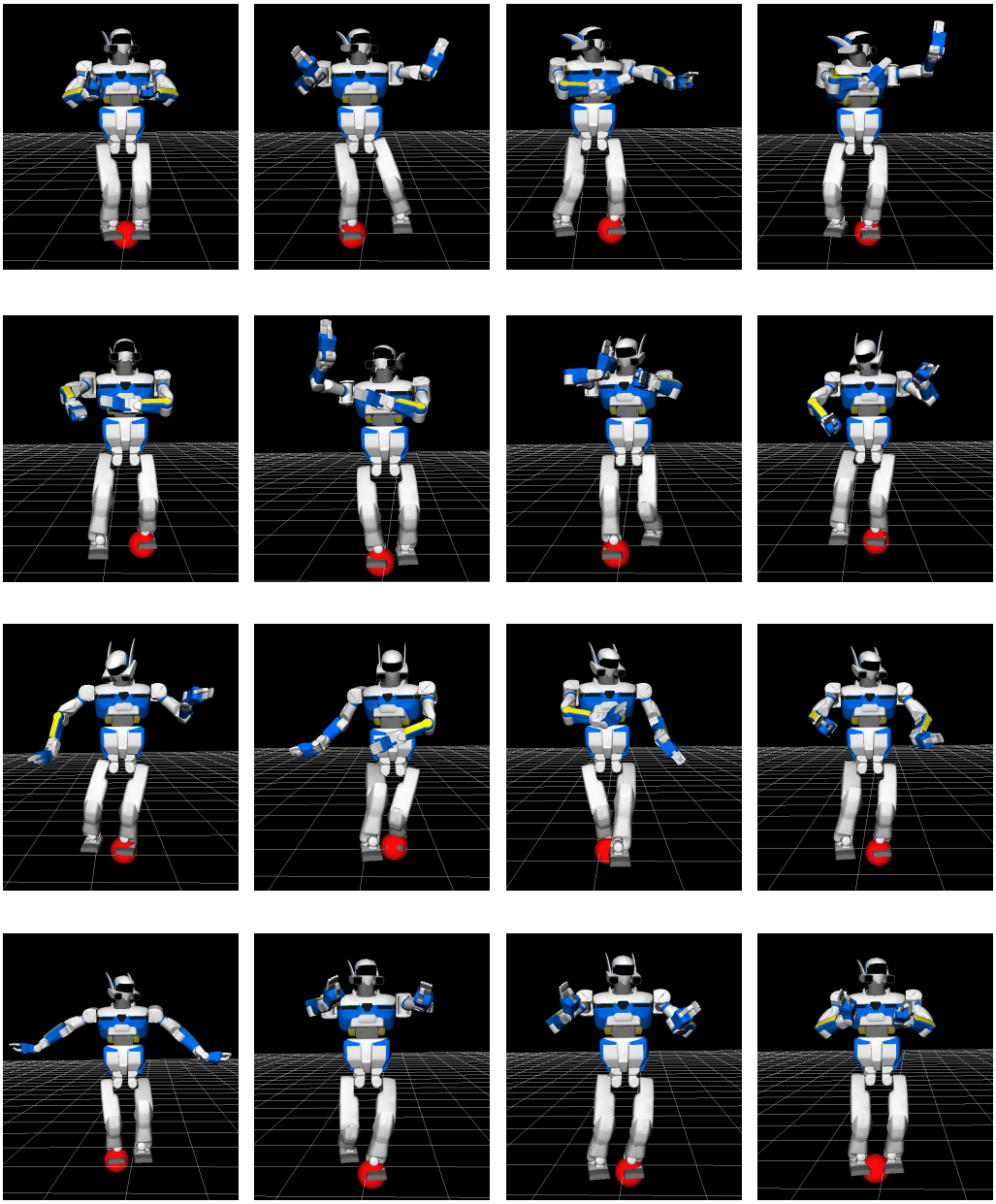


Figure 3.28: Simulation result for Dancer2's motion. Red sphere represents the ZMP of the generated motion.

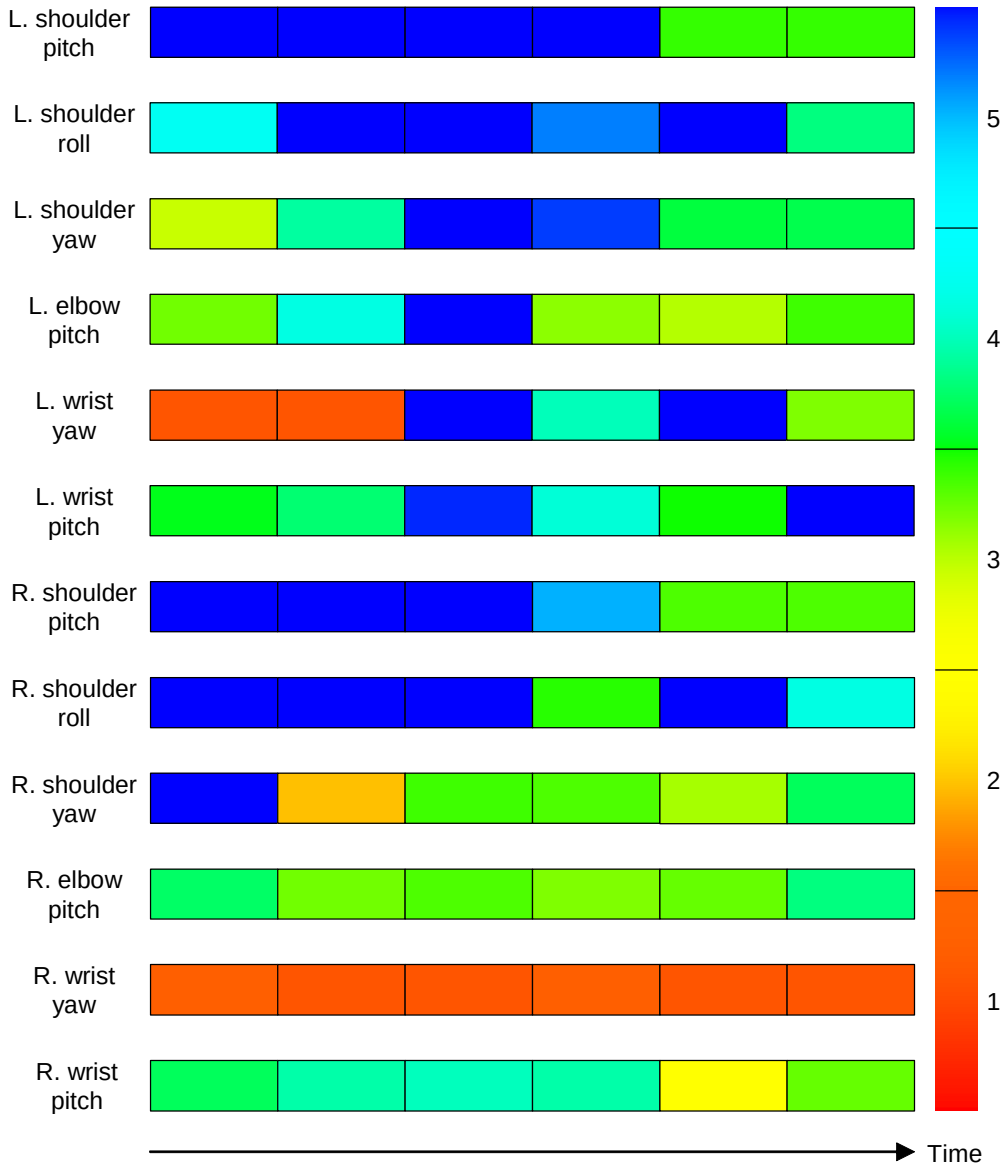


Figure 3.29: Layers and weighting factors used to generate humanoid robot motion from Dancer2's motion.

2. Keyposes will be preserved even when high frequency components are attenuated.

We applied these insights to model motion modification which can generate motion satisfying specified kinematic constraints. Our experimental results show the effectiveness of our method. We also show an application of this method in humanoid robot motion generation.

Chapter 4

Dancing-to-Music Character Animation Based on Aspects of Human Emotion

4.1 Introduction

Synthesizing realistic human motion is currently one of the most important topics in computer graphics research. Most motion synthesis techniques use motion capture data and synthesize new motion which is synchronized with external input signals such as trajectories designed by users [KGP02], environmental obstacles [LCR*02], speech information [SDO*04], motion of another character [HGP04], and so on. The main issue surrounding these techniques is the nature of the cues used to search and distinguish appropriate motions from the typically large amount of data in a motion database. Animators need to choose suitable cues in order to create the motion sequences they really want. So, in this chapter, we propose a novel approach for synthesizing *expressive* dance motion matched to music. Our approach uses music signals as a cue to synthesize new dance motion. The goal of this approach is the realization of a dance algorithm that mimics human motions based on emotional aspects.

The ability to dance to music is a natural skill for a human. Everyone has experienced a desire to move their bodies while listening to a rhythmic song. Some dances are formal and specified a priori, but this is not necessary: hip-hop dancers can immediately compose a dance motion corresponding to the musical sounds they are hearing. Although this ability may appear amazing; actually, these performers do not create these motions, but instead combine appropriate pre-existing motion segments from their personal knowledge database with mu-

sic. Considering this ability, we are led to believe that dance motion has strong connections with music in the two following aspects:

- The rhythm of dance motions is synchronized to that of music.
- The intensity of dance motions is synchronized to that of music.

The first assumption is derived from the fact that almost all people can recognize the rhythm of music; they can clap or wave their hands, and dance to music. The second assumption is derived from the fact that people feel quiet and relaxed when listening to relaxing music such as a ballad, and they feel excited when listening to intense music such as hard rock.

Our approach consists of three steps: motion analysis, music analysis, and motion synthesis based on the extracted features.

In the motion analysis step, we analyze the rhythm and intensity features of input dance motions, and assign these features to each motion in a database. Our analysis method depends on recent studies regarding the emotional aspects of human motions. Using these features, our system finds a sequence of motion segments matched to the input music sequence with respect to the rhythm and the intensity of the music.

In the music analysis step, first, we analyze the structure of an input music sequence, and extract music segments based on the structure analysis results. Second, musical rhythm and intensity features are extracted, and are assigned to each music segment.

In the motion synthesis step, our method automatically synthesizes new dance motion by interpolating between the motion segments. Additionally, our system has a user interface that enables animators to control the synthesis process by choosing the motion segments best matched to their intentions during music segments. For example, animators can set key motions in the motion database for desired music segments, such as setting a jumping motion to the final scene of the song, or a punch motion to a particular sudden sound in the music.

4.2 Prior Work

In this section we introduce related work on data-driven character animation and auditory scene analysis, both of which are very important for our approach.

4.2.1 Data-driven Character Animation

The proposed method is based on motion synthesis using a data-driven animation synthesis technique. Data-driven character animation is mainly derived from two domains: synthesizing motion by editing motion data, and synthesizing motion by extracting segments from a motion database.

Animation Synthesis by Editing

A typical approach to editing motion data is to apply a signal processing method to motion capture data. Bruderlin *et al*. [BW95] proposed methods to edit motion data using signal processing techniques such as a filter bank or dynamic programming for blending motions. Witkin *et al*. [WP95] tried to modify motion data by warping the end-effectors' trajectories with displacement maps approximated by B-spline curves, and this technique is called the *motion-warping* method. Wang etal [WDAC06] presented a new filtering technique called the *cartoon animation filter* that could make an input motion sequence appear to be more animated. This method could be applied to not only motion capture data, but also to layered video sequences.

Another common way to edit motion is *spacetime constraints*, a method of generating motion under the constraint that a specified body part is in a specified position at a specified time. This method was proposed by Witkin *et al*. [WK88]. The technique was designed for a general articulated object including a human figure; it calculates the required external forces and joint torques so that an object can exist in a specified position at a specified time. This technique was extended for the *retargeting* problem, in which motion capture data are transferred to new characters while retaining important constraints. A simple approach to retargeting involves traditional inverse kinematics such as the calculation of a Jacobian, but unfortunately, temporal consistency may be violated; i.e., an end-effector may be required to move impossibly fast in order to satisfy all the motion constraints. Gleicher [Gle98] solved this problem with an optimization based on the motion-warping method. Lee *et al*. [LS99] improved this method by parameterizing motion with a hierarchical B-spline in order to reduce computational cost. In these methods, constraints such as contact between the character's feet and the ground must be specified by users. Shin *et al*. [SLSG01] presented a real-time retargeting method in which the constraints were automatically analyzed via their proposed *importance analysis*; the retargeting issue was then addressed via their quick inverse kinematics solver.

Animation Synthesis from Motion Database

Recently, many researchers are focusing on the advantages of using a motion database. One typical method of using a motion database is a *motion graph* algorithm, in which motion capture data segments are connected to each other. This method also includes the idea of tracing a motion graph to satisfy users' inputs; path or environmental obstacles can thereby be used to force the synthesis of new motions [KGP02, AF02, LCR*02, LWS02, AFO03, SO06, HG06]. Lai *et al*. [LCF05] extended this algorithm to synthesize crowd animation. Lee *et al*. [LCL06] developed a method to annotate environments with motion capture data, and to thereby synthesize new motion by integrating the annotated small environments with interpolated transition motion between the environments. Reitsma *et al*. [RP04] proposed a method to evaluate whether or not input character's path can be satisfied through constructed motion graphs.

The standard problem of spacetime constraints can also be solved using multiple motion capture data sets. Wiley *et al*. [WH97] proposed a method to find an optimal set for motion capture data whose end-effectors' positions are near to positions specified by users, and to thereafter linearly interpolate the motion capture data's positions to satisfy the specified positions. Rose *et al*. [RBC98] considered motion data to be analogous to *verbs* and important features of a motion data set to be *adverbs*. By adjusting their "adverbs," new motion adequately satisfying spacetime constraints was generated. They also developed a method to interpolate motion sequences using a radial basis function [RSC01]. Kovar *et al*. [KG04] extended these methods to select the best motion data set to be used for interpolation and to estimate the needed weighting parameters automatically and effectively. Mukai *et al*. [MK05] proposed an automated method to interpolate motion data using geostatistics.

Unfortunately, interpolated motions do not always realistically portray real human motion. Some researchers have focused on evaluating the degree of synthesized motion naturalness. Tak *et al*. [TSK00] presented a method to re-create a balance-maintained motion by calculating and adjusting a *zero moment point* (ZMP); this ZMP can be used to test whether a desired motion can be executed without falling down [VJ69]. Reitsma *et al*. [RP03] paid attention to character's changes in horizontal and vertical velocity, and evaluated whether synthesized motion appeared natural. Safonova *et al*. [SH05] developed a method to analyze human physical correctness from the aspects of changes in momentum and angular momentum during flight, foot contact, balance maintenance, and friction. Ren *et al*. [RPE*05] proposed a method to evaluate human-motion naturalness using a motion capture database and stochastic models.

Motion Synthesis Considering Human Perceptual Models

There are several existing methods to synthesize human animation based on human perception which are very similar to our approach. Peters *et al*. [PO03] proposed a method of human animation synthesis based on a CG character's focus of visual attention. In their method, from the visual information, a CG character can "perceive" its surrounding environment and "plan" to perform a movement. Sakuma *et al*. [SMK05] proposed a psychological paradigm model for human crowd simulation in which neighboring CG characters impose mental stress each other, and therefore, they move in order to avoid too much closeness to each other.

Stone *et al*. [SDO*04] proposed a method whose approach is quite similar to ours in that input sound signals are considered. The goal of their method was to synthesize speech motion by extracting emphasis features of motion and speech data and synchronizing them; however, this method is not applicable to dance performance synthesis. Their feature extraction method requires many manual steps and is accordingly a very time-consuming system for synthesizing new speech motions. In contrast, our method can automatically extract motion and musical features and synthesize dance motion from musical input.

Kim *et al*. [KPS03] proposed a rhythmic motion synthesis method using the results of motion rhythm analysis. When using their method, music data input must have a rhythm interval that is similar to that of the resulting motion. It is quite difficult to apply this method with diverse music data. Alankus *et al*. [ABB05] and Lee *et al*. [LL05] also proposed a method to synthesize dance motion by considering the rhythm of input music. The drawback of both these methods is to consider only musical rhythm; because of this, it is very difficult to synthesize expressive dance motion.

4.2.2 Auditory Scene Analysis

Computational analysis methods for a music scene are important for understanding how humans recognize musical features; the topic is called *computational auditory scene analysis* [Bre90, CB93]. In this field, many researchers have focused on separating acoustic signals into each instrument's sound, and converting them into a musical score. Recently, however, their interests are shifting to how people recognize musical features in acoustic signals and apply such features to music signal processing [GH04].

One such method of music analysis is musical structure analysis. Most musical songs have repeating patterns and a prominent structure. Musical structure

analysis methods have been used to accomplish applications such as music summarization. In general, repeating patterns are considered as melody similarity. In order to extract the melodic similarity, musical intensity features are extracted from spectral components [LC00, WLZ04, SXWK04], or amplitude envelopes [LZ03] may be used.

Another extraction possibility involves mood analysis. People may feel a mood when listening to a song, such as sadness, happiness, and so on. However, few studies have touched this field. Katayose *et al*. [KII88] presented a sentiment extraction system for pop music from MIDI signals. Liu *et al*. [LLZ06] proposed a method to analyze the mood of classical music; their four possible moods are *exuberance*, *anxious*, *contentment*, or *depression*.

4.3 Approach

Our approach uses musical information as a cue to retrieve motion segments from a motion capture database. We start by discussing a human perception model based on the relationship between human motion and music. To define this music and motion relationship model, previous studies of human dance motion analysis are of great help.

Laban, who is famous for his novel dance description method called Labanotation, is a pioneer in the study of this topic. He has studied the emotional aspects of body movements [LU60]. According to his theory, the emotion of human motion comes from motion features consisting of *Effort* and *Shape* components. The Effort component is defined as the movements of body portions, and the Shape component is defined as the shape of elements he calls *keyposes*. Chi *et al*. [CCZB00] have developed a method to synthesize character animation based on these concepts. More recently, Nakata *et al*. [NMS02] have tested the validity of Laban's theory using a small robot and user studies. Although they could not find a significant relationship between the shape component and emotions, they found that the *Weight Effort* component, one of the Effort components, is closely related to the excitement of the motion. Laban defined the Weight Effort component as the strength of a movement, and Nakata considered them physically as the linear sum of rotation velocities of each body joint. We use these metrics to define the motion intensity component F_I^{Motion}.

As described in Chapter 2, we have developed a method that analyzes the relationship between stop motions and musical rhythm. The results indicate that musical rhythm has a strong connection with motion elements we call keyposes. Accordingly, our motion analysis method extracts the local minimums of the

Weight Effort component in order to extract the motion rhythm feature F_R^{Motion}. A motion feature vector for each frame is obtained via the motion feature analysis:

$$\textbf{MotionFeature}(f) = \begin{bmatrix} F_R^{\text{Motion}}(f) \\ F_I^{\text{Motion}}(f) \end{bmatrix}. \tag{4.1}$$

The next issue is to extract musical features. We believe that there are three important musical features for dance performance. One is *musical rhythm*. As everyone has experienced, there is a very close relationship between musical rhythm and motion rhythm. We consider musical principle about what is called "the onset component" to estimate musical rhythm F_R^{Music}. Another important factor is *music structure*, which consists of several musical phrases. Both music players and dancers try to keep the structure from being violated during their performances. We extract repeating patterns to detect the musical structure, and obtain music segments from the music sequence. The other important component is *music intensity*. People feel various emotions depending on the mood set in music, and the same is often true for dance. In music mood analysis, we mainly focus on music intensity as it is one of the more important factors in establishing musical mood. We extract the music intensity component F_I^{Music} using the energy of the melody line. Accordingly, a music feature vector for each music segment $\mathcal{M}$ is obtained:

$$\textbf{MusicFeature}(f; \mathcal{M}) = \begin{bmatrix} F_R^{\text{Music}}(f; \mathcal{M}) \\ F_I^{\text{Music}}(f; \mathcal{M}) \end{bmatrix}. \tag{4.2}$$

Our motion synthesis step extracts the most appropriate motion segment sequence by evaluating motion and music features. This step has two types of algorithms: a Motion Graph-based locally optimal algorithm, and a segment-based globally optimal algorithm. Both of them have a common approach in which first motion segment sequences are selected by matching rhythm features, and then the best motion sequence is selected by evaluating the similarity of intensity features.

4.4 Motion Feature Analysis

As described in Section 4.3, our motion analysis method strongly relies on Laban's Weight Effort component. In this section, we describe our definition of the Weight Effort component and how to extract motion features.

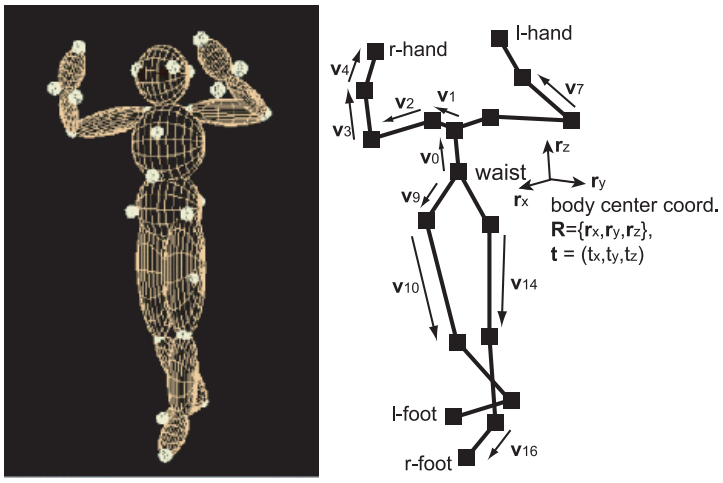


Figure 4.1: Our human body model for motion feature extraction. The shape and pose are described by the base matrix $\{R, \boldsymbol{t}\}$ and the 17 vectors $\boldsymbol{v}_n$. The lengths of the body links are given by l_n. Our method converts the pose at each frame into this coordinate system.

4.4.1 Human Model for Motion Feature Extraction

We first convert motion capture data into our simple human body model for motion feature extraction. Figure 4.1 illustrates our human model. In our model, a human pose at each frame is converted into body center coordinate system as described in Section 2.5.1. In the following, the x-axis, the y-axis and z-axis are referred to as $\boldsymbol{r}_x$, $\boldsymbol{r}_y$ and $\boldsymbol{r}_z$, respectively, and the origin position of the body center coordinate system is denoted as $\boldsymbol{t}$. $\boldsymbol{v}_n$ is a unit vector representing the direction of the n-th body link in the body center coordinate system $\{R, \boldsymbol{t}\}$, and l_n represents the length of the n-th body link.

4.4.2 Weight Effort

According to Laban's definition, the Weight Effort component represents the strength of motion. Thus, we define the Weight Effort component W as the linear sum of the approximated instantaneous momentum magnitude calculated from

the link and body directions:

$$W(f) = \sum_i \alpha_i \arccos\left(\frac{\boldsymbol{v}_i(f)}{|\boldsymbol{v}_i(f)|} \cdot \frac{\boldsymbol{v}_i(f+1)}{|\boldsymbol{v}_i(f+1)|}\right) + \sum_{j\in\{x,y,z\}} \arccos\left(\frac{\boldsymbol{r}_j(f)}{|\boldsymbol{r}_j(f)|} \cdot \frac{\boldsymbol{r}_j(f+1)}{|\boldsymbol{r}_j(f+1)|}\right), \tag{4.3}$$

where α_i is a regularization parameter for the i-th link. These regularization parameters depend on which parts we recognize as important for dance expression. For example, if we recognize the hands and feet as important, α corresponding to the hands and feet will be greater than those corresponding to other parts.

4.4.3 Motion Rhythm Feature

Considering the characteristics of the Weight Effort component, the local minimums of this component indicate stop motions, which are important moments in dance performance. We recognize these local minimums as motion keyposes, and define the motion rhythm features F_R^{Motion} as follows:

$$F_R^{\text{Motion}}(f) = \begin{cases} 1 & \text{if } W(f) \text{ is around the local minimum} \\ 0 & \text{otherwise} \end{cases}. \tag{4.4}$$

4.4.4 Motion Intensity Feature

Motion intensity is related to not only momentum but also forward translation. We obtain instant motion intensity I from the momentum W and the speed of the forward direction $\boldsymbol{r}_x \cdot \dot{\boldsymbol{t}}$:

$$I(f) = W(f) \cdot \left(1.0 + k \cdot \boldsymbol{r}_x(f) \cdot \dot{\boldsymbol{t}}(f)\right) \tag{4.5}$$

in which k is a regularization parameter between the Weight Effort and the speed. Finally, we calculate the average of the instantaneous motion intensity from the previous motion keypose f_i^R to the next one f_{i+1}^R, and set it to the motion intensity:

$$F_I^{\text{Motion}}(f) = \sum_{i=f_i^R}^{f_{i+1}^R} \frac{I(i)}{f_{i+1}^R - f_i^R}. \tag{4.6}$$

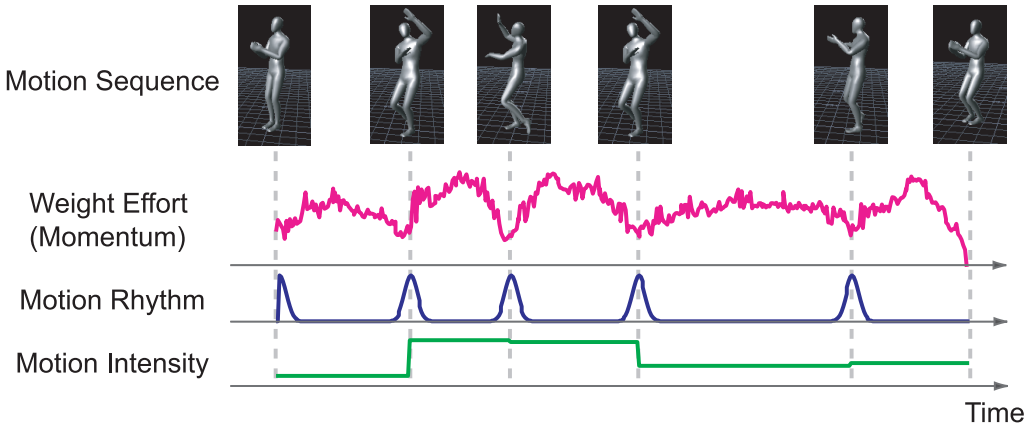


Figure 4.2: Motion feature vector of an example motion. Motion rhythm and intensity components are obtained from Weight Effort of body movement. The motion rhythm component is the local minimum of the Weight Effort component (dashed lines), and motion intensity comes from the average of Weight Effort and forward translation of the body within the neighboring motion rhythm frame.

4.5 Music Feature Analysis

When people listen or dance to music, they extract musical features from an audio signal. The important features for dance performance are music structure, rhythm, and intensity. This section describes how to acquire music segments, and how to extract the music rhythm and intensity features. In the following, $X(t,k)$ denotes the spectral power of k-th note at t-th temporal frame.

4.5.1 Music Segment Acquisition

With respect to music structure, we first note the following key principle:

Principle 3: Music structure consists of the repetition of several phrases.

The goal of this analysis is to extract the patterns of the repeating phrases and to segment the music by the extracted repeating patterns.

Some phrases may be repeated by different instruments; e.g., one phrase is performed by a vocalist, and a repetition of the phrase is performed by a

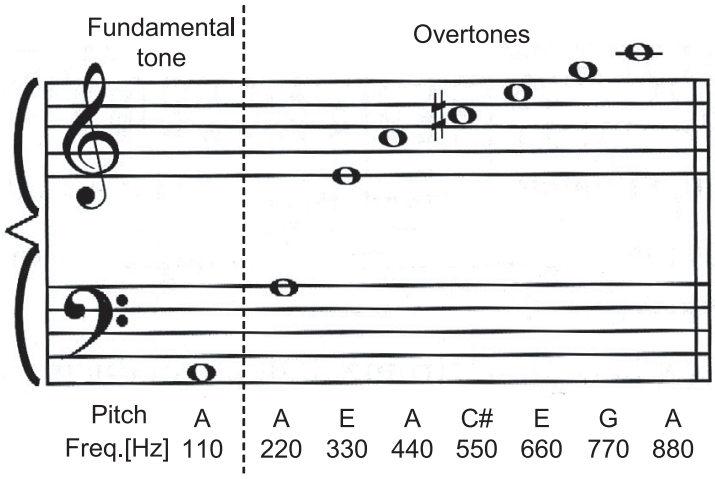


Figure 4.3: Example of fundamental tone 'A' and its overtones. When a sound 'A' whose frequency is around 110Hz is produced, its overtones, whose frequencies are integral multiples of the fundamental tone, are also produced.

guitar). However, people can easily recognize that they are the same phrases, and therefore the structure analysis method should depend on the sequence of the notes, but not be affected by the timbre of the instruments.

Figure 4.3 shows a mechanism of timbre. The timbre of every instrument has a basic characteristic that it always consists of a fundamental tone and its overtones, whose frequencies are integral multiples of the fundamental frequency, but the energies of the overtones differ from one instrument to another. Because of this, it is difficult to extract accurate repeating patterns directly in the frequency domain.

In order to find repeating patterns, we use CQT feature vectors and evaluate these with a structure-based similarity measurement that is independent of timbre effects, as proposed by Lie *et al*. [WLZ04]. First, we calculate the auto-correlation of the elements of the difference vector:

$$r_{ij}(m) = \sum_{n=0}^{N-m-1} \Delta v_{ij}(n+m) \cdot \Delta v_{ij}(n), \tag{4.7}$$

where $\Delta v_{ij}(n)$ is the absolute difference of the n-th CQT feature vector element between the i-th and j-th temporal frames:

$$\Delta v_{ij}(n) = |X(i,n) - X(j,n)|, \tag{4.8}$$

and N is the number of the elements of CQT feature vectors. If the CQT feature vectors contain the same pitch sound, the peaks of $r_{ij}(m)$ will have *harmonic intervals* that are based on the characteristics of the overtones, and if not, the peaks will appear without this interval. In detail, if the vectors contain the same pitch, the peak of $r_{ij}(m)$ will strongly appear at $m = 0, 12, 19, 24, 29$ etc., which represent the fundamental frequency f_b and its integral multiples $2f_b, 3f_b, 4f_b, 5f_b$. This characteristic is modeled as the spiral array [Che01], and the elements of the weighting vector $w(m)$ for $\boldsymbol{r}(i,\ j) = [r_{ij}(0), r_{ij}(1), \cdots, r_{ij}(N)]^T$ are represented as

$$w(m) = \frac{1}{A}|\boldsymbol{p}(7m \bmod 12) - \boldsymbol{p}(0)|, \tag{4.9}$$

where A is a normalization factor to satisfy $\sum_m w(m) = 1$, and

$$\boldsymbol{p}(m) = \begin{bmatrix} \sin\frac{m\pi}{2} \\ \cos\frac{m\pi}{2} \\ \frac{m\pi}{2} \end{bmatrix}. \tag{4.10}$$

Accordingly, the distance D between two CQT feature vectors is considered the neighboring frames and evaluated as follows:

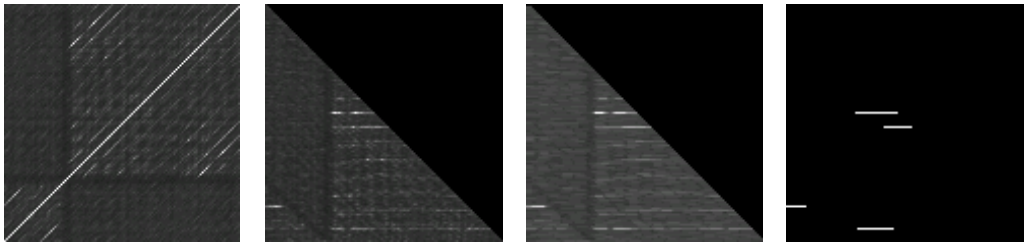
$$D(i,j) = \frac{1}{2N_r}\sum_{k=-N_r}^{N_r-1} \boldsymbol{w} \cdot \boldsymbol{r}(i+k,\ j+k), \tag{4.11}$$

where $\boldsymbol{w}$ represents the weighting vector, and $2N_r$ is the range for the distance calculation.

Once the distance function is defined, we can get the similarity matrix $\boldsymbol{S}$ whose elements are the similarity measurements $1/D(i,j)$, and then convert it to *time-lag matrix* T:

$$T_{ij} = S_{i,i+j} = \frac{1}{D(i,i+j)}. \tag{4.12}$$

Figure 4.4 shows examples of these matrices. In this figure, the brighter regions show the greater similarity, and several white horizontal lines appear clearly in the time-lag matrix. These lines denote the repeating patterns. By extracting them, we can acquire the repeating phrases, and analyze the structure of the input music. More specifically, the *Erosion* and *Dilation* operators often used in image processing [GW02] are applied to make the lines more clear; the lines



(a) (b) (c) (d)

Figure 4.4: Repeating pattern analysis steps. (a) Similarity matrix, (b) time-lag matrix, (c) time-lag matrix after Erosion and Dilation operations, and (d) result of repeating phrases extraction.

can then be extracted with a thresholding process. Finally, music segments are extracted by dividing the music sequence at the boundaries of resulting repeating phrases. The other musical features are extracted and assigned to each music segment.

4.5.2 Music Rhythm Feature

To extract music rhythm, we employ the onset component-based rhythm estimation described in Section 2.4. After the music rhythm estimation process, the musical rhythm feature F_R^{Music} is defined as follows:

$$F_R^{\text{Music}}(f; \mathcal{M}) = \begin{cases} 1 & \text{if } f \text{ in } \mathcal{M} \text{ is estimated rhythm time} \\ 0 & \text{otherwise} \end{cases} . \tag{4.13}$$

4.5.3 Music Intensity Feature

To extract music intensity, we first note the following principles:

Principle 4: The spectral power of a melody line is likely to increase during increasing intensity in the music.

Principle 5: A melody line is likely to be performed using a higher range than the C4 note.

Many surveys on auditory psychology [Roa96] say that our ears tend to recognize only the sound whose spectral power is the strongest among the neighboring frequency sounds. Application of this principle is used in many audio signal compression algorithms such as MP3. Accordingly, a temporally average spectral power $\bar{X}$ of k-th note within a music segment $\mathcal{M}$ is calculated to figure out which note sounds are produced in the music segment:

$$\bar{X}(\mathcal{M}, k) = \frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} X(t, k), \tag{4.14}$$

where $|\mathcal{M}|$ denotes the number of the CQT feature vectors in $\mathcal{M}$, and then the local peaks X_{peak} of each average CQT feature vectors are picked up:

$$X_{\text{peak}}(\mathcal{M}, k) = \begin{cases} \bar{X}(\mathcal{M}, k) & \text{if } \bar{X}(\mathcal{M}, k) > \bar{X}(\mathcal{M}, k \pm 1) \\ 0 & \text{otherwise} \end{cases}. \tag{4.15}$$

In order to extract music intensity feature F_I^{Music}, we approximately calculate the *sound pressure level*, which considers human auditory properties, and is related to both the amplitude and the frequency:

$$F_I^{\text{Music}}(f; \mathcal{M}) = \log_{10} \left(\sum_{k \in [\text{C4,C6}]} X_{\text{peak}}(\mathcal{M}, k)^2 \cdot f_k^2 \right). \tag{4.16}$$

4.6 Motion Synthesis Considering Motion and Music Features

The final step of our approach is to synthesize new dance motion considering both the motion and music feature vectors. The main purpose and problem of this step is to select the motion segment set from the motion database with as low a loss of correlation as possible. Toward this goal, we propose two methods to synthesize new dance motion:

1. A locally optimal search based on the motion graph algorithm.
2. A segment-based globally optimal search.

The first method locally evaluates the similarity between motion and music features, and finds the optimal path of the constructed motion graph. The second method extracts motion candidate segments by evaluating the similarity of the rhythm features, and finds the optimal sequence of these segments.

4.6.1 Locally Optimal Motion Synthesis

Motion Graph Algorithm

The motion graph algorithm [KGP02] connects similar poses among existing motion sequences and indicates all possible transitions among the existing motion sequence. New synthesized transitions as well as existing motion sequences are included. The analysis step generates such possible new transition paths, and the synthesis step chooses appropriate ones according to the features of the input music. First, we calculate pose similarity between each pair of motion sequences and connect them based on the degree of pose similarity by creating transition motions. This graph structure of a motion data set is called a motion graph. We assign the extracted motion features in the motion graph, and a new dance motion is synthesized by calculating the correlation of the music features and the motion features, and tracing the motion graph based on the correlation results.

The pose distance between frame $f^{\mathcal{A}}$ in the motion sequence $S^{\mathcal{A}}$ and frame $f^{\mathcal{B}}$ in the motion sequence $S^{\mathcal{B}}$ is given by following:

$$\mathrm{Dist}\left(S^{\mathcal{A}}\left(f^{\mathcal{A}}\right), S^{\mathcal{B}}\left(f^{\mathcal{B}}\right)\right) = \sum_i \left(\boldsymbol{v}_i^{\mathcal{A}}\left(f^{\mathcal{A}}\right) \cdot \boldsymbol{v}_i^{\mathcal{B}}\left(f^{\mathcal{B}}\right) + \alpha_i \cdot \dot{\boldsymbol{v}}_i^{\mathcal{A}}\left(f^{\mathcal{A}}\right) \cdot \dot{\boldsymbol{v}}_i^{\mathcal{B}}\left(f^{\mathcal{B}}\right)\right), \quad (4.17)$$

where α_i is the regularization parameter indicating the importance of i-th body portion. The first term can calculate the similarity of pose, and the second term can calculate the similarity of movement of the links. The value of the distance function Equation (4.17) is maximized if the poses and movements at $f^{\mathcal{A}}$ and $f^{\mathcal{B}}$ are similar. In order to detect the connection frames, we apply thresholding to the value of the distance function and generate a new transition motion between selected frames.

Transition Motion Synthesis

Transition motions are calculated using 3rd order interpolation of body links. This interpolation can consider the smoothness of position, velocity, and acceleration. For example, assume that we would like to interpolate the motion between the pose $S^A(f^A)$ and the pose $S^B(f^B)$ with given duration T (frames), then the vectors $\{\boldsymbol{v}_i(f) | 0 \leq f \leq T, 0 \leq i \leq 16\}$ are given by the following:

$$\boldsymbol{v}_i(f) = f^3 \cdot \boldsymbol{a}_i + f^2 \cdot \boldsymbol{b}_i + f \cdot \boldsymbol{c}_i + \boldsymbol{d}_i, \quad (4.18)$$

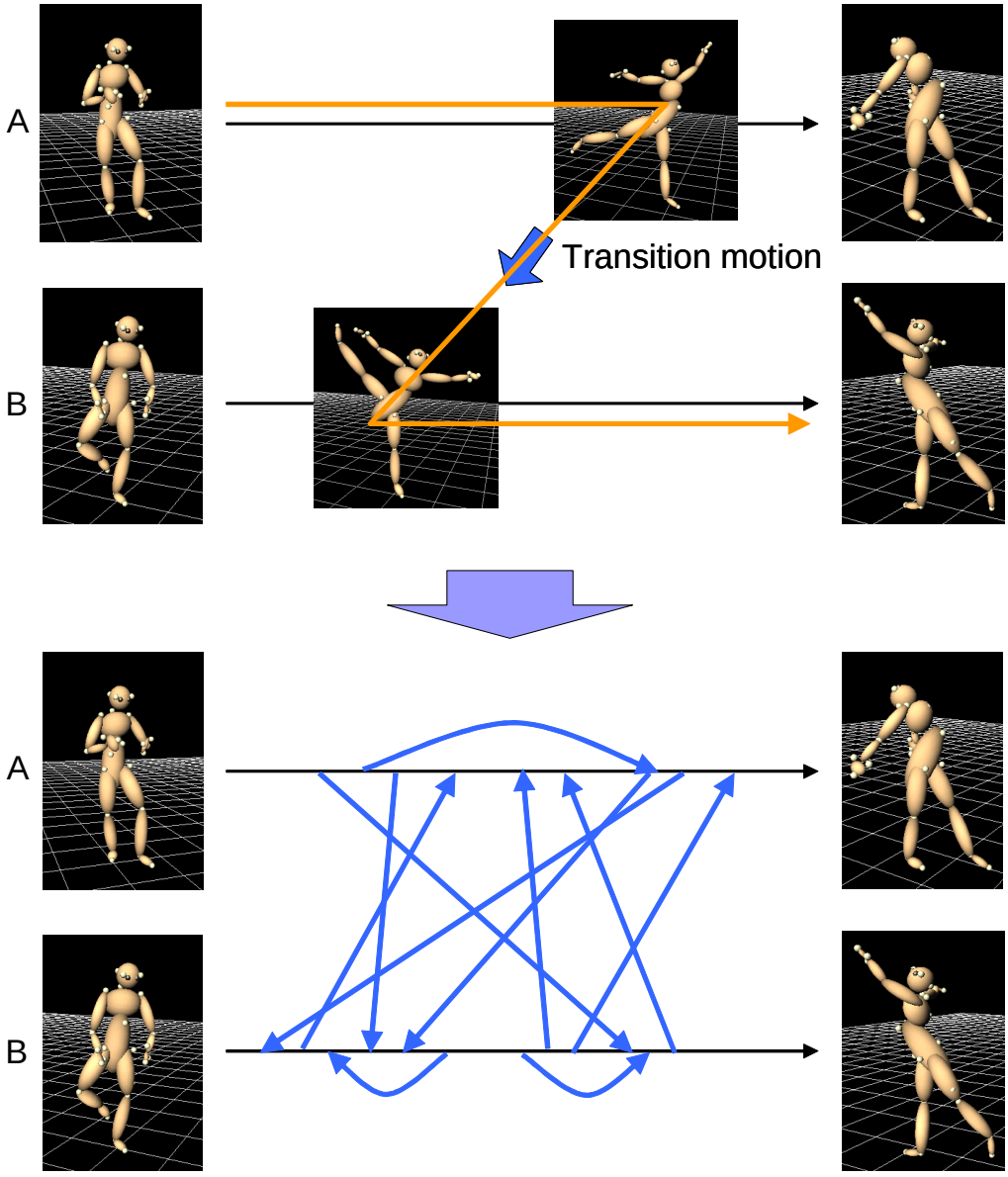


Figure 4.5: Illustration of motion graph construction. The motion graph algorithm tries to find the transition frames whose postures and movements are similar to those in other motion sequences. In this example, a new motion represented by the orange-colored arrows is generated from motion sequence A and B. The transition frame search is applied to all the possible frame pairs, and an oriented graph structure consisting of motion sequences is constructed.

where

$$\boldsymbol{a}_i = \left\{T\left(\dot{\boldsymbol{v}}_i^B - \dot{\boldsymbol{v}}_i^A\right) - 2\cdot\left(\boldsymbol{v}_i^B - \boldsymbol{v}_i^A - \dot{\boldsymbol{v}}_i^A T\right)\right\}/T^3, \tag{4.19a}$$

$$\boldsymbol{b}_i = \left\{-T\left(\dot{\boldsymbol{v}}_i^B - \dot{\boldsymbol{v}}_i^A\right) + 3\cdot\left(\boldsymbol{v}_i^B - \boldsymbol{v}_i^A - \dot{\boldsymbol{v}}_i^A T\right)\right\}/T^2, \tag{4.19b}$$

$$\boldsymbol{c}_i = \dot{\boldsymbol{v}}_i^A, \tag{4.19c}$$

$$\boldsymbol{d}_i = \boldsymbol{v}_i^A. \tag{4.19d}$$

As for the root motion, we also use 3rd order interpolation for body center coordinates $\{\dot{\boldsymbol{t}}^{\mathcal{A}}, R^{\mathcal{A}}, \dot{R}^{\mathcal{A}}\}$ and $\{\dot{\boldsymbol{t}}^{\mathcal{B}}, R^{\mathcal{B}}, \dot{R}^{\mathcal{B}}\}$. In order to maintain relative posture as well as maintain both feet in a state of contact with the ground, a vertical translation t_z and a vertical angle of the body center coordinate system $\theta_z = \arccos(\boldsymbol{r}_z \cdot \boldsymbol{z})$, where $\boldsymbol{z}$ is the vertical axis in the global coordinate system, must be maintained at the destination frame. To accomplish this, we first determine the frontal direction of the body center coordinate by using third order interpolation, then tilt and translate these parameters to satisfy the constraint.

The duration for a transition is determined by the angular distance of the concatenated frames and the maximum velocity of the concatenated motions. In the motion analysis step, we determine the maximum angular velocities of all body portions for all motion data. The duration is determined within the range in which the angular velocities during transition do not exceed these maximums. This process is used to avoid unnatural transitions such as the hands moving too fast compared to neighboring motion sequences.

Dance Motion Synthesis Using Motion Graph

Now we have the motion graph and motion features from a set of motion sequences, and we also have extracted musical features from input music data. Paths of the motion graph are selected by calculating the correlation between the music rhythm feature F_R^{Music} and the motion rhythm feature F_R^{Motion}. In theory, all the frames of motion sequence in the motion graph and all the music frames should be considered to detect the best motion graph path. But this would have a heavy computational cost. So our algorithm considers every motion graph path from the current time t to $t + T$, where T is the search range and is set to 3 seconds in our experiments. The motion and music similarity of rhythm component S_{rhythm} is described as follows:

$$S_{\text{rhythm}}(t, path) = \sum_{\tau=0}^{T}\left(F_R^{\text{Motion}}(t+\tau; path)\cdot F_R^{\text{Music}}(t+\tau)\right), \tag{4.20}$$

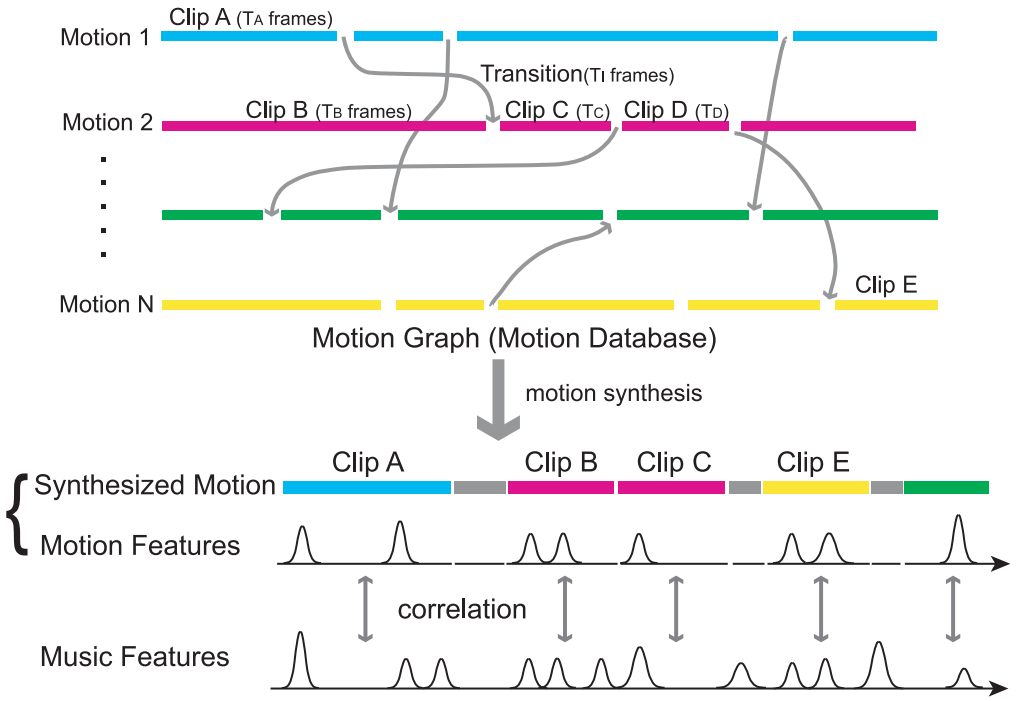


Figure 4.6: Overview of our locally optimal motion synthesis algorithm. Motion is generated by tracing a particular motion graph in the database. Evaluation of the synthesized motion is the sum of the convolution between the motion feature vectors and the music feature vectors. The system chooses the path evaluated at the largest possible value.

where $F_R^{\text{Motion}}(t+\tau; path)$ is the motion rhythm component along the motion graph path *path*. We choose several motion graph paths whose S_{rhythm} are highest. We define these chosen motion graph paths as "$\mathcal{RP}$."

The final step in synthesizing a dance motion is to detect the best motion graph path from $\mathcal{RP}$. In this step, the highest evaluation path is found, and the resulting motion is produced. However, a problem remains because the evaluation values accumulated on each path depend on the selection of previous paths. This is addressed by calculating the following correlation function between the music intensity feature F_I^{Music} and the motion intensity feature F_I^{Motion}; the locally optimal path $\hat{p}$ of the motion graph is then obtained as follows:

$$\hat{p} = \arg\max_{p \in \mathcal{RP}} \sum_{\tau=0}^{T} \left\{ F_I^{\text{Motion}}(t+\tau; p) \cdot F_I^{\text{Music}}(t+\tau) \right\}. \tag{4.21}$$

On each transition path, we calculate a matching evaluation between the music feature and the motion feature of the destination path, and thus the locally optimized path is obtained.

4.6.2 Globally Optimal Motion Synthesis

In order to find a globally optimal solution, we perform three steps to synthesize a new dance motion. Figure 4.7 gives an overview of our motion synthesis algorithm. First, we evaluate the similarity of the rhythm components and detect candidate motion segments which strongly correspond to each music segment. Next, connectivity analysis is applied to determine whether or not transition motions between the neighboring motion segments look natural. Then, possible sequences of motion segments are extracted. Finally, we analyze the similarity of the intensity components of the music segments and the selected motion segment sequences, and thereby synthesize new dance motions by connecting the motion segments with each other.

Similarity Evaluation of Rhythm Features

In this step, we extract the candidate motion segments from every input motion sequence, considering motion and music rhythm components. To include more detail, we focus on one input motion sequence whose length is L_{motion} and a music segment $\mathcal{M}$ whose length is L_{music}. In our method, we allow a slight stretching of the duration of the input motion sequence. When calculating

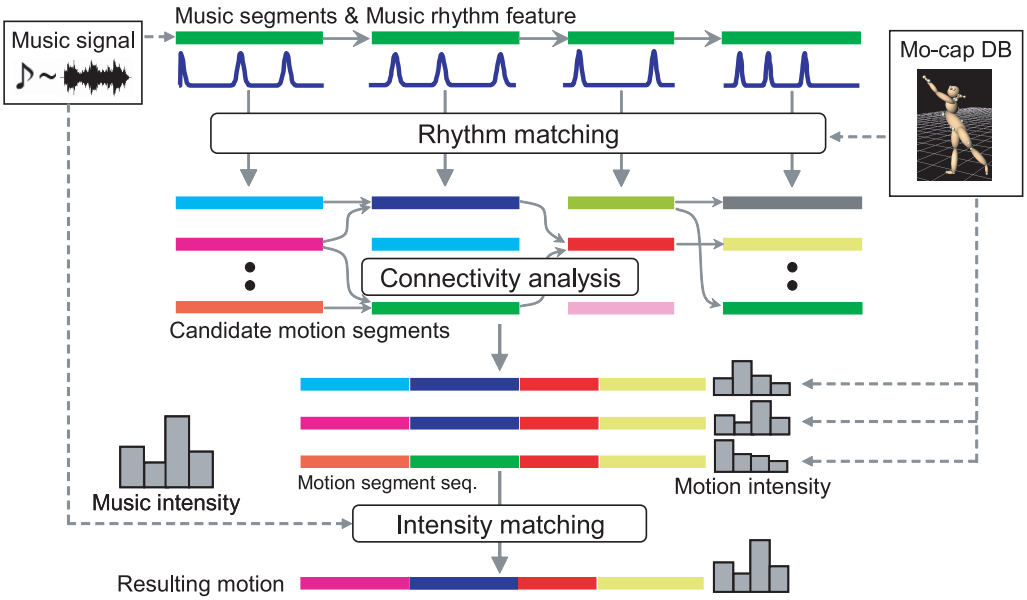


Figure 4.7: Overview of our globally optimal motion synthesis algorithm. For each music segment, candidate motion segments are obtained from a motion database by evaluating the similarity with music rhythm components. All possible motion segment sequences can be acquired by connectivity analysis between neighboring motion segments. Finally, we evaluate the similarity of the intensity components of the motion segment sequences and the music segments, and thereby synthesize new dance motions by connecting the motion segments with each other.

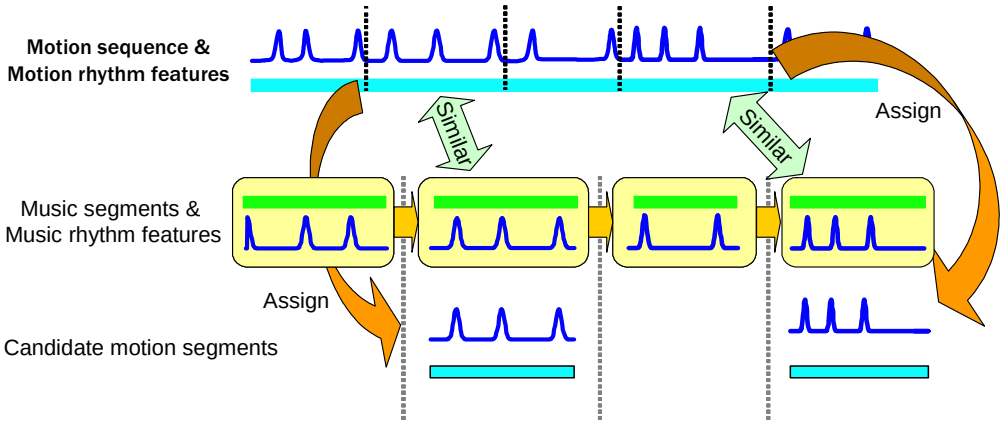


Figure 4.8: Procedure for rhythm feature similarity evaluation. For each music segment, Equation (4.22) is applied to each motion sequence, and candidate motion segments are acquired.

the similarity evaluation of two rhythm components, we consider not only the rhythm components themselves but also the scaling parameter $s \in [0.9\ 1.1]$ and the offset parameter f_o, which represents the frame from which a motion segment starts. We extract the scaling parameter $\hat{s}$, which maximizes the similarity measurement

$$\hat{s} = \arg \max_{s} \sum_{f=0}^{L_{\text{music}}} \frac{F_R^{\text{Music}}(f; \mathcal{M}) \cdot F_R^{\text{Motion}}(s \cdot f + f_o)}{F_R^{\text{Music}}(f; \mathcal{M}) + F_R^{\text{Motion}}(s \cdot f + f_o)} \tag{4.22}$$

for each $f_o \in [0,\ L_{\text{motion}} - L_{\text{music}}]$.

We extract all possible sets of (s, f_o) for each motion sequence, and apply a simple thresholding process to the parameter sets. Using the remaining parameters, we can extract candidate motion segments for each music segment.

Connectivity Evaluation of Motion Segments

Whether or not synthesized motion looks natural strongly depends on connectivity analysis. In this step, we consider both posture similarity S_{pose} and movement similarity S_{move}.

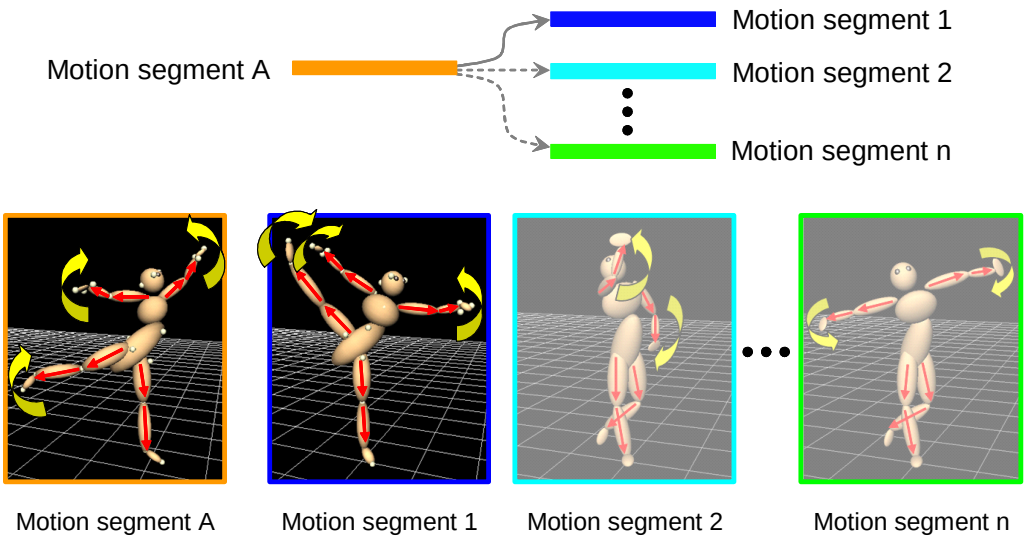


Figure 4.9: Procedure for connectivity evaluation between motion segments. This procedure is based on the two similarity measurements: posture similarity and movement similarity. Posture similarity is evaluated with each link's direction vectors (red arrows), while movement similarity is evaluated from each link's movements (yellow arrows). In this example, Motion segment A is connected with Motion segment 1.

Posture similarity S_{pose} between the $i^{\mathcal{A}}$-th frame of the motion segment $\mathcal{A}$ and the $j^{\mathcal{B}}$-th frame of the motion segment $\mathcal{B}$ is defined as the angular similarity of the link direction vectors:

$$S_{\text{pose}}(i^{\mathcal{A}}, j^{\mathcal{B}}) = \sum_{l} \beta_l \cdot \boldsymbol{v}_l(i^{\mathcal{A}}) \cdot \boldsymbol{v}_l(j^{\mathcal{B}}), \tag{4.23}$$

where β_l is a regularization factor for the l-th link.

With regard to movement similarity S_{move}, we use velocity vectors in homogeneous coordinates. This is possible because the angular distance measure of their unit vectors in the homogeneous coordinates accounts for the differences in both direction and magnitude. Specifically, movement similarity S_{move} is

calculated as follows:

$$S_{\text{move}}(i^{\mathcal{A}}, j^{\mathcal{B}}) = \prod_{l} g\left[h\left(\boldsymbol{v}_l(j^{\mathcal{B}}) - \boldsymbol{v}_l(i^{\mathcal{A}})\right) \cdot h\left(\dot{\boldsymbol{v}}_l(i^{\mathcal{A}})\right)\right] \cdot g\left[h\left(\boldsymbol{v}_l(j^{\mathcal{B}}) - \boldsymbol{v}_l(i^{\mathcal{A}})\right) \cdot h\left(\dot{\boldsymbol{v}}_l(j^{\mathcal{B}})\right)\right], \tag{4.24}$$

where

$$g[x] = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}, \tag{4.25}$$

and $\dot{v}$ is calculated from the original input motion sequence, not the candidate motion segment. Through h, an input 3D vector is converted to the 4D unit vector, which represents the direction vector in 4D homogeneous coordinates.

$$h(\boldsymbol{v}) = \frac{\left(\boldsymbol{v}^T, 1\right)^T}{|\boldsymbol{v}^T, 1|} \tag{4.26}$$

Thanks to this conversion, Equation (4.24) can account for differences in both direction and magnitude [BFB94, SMKT06]. That is, Equation (4.24) evaluates the similarity of the directions between the original movement in the input motion sequence and the motion to be synthesized.

Finally, connectivity is analyzed from both S_{pose} and S_{move} between the end frame of one motion segment and the beginning frame of the neighboring motion segments. From the results of the connectivity evaluation, we obtain the candidate sequences of the motion segments that satisfy the requirements for similarity with the rhythm features and naturalness of the synthesized motion.

Similarity Evaluation of Intensity Features

Next, we evaluate the intensity components of the candidate sequences of the motion segments and input music. In order to find a globally optimal solution, we consider the time series of the intensity features as a histogram, and the Bhattacharyya coefficient [Kai67] is considered to relatively evaluate the similarity between the motion and music intensity histograms. Hence, we finally obtain the motion segment sequence $\hat{D}$ that maximizes the Bhattacharyya coefficient:

$$\hat{D} = \arg \max_{D \in \mathcal{CS}} \sum_{j} \sqrt{\frac{F_I^{\text{Music}}(j)}{\sum_k F_I^{\text{Music}}(k)} \cdot \frac{F_I^{\text{Motion}}(j)}{\sum_{k \in D} F_I^{\text{Motion}}(k)}}, \tag{4.27}$$

where $\mathcal{CS}$ represents the candidate sequences of the motion segments after the analyses of rhythm similarity and connectivity.

Transition Motion Generation

The resulting motion sequence is acquired by connecting the neighboring motion segments. For posture, we use a spline function with a first and second order differential to interpolate motion segments. This is slightly different from the interpolation described in Section 4.6.1 in that the transition motion generation method described here is a blending process, while that described in Section 4.6.1 is a interpolation technique. For the position of a character, we pay attention to the position and posture relative to the ground in order to avoid effects such as sliding or being stuck in one position.

Interface for Designing Dance Motions

Our method can synthesize new dance performances that match the input music well. However, the resulting motion sequence may not reflect a given animator's intention. For example, an animator may want a character to jump when vocal input music says, "*Jump!*"

Our system supports many common animator designs. Figure 4.10 shows our interface that enables animators to design motions. The left list shows the music segment sequence, and the central list shows the extracted motion segments corresponding to the currently assigned music segment. A user can confirm the music segments and the motion segments by selecting and double-clicking an item from the lists.

Using our system, a desired motion segment can be assigned to a music segment as an animator wishes. However, it is conceivable that there are no candidate sets of motion segments that satisfy a desired design. If so, our system re-evaluates the motion and music features under this new constraint.

4.7 Experiments

4.7.1 Experimental Data

We have tested our proposed method on six Japanese dances; the *Aizu-bandaisan* dance, the *Jongara-bushi* dance, the *Kansho-odori* dance, the *Soran-bushi* dance, the *Mikagura* dance, and the *Nishimonai-ondo* dance. The first three dances, the Aizu-bandaisan dance, the Jongara-bushi dance and the Kansho-odori dance, were captured with an optical motion capture system produced by Vicon at a sampling rate of 120 Hz. The other three dances, the Soran-bushi dance, the

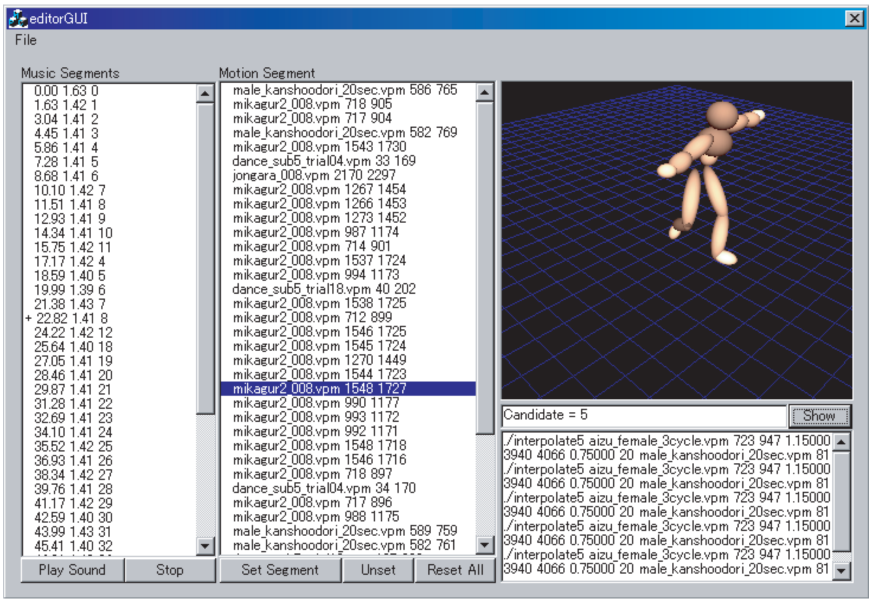


Figure 4.10: Our user interface for designing motion. A user can confirm the music and motion segments by selecting and double-clicking an item out of the lists in which the music segments and their corresponding motion segments are displayed from left to center, respectively. The process of designing motion is accomplished by assigning the desired motion segment to the music segment. The resulting motion is displayed in the top-right window.

Mikagura dance and the Nishimonai-ondo dance, were captured using a magnetic motion capture system produced by Ascension Technology Corporation at a sampling rate of 30 Hz (see Appendix B.2) in cooperation with Warabi-za [War]. These magnetic motion data required conversion to the same format as that of the optical data before it could be inserted in our motion database. The total length of the motion capture data set is about 180 seconds.

The input music data for dance motion synthesis was the *Kansho-odori* dance. The length of this musical input data was approximately 40 seconds, with 16-bit sampling at 32 kHz.

4.7.2 Results of Japanese Original Dance Synthesis

We estimated the rhythm of the Kansho-odori dance music; its average rhythm interval was approximately 0.504 seconds. Additionally, when a vocalist sings more loudly, this increased volume results in increased musical intensity via our music intensity analysis.

The result of dance performance synthesis is shown in Figure 4.11, and the result of our proposed method's capability at motion and music feature matching is shown in Figure 4.12. In this figure, the yellow and light blue lines show the motion and music rhythm components, and the blue and red lines are the intensity histograms of motion and music segments, respectively. The synthesized dance performance seems to be well matched to the rhythm of the music. In addition to the results of rhythm feature matching, the increased music intensity results in the increased motion intensity and the resulting dance motion seems to be more exciting. Thus, we conclude that our proposed method is effective.

4.7.3 Results of Original Dance Synthesis with Various Motion Database

Our proposed method is applicable not only to Japanese folk dance but also different styles of dance such as break dancing. We have tested our proposed method on portions of a large motion database consisting of break dance, Indian dance, and dance motion with simple arm and leg movement; these data are all downloadable from the CMU Motion Capture Database [CMU]. All the motion data were captured with an optical motion-capture system produced by Vicon running at a sampling rate of 120 Hz. The input music data we used for our experiments was approximately 60 seconds long; the sampling format was 16-bit stereo at 44.1 kHz.

Figure 4.11: Synthesis result for Japanese dance music *Kansho-odori*.

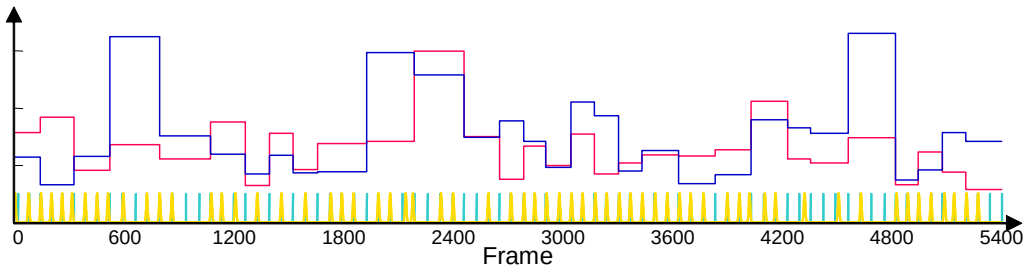


Figure 4.12: Feature matching result for Japanese dance music *Kansho-odori*. Yellow and light blue lines represent motion and music rhythm components and blue and red lines represent motion and music intensity components, respectively.

Title (Genre)	Rhythm [sec] ([bpm])
La Cumparsita (tango)	0.454 (132)
Tonite (pops)	0.476 (126)
Carmen Suite (classic)	0.417 (144)
Nutcracker Suite (classic)	0.714 (84)

Table 4.1: Results of music feature analysis.

Results of Music Feature Analysis

We first show the results of our proposed method's capability in music feature analysis. We applied the rhythm tracking method to 13 music data sets that contain classical music, rock, tango music and so on. Of these 13 input types, 10 correctly tracked the rhythm and 3 were considered unsuccessful. These errors were derived from the fact that sound onset of string instruments such as violin is very slow. Table 4.1 shows a part of the successful rhythm tracking results. Additionally, our proposed method's music intensity analysis was also successful.

Results of Original Dance Synthesis

Figure 4.13 shows the synthesized motion for tango music "*La Cumparsita*." Figure 4.14 shows the features of the synthesized motion and the input music. In this figure, the yellow and light blue lines show the motion and music rhythm components, and the blue and red lines are the intensity histograms of motion and music segments, respectively. We can easily confirm that most of the musical rhythm is matched to the motion rhythm, and that the distributions of the intensity components are quite similar.

Figure 4.15 shows another synthesized motion for popular music "*Tonite*." Figure 4.16 shows the features of the synthesized motion and the input music. In this figure, the yellow and light blue lines show the motion and music rhythm components, and the blue and red lines are the intensity histograms of motion and music segments, respectively. We can also easily confirm that most of the

musical rhythm is matched to the motion rhythm, and that the distributions of the intensity components are quite similar.

4.7.4 Quantitative Evaluation

When we employed our motion graph-based motion synthesis method, we obtained the motion graph shown in Figure 4.17. In the case of the globally optimal motion synthesis method, we obtained approximately 2500 candidate motion segments through our rhythm matching procedure, and approximately 500 candidate motion sequences after our connectivity evaluation procedure. Note that these numbers can be affected by computing environment factors such as memory limitations. A more powerful computing environment could conceivably arrive at more matching segments.

Computational Cost

Both the locally optimal and globally optimal motion synthesis take much longer than the other analysis steps. Connectivity analysis between neighboring candidate motion segments is the most time-consuming process, because all possible sets of the neighboring segments are checked. In the case of the one-minute-long music *Tonite*, it took approximately 10 minutes to synthesize motion with our globally optimal method, and approximately 30 minutes with our locally optimal method. This resulting motion was generated from 27 input motion data sets (about 520 sec in total) using a Pentium-D 2.8GHz PC without any multi-threaded programming.

4.8 Discussion

Our algorithm can synthesize new dance motion taking into consideration musical and motion rhythms, and musical and motion intensities. This is based on empirical observations that motion rhythm is correlated with musical rhythm, and that music intensity and motion intensity have a direct correlation. Our contribution is, with regard to CG animation, to automatically synthesize motion that synchronizes input music signals, and to take motion expressions extracted from Laban's Weight Effort component into consideration. With regard to artificial intelligence, we have been able to imitate the simple models of human

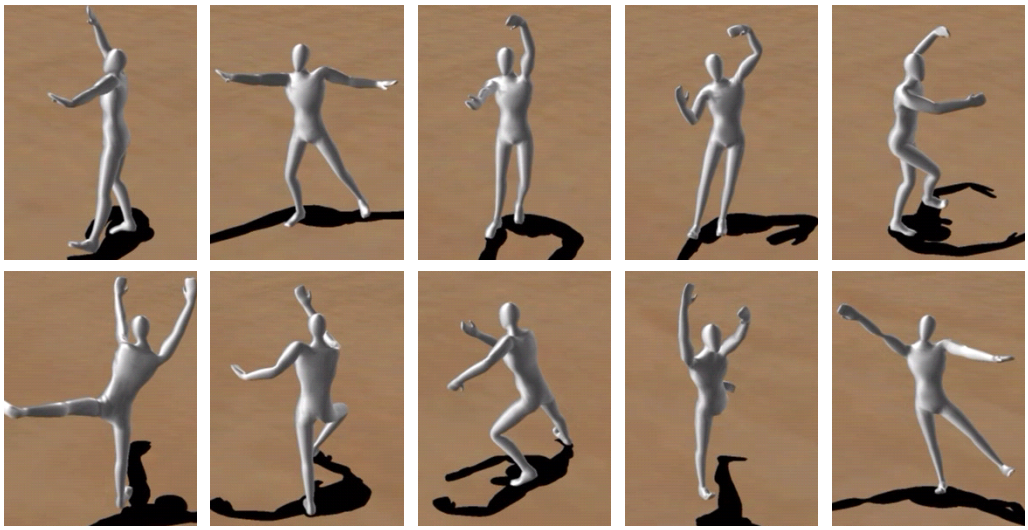


Figure 4.13: Synthesis result for tango music *La Cumparsita*.

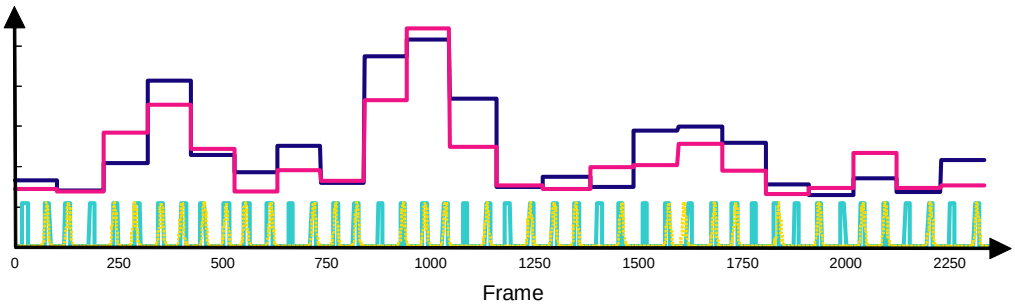


Figure 4.14: Feature matching result for tango music *La Cumparsita*. Yellow and light blue lines represent motion and music rhythm components, and blue and red lines represent motion and music intensity components, respectively.

Figure 4.15: Synthesis result for pops music *Tonite*.

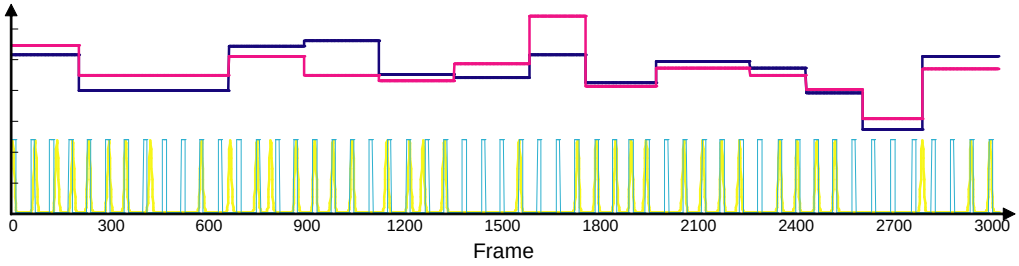


Figure 4.16: Feature matching result for pops music *Tonite*. Yellow and light blue lines represent motion and music rhythm components, and blue and red lines represent motion and music intensity components, respectively.

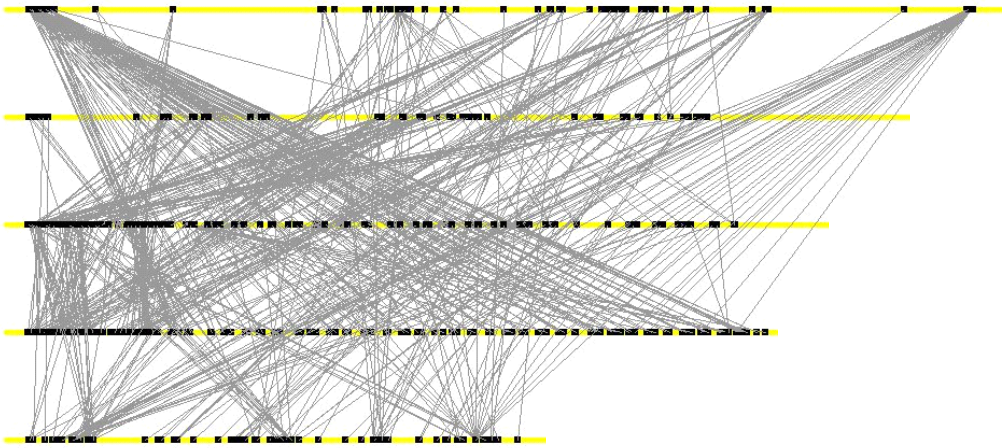


Figure 4.17: Part of the motion graph constructed from 24 input motion sequences.

emotional aspects and the human ability to recognize music features for dance performance while listening to music.

We believe that it is possible to introduce other features for matching, such as relationships between a music chord or key (major/minor) and mood of motion, or a category of music and its appropriate expression in dance. For example, people tend to feel gloomy when listening to music in a minor key, and feel happy when listening to music in a major key. To improve our approach, music psychology could be incorporated. Additionally, motion expressions, which have not been well studied in CG animation, might also be important factors. As future work, we intend to develop a motion expressions analysis method, and introduce this into our method along with corresponding music psychology.

Additionally, we are now developing another application to synthesize dance motions in real time; a character composes new dance motion while listening to music. The purpose of this application is to imitate the ability of *ad-lib* dance which all people, and particularly children, have. This application will also enable a humanoid robot to dance to music as an entertainment robot.

	Locally optimal method	Globally optimal method
Pros	It is always possible to find a solution.	The result is always well matched to input music.
Cons	The resulting motion is not always matched to input music.	It may fail to synthesize a new motion.
Applications	Real-time entertainment system such as video games.	CG animation system such as movie production.

Table 4.2: Locally optimal method vs. Globally optimal method.

4.8.1 Comparison

We have proposed two motion synthesis methods: locally optimal motion synthesis and globally optimal motion synthesis. As for the locally optimal method, we can always obtain a new dance motion because of the motion graph algorithm. However, the features of the resulting motion are not always matched to the features of the input music, because we only obtain the motion sequence with the motion features along the path of the constructed motion graph.

If we obtain the resulting dance motion using our globally optimal method, this motion is always well matched to the features of the input music. However, the problem of the globally optimal method is that this method does not always have a solution. Because this method may not find an optimal motion segments whose features are matched to those of the input music, it may fail to produce a new dance motion.

Therefore, we believe that the locally optimal method would be suitable for real-time entertainment systems such as video games, while the globally optimal method would be more suitable for the creation of CG animation as used in movie production.

4.9 Summary

This chapter presented a method for synthesizing new motion synchronized to music. Our idea is to consider the musical rhythm and intensity components

to be matched to motion rhythm and intensity components. This is an imitation of a dancer's skill in performing motions as they listen to music. Our method can automatically retrieve music features from input music signals and motion features from motion sequences, and synthesize new dance motions whose features are closely matched to those of the music. From the results, it is confirmed that we can successfully synthesize expressive dance performance.

Chapter 5

Conclusions

5.1 Summary

The ultimate purpose of this dissertation is to apply human perceptional models to human motion synthesis. This purpose is strongly motivated by the fact that, although most previous work did not consider the perceptions of performers, human motion is highly affected by these aspects. In particular, because dance performance is so strongly affected by the features of music, in considering Japanese folk dancing as an experimental subject we have sought to imitate our human exploitation of hearing and apply this to CG characters and humanoid robots. To accomplish this, we have developed three methods to analyze and synthesize human dance motion.

Keypose Extraction for Dance Structure Analysis

The first aspect of our proposed method, as described in Chapter 2, is to analyze the keyposes in dance motion. We empirically know that dance motion is always performed to be matched to musical rhythm. We exploit this knowledge by detecting the stop motions in the dance motion data and by estimating the musical rhythm itself, in the form of its onset components. By integrating these information, we can detect keyposes in the dance motion with higher accuracy than has been possible using previous methods. Dancers themselves have corroborated the results of our methods.

Synthesis of Temporally-Scaled Upper Body Motion Based on Aspects of Human Motion

The second aspect of our proposed method, as described in Chapter 3, is to model how upper body motion can be modified depending on musical playback speed. Research in this arena is motivated by the observation that, as music speeds up, dancers omit details of dances in order to keep up with the musical rhythm. To perform this research, we have applied a hierarchical B-spline technique to joint angle sequences in order to analyze them in the frequency domain. From our analysis, we have obtained two insights: (1) keyposes are preserved independently of music speed, and (2) high frequency components of joint angle sequences are gradually attenuated as music speeds up. Using these insights, we modeled our proposed algorithm for modification of upper body motion based on music speed, and we demonstrated that the applications of our technique to CG animation and humanoid robots can result in synthetic motion which is much more realistic than that obtained using previous methods.

Dancing-to-Music Character Animation Based on Aspects of Human Emotion

The third aspect of our proposed method, as described in Chapter 4, is to synthesize *expressive* dance motion using motion and musical features. Our algorithm arises from the observation that people feel quiet and relaxed when listening to relaxing music whereas they feel excited when listening to intense music. Our algorithm is designed such that motion rhythm is synchronized with musical rhythm, and that motion intensity is synchronized with musical intensity. The method can automatically extract musical structure, rhythm, and intensity components from musical signals, motion rhythm and intensity components from motion capture data, and synthesize new dance motion which matches input musical features with realism and high fidelity.

5.2 Contributions

The contributions of this dissertation to this field of research can be summarized as follows:

- We have converted human perceptual models into frameworks for human motion analysis and synthesis. While previous work has considered certain human visual aspects, our method has greatly augmented research on

human motion synthesis by considering auditory aspects as well. This has enabled us to synthesize dance motion very close to actual dancers' understandings of a piece.

- We have proposed a method to extract keyposes from dance motion. Existing methods of keypose extraction or dance structure analysis used only motion capture data, whereas our proposed method considers both motion and musical information. This has enabled us not only to extract the most important aspects of dance performances, but also to understand dance motion structure and generate the most important prescribed motions for a given dance.

- Through our analysis of dance motion performed at varying musical speeds, we have arrived at the two crucial insights mentioned above, which, to our knowledge, have never been formally stated before. Our method's success in generating dance motion confirmed that keyposes are very important musical-speed-independent features.

- We have proposed a method to modify upper body motion based on our obtained insights. Our proposed method can not only synthesize dance performance based on the properties of human motion for CG animation, but can also be applied to humanoid robot motion generation in order to satisfy kinematics constraints.

- We have proposed a method to characterize the emotional features of both motion and music for human motion synthesis. In particular, we have considered a new component, *intensity*, which no studies have previously considered. Most previous methods have proposed analysis methods to evaluate the degree to which multiple motion sequences are seamlessly connected depending on their numerical or physical properties. However, we have discovered that by considering semantic features, much more expressive and realistic dance performances can be synthesized.

- We have proposed a method which can solve the complex optimization issue presented when motion features are to be matched to musical features without the many manual steps required in previous methods. To achieve this, we have proposed two types of optimization; a globally optimal solver using music segments, and a locally optimal solver using the motion graph algorithm. A user can select the most appropriate algorithms depending on his/her needs.

5.3 Future Directions

We conclude this dissertation with a frank discussion of open problems and future improvements in our proposed method which we are interested in pursuing and would like to see pursued by other researchers.

Symbolization of dance performance Via our keypose extraction method, as described in Chapter 2, we can extract or condense important features of dance performance. Traditional motion description methods such as Labanotation can roughly archive dance motion, but they cannot describe the details of dance motion, e.g. which postures are most important for the performance. A complete description of a dance – which would include all the dance's crucial features plus its details – is necessary for archiving and to implement a proper dance teaching system. Recently, intangible cultural heritage in the form of traditional dances such as those we have analyzed is disappearing due to the lack of properly trained successors to our current masters of this art, so demand for a dance teaching system is increasing. Our method should make it possible to preserve important dance performances with complete features.

Style analysis using our hierarchical decomposition technique Through the hierarchical motion analysis discussed in Chapter 3, we have found that the high frequency components – detailed movements of the hands and other extremities – are attenuated depending on the musical playback speed. Additionally, it is clearly confirmed that dance motions between different dancers, e.g. men and women, will vary; all such differences in the details of individual performance can be attributed to style. As described in Section 3.2, most previous methods relied on various stochastic models to analyze human motion style. By extending our hierarchical decomposition technique, we believe that it is possible to analyze motion and to decompose the motion into a base motion vs. style features even if we have only one motion sequence.

Combination of motion modification and dance performance synthesis Our dancing-to-music motion synthesis method considers the rhythm of existing motion data. That is, we cannot guarantee that we synthesize an optimal dance performance in the case that the input musical rhythm varies widely from the motion rhythm. We believe that by combining our hierarchical motion modification method and our dance motion synthesis algorithm in a new way, we will solve this issue and obtain widely varying kinds of new dance performance.

Consideration of other motion and musical features As has been discussed in Section 4.8, it is surely possible to introduce other features for motion/music matching, such as the relationship between a music chord (major/minor) and the mood of the dance motion, or a category of music and its appropriate expression in dance. Currently, if we want to synthesize ballet motion, we must start with a motion database which contains only ballet motion primitives. However, we believe that human motion will contain some unidentified, yet essential, characterization of the mood of human emotion, because all humans can easily recognize a mood when expressed as a motion. Therefore, there might be some aspects of the psychology of music which could be incorporated. It might be quite fruitful to research and develop some form of a motion expressions analysis method, and to integrate such methodology into our method.

Real-time dance motion synthesis As has also been discussed in Section 4.8, both the globally optimal and the locally optimal methods cannot currently run in real-time. We are currently developing another application which would synthesize dance motions in real-time: a character composing new dance motion while listening to music. The purpose of this application is to imitate the ability of *ad-lib* dance which all people, and particularly children, have. This application would also enable a humanoid robot to dance to music; one can easily imagine how this could be used to create an entertainment robot.

Appendix A

Constant Q Transform

A.1 Fourier Transform

A Fourier transform decomposes a function into a continuous spectrum of its frequency components. In mathematical physics, the Fourier transform of an input signal $x(t)$ can be thought of as that signal in the frequency domain [Mit98]. The formulation is as follows:

$$X(f) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} x(t) \exp(-jft)\, dt \tag{A.1}$$

for each frequency f, where j represents $\sqrt{-1}$, and $X(f)$ represents the spectral power at the frequency f.

When the input signal is discrete, such as audio wave signals or the color values of images, the input signal $x_d(n)$ is formulated as follows:

$$x_d(n) = x(t) \cdot \delta(t - nT), \tag{A.2}$$

where T is the sampling interval. From Equation (A.1) and Equation (A.2), the discrete Fourier transform is given as follows:

$$X(k) = \sum_{n=0}^{N-1} x_d(n) \exp\left(-j\frac{2\pi}{N}kn\right) \tag{A.3}$$

for $k \in [0, N-1]$, where $X(k)$ is the spectral power at k-th frequency, and N represents the number of the sampled data.

This equation assumes that the input signal $x_d(n)$ is repeating. If there is a discontinuity between $x_d(N-1)$ and $x_d(0)$, the resulting frequency components include undesirably strong dense spectral power in the high frequency range. To avoid this problem, a window function $W(n)$ that smoothes the gap between $x_d(N-1)$ and $x_d(0)$ can be applied to the sampled data [Har78]:

$$X(k) = \sum_{n=0}^{N-1} W(n) x_d(n) \exp\left(-j\frac{2\pi}{N}kn\right). \tag{A.4}$$

The Hanning window, Hamming window, and Blackman window are well-known window function for the Fourier transform. Especially, widely used for speech/music signal processing is the Hamming window.

A.2 Constant Q Transform

Music is different from speech in that music consists of a sequence of musical notes whose frequencies are already sharply defined. Ideally, it is most appropriate for the extraction of musical features that music signals are converted into a note sequence. But most of the frequency component extraction methods, such as Fourier transforms, do not consider this property of music. In order to extract frequency components representing musical notes more accurately, the *constant Q transform* (CQT) was proposed by Brown [Bro90]. The CQT method sets up a bank of filters whose center frequencies represent musical notes, and enables extraction of the spectral energy of each note.

In our implementation, we extract the spectral energies of the 37 semi-tones (over three octaves from the C3 note to the C6 note) from audio signal $x(n)$ as follows:

$$X(k) = \frac{1}{N_k} \sum_{n=0}^{N_k-1} x(n) \exp\left(-j\frac{2\pi Q n}{N_k}\right), \tag{A.5}$$

where j represents $\sqrt{-1}$, $X(k)$ represents the spectral power of the k-th note, and N_k is the window size.According to music theory, the frequency of the k-th note is calculated as

$$f_k = f_0 \cdot 2^{k/N_{\text{octave}}}, \tag{A.6}$$

where f_0 is the minimal frequency that we are interested in for analysis and is set to 130.8 Hz, the pitch of the C3 note, and N_{octave} denotes the number of semi-tones in one octave and is typically set to 12. Q is a constant ratio of frequency to resolution:

$$Q = \frac{f_k}{f_{k+1} - f_k} = \frac{1}{2^{1/N_{\text{octave}}} - 1}. \tag{A.7}$$

Accordingly the window size N_k is set to be

$$N_k = \lfloor f_s Q / f_k \rfloor, \tag{A.8}$$

in which f_s represents the sampling rate of the input audio signal. Our method uses the Hamming window function, shifts it by a certain interval, and then calculates the CQT component until the window reaches the end of the music, similar to the technique used when computing the short-time FFT.

Appendix B

Motion Capture Systems

In this thesis, we use motion capture systems to precisely record a performer's motion. There are two types of motion capture systems; optical motion capture systems and a magnetic motion capture systems. In this appendix, we explain some details regarding these two types of motion capture systems and the implications of these technologies for our obtained data.

B.1 Optical Motion Capture Systems

There are several makers of optical motion capture systems; we used one produced by Vicon. Figure B.1 shows a scene where the motion of a performer is being captured using this type of optical motion capture system. The system consists of eight infra-red cameras that generate infra-red illumination and many markers that reflect infra-red rays. The system calculates the 3D position of all the markers through triangulation. In order that the cameras can see the markers, a performer must wear a special suit that is closely fitted to the performer's body; the markers must not be occluded by clothes.

The marker model we used is shown in Figure B.2. An optical motion capture system can only capture the 3D position of the markers, and not their spatial orientations. Most computer graphics and robotics applications, however, need joint angle information to manipulate CG characters and robots. Therefore, they need to calculate the implied joint angles from marker position data. So many markers are typically attached to joints such as wrist and ankle which have high degrees of freedom . Using an optical motion capture system, a motion capture

data contains all the markers' position data frame by frame, and does not contain their orientation data.

B.2 Magnetic Motion Capture Systems

The magnetic motion capture system we used is produced by Ascension Technology Corporation. Figure B.3 shows a scene where the motion of a performer is being captured via this kind of magnetic motion capture system. This system consists of one or two transmitters from which the magnetic field is generated, and ten or more magnetic markers. The system captures the magnetic markers' position with respect to the generated magnetic field. Unfortunately, a magnetic field is easily affected by the ambient environment. For example, using a magnetic motion capture system in rebar buildings is not appropriate because the generated magnetic fields are much affected by the iron of the rebar; captured data under this condition contains too much noise. It is most appropriate to capture motion with this kind of system inside wooden buildings, or outdoors.

The marker model we used is shown in Figure B.4. Unlike an optical motion capture system, a magnetic capture system records not only the markers' 3D positions but also their spatial orientations, albeit at lower frequency (30 Hz) than the optical system (120 Hz). From magnetic marker data, we can directly calculate joint angle data even though the magnetic system does not employ many markers. In our implementation, motion capture data obtained from a magnetic motion capture system is converted to the data format of an optical motion capture system using its orientation data.

Figure B.1: Optical motion capture system. Red points are infra-red cameras, and white shining points on the human body are the optical markers.

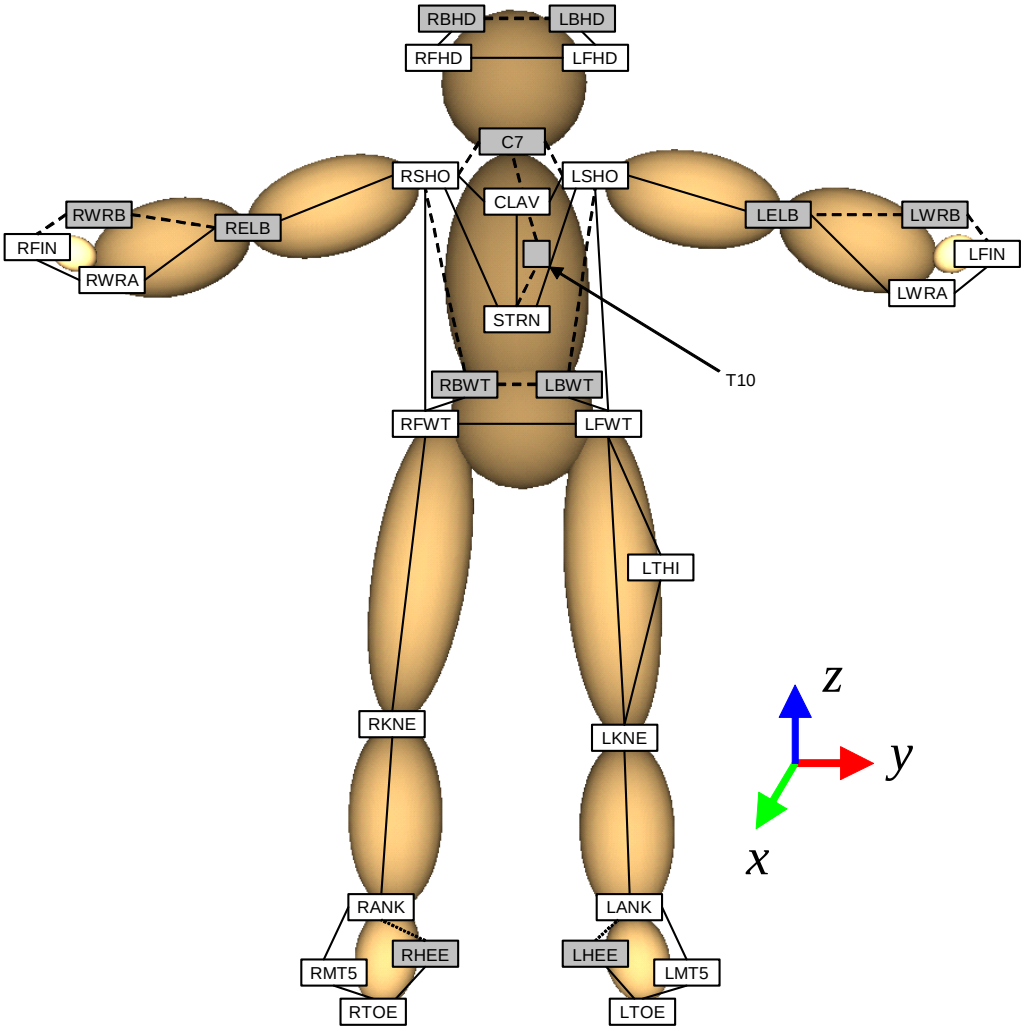


Figure B.2: Optical markers. Each box represents an optical marker with a marker name, and gray-colored boxes represent markers behind the body from this viewpoint.

Figure B.3: Magnetic motion capture system. A black box on the right hand side is a transmitter. The performer is carrying a backpack, and magnetic markers are attached to the performer's body and are connected to the backpack.

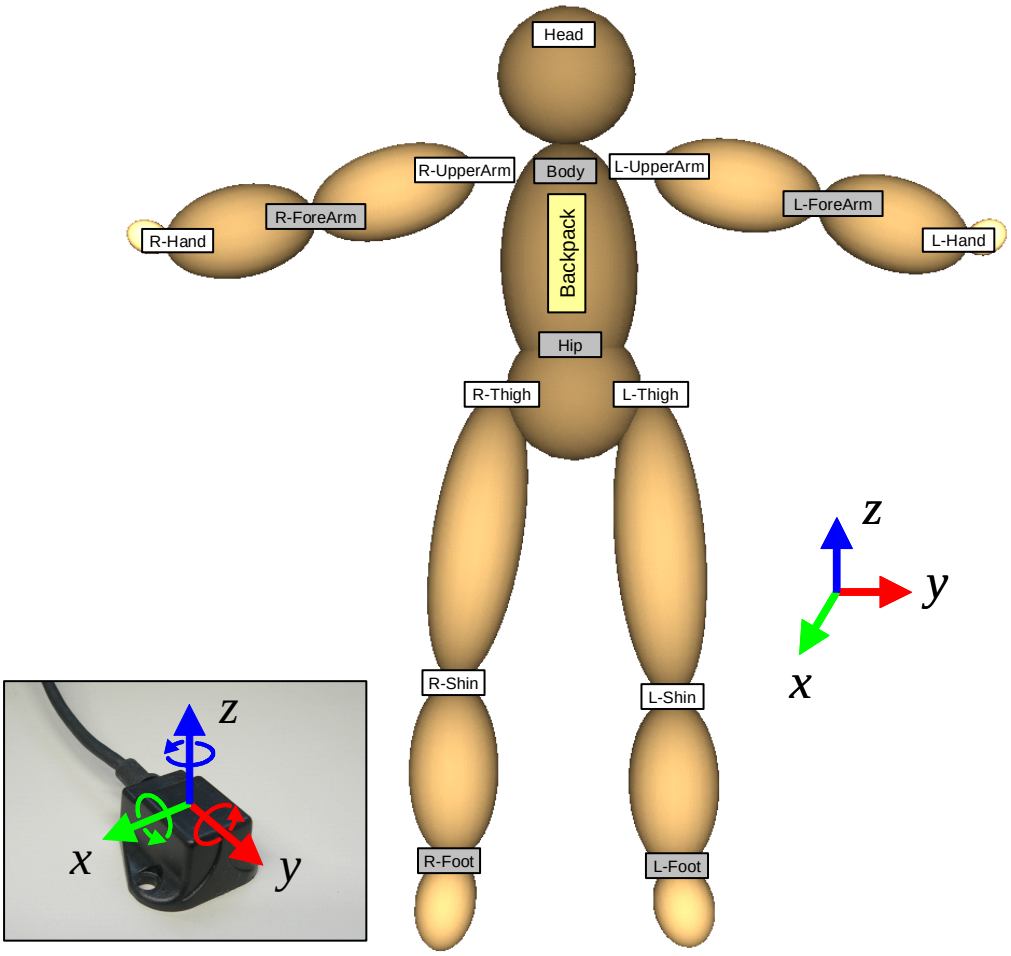


Figure B.4: Magnetic markers. Each box represents a marker with a marker name. Each captured marker data point has both 3D position and spatial orientation data as shown in the left bottom figure.

Appendix C

Calculation of Joint Angles

In Chapter 3, we focused on upper body motion. This appendix describes our inverse kinematics method to calculate joint angles in the upper body derived from marker position data. In the following, $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ denote normalized vectors representing the direction of the x-axis, y-axis, and z-axis in local coordinates, respectively, and we refer to marker position data as $\boldsymbol{p}_i$, where i is a marker label denoted in Figure B.2. We also use the notation $\boldsymbol{x}_{\text{temp}}$, $\boldsymbol{y}_{\text{temp}}$, and $\boldsymbol{z}_{\text{temp}}$ to represent temporary vectors that are not always orthogonal to each other. Additionally, we define the function that normalizes an input vector as

$$\text{Normalize}(\boldsymbol{a}) = \frac{\boldsymbol{a}}{|\boldsymbol{a}|}. \tag{C.1}$$

C.1 Calculation of Joint Angles for CG Characters

All the body links used in our hierarchical motion modification method are shown in Figure C.1. When calculating the joint angles of CG characters, we use quaternion in order to avoid a well-known gimbal lock problem. In our implementation, we first extract the local coordinates of each body link, and then calculate the rotation matrix that converts the local coordinates of the parent link to those of the currently focused link of current interest. We then convert each of the rotation matrices to quaternions as described in Appendix D. We define the function "$\boldsymbol{q}(\cdot)$" that converts a rotation matrix to a quaternion.

Body With regard to the body, first, the y-axis whose direction is from the waist toward the breast is determined and fixed, and then the x- and z-axes are

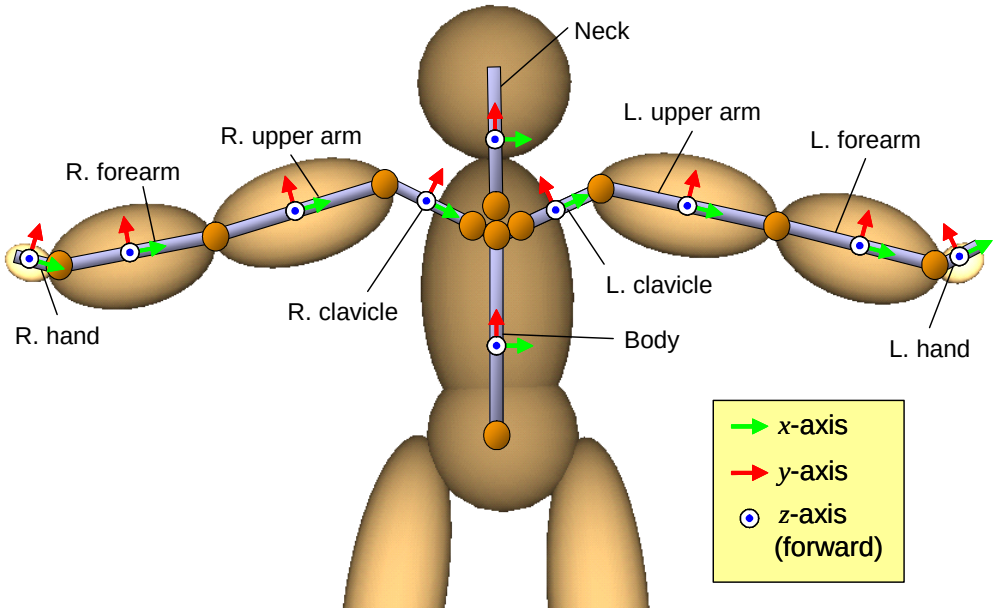


Figure C.1: Local coordinate systems of a CG character's body links.

determined:

$$y_{\mathrm{B}} = \mathrm{Normalize}\left(\frac{p_{\mathrm{CLAV}} + p_{\mathrm{C7}}}{2} - \frac{p_{\mathrm{LFWT}} + p_{\mathrm{LBWT}} + p_{\mathrm{RFWT}} + p_{\mathrm{RBWT}}}{4}\right), \tag{C.2a}$$

$$x_{\mathrm{temp}} = \left(\frac{p_{\mathrm{LFWT}} + p_{\mathrm{LBWT}}}{2} - \frac{p_{\mathrm{RFWT}} + p_{\mathrm{RBWT}}}{2}\right), \tag{C.2b}$$

$$z_{\mathrm{B}} = \mathrm{Normalize}\left(x_{\mathrm{temp}} \times y_{\mathrm{B}}\right), \tag{C.2c}$$

$$x_{\mathrm{B}} = \mathrm{Normalize}\left(y_{\mathrm{B}} \times z_{\mathrm{B}}\right). \tag{C.2d}$$

Thus, the local coordinate system of the body ${}^{0}R_{\mathrm{B}}$ is determined as

$${}^{0}R_{\mathrm{B}} = \begin{pmatrix} x_{\mathrm{B}} & y_{\mathrm{B}} & z_{\mathrm{B}} \end{pmatrix}. \tag{C.3}$$

Neck In the region of the neck, first, the y-axis whose direction is from the breast toward the head is determined and fixed, and then the z- and x-axes are

determined:

$$\boldsymbol{y}_{\mathrm{N}} = \mathrm{Normalize}\left(\frac{\boldsymbol{p}_{\mathrm{LFHD}} + \boldsymbol{p}_{\mathrm{LBHD}} + \boldsymbol{p}_{\mathrm{RFHD}} + \boldsymbol{p}_{\mathrm{RBHD}}}{4} - \frac{\boldsymbol{p}_{\mathrm{CLAV}} + \boldsymbol{p}_{\mathrm{C7}}}{2}\right), \tag{C.4a}$$

$$\boldsymbol{x}_{\mathrm{temp}} = \left(\frac{\boldsymbol{p}_{\mathrm{LFHD}} + \boldsymbol{p}_{\mathrm{LBHD}}}{2} - \frac{\boldsymbol{p}_{\mathrm{RFHD}} + \boldsymbol{p}_{\mathrm{RBHD}}}{2}\right), \tag{C.4b}$$

$$\boldsymbol{z}_{\mathrm{N}} = \mathrm{Normalize}\left(\boldsymbol{x}_{\mathrm{temp}} \times \boldsymbol{y}_{\mathrm{N}}\right), \tag{C.4c}$$

$$\boldsymbol{x}_{\mathrm{N}} = \mathrm{Normalize}\left(\boldsymbol{y}_{\mathrm{N}} \times \boldsymbol{z}_{\mathrm{N}}\right). \tag{C.4d}$$

Thus, the local coordinate system of the neck ${}^{0}R_{\mathrm{N}}$ is determined as

$${}^{0}R_{\mathrm{N}} = \begin{pmatrix} \boldsymbol{x}_{\mathrm{N}} & \boldsymbol{y}_{\mathrm{N}} & \boldsymbol{z}_{\mathrm{N}} \end{pmatrix}, \tag{C.5}$$

and the joint angle of the neck $\boldsymbol{q}_{\mathrm{N}}$ is determined as

$$\boldsymbol{q}_{\mathrm{N}} = \boldsymbol{q}\left({}^{0}R_{\mathrm{B}}^{T}\,{}^{0}R_{\mathrm{N}}\right). \tag{C.6}$$

Left collar With regard to the left clavicle, first, the x-axis whose direction is from the breast toward the left shoulder is determined and fixed, and then the y- and z-axes are determined:

$$\boldsymbol{x}_{\mathrm{LC}} = \mathrm{Normalize}\left(\boldsymbol{p}_{\mathrm{LSHO}} - \frac{\boldsymbol{p}_{\mathrm{CLAV}} + \boldsymbol{p}_{\mathrm{C7}}}{2}\right), \tag{C.7a}$$

$$\boldsymbol{y}_{\mathrm{temp}} = \left(\boldsymbol{p}_{\mathrm{RSHO}} - \frac{\boldsymbol{p}_{\mathrm{CLAV}} + \boldsymbol{p}_{\mathrm{C7}}}{2}\right), \tag{C.7b}$$

$$\boldsymbol{z}_{\mathrm{LC}} = \mathrm{Normalize}\left(\boldsymbol{x}_{\mathrm{LC}} \times \boldsymbol{y}_{\mathrm{temp}}\right), \tag{C.7c}$$

$$\boldsymbol{y}_{\mathrm{LC}} = \mathrm{Normalize}\left(\boldsymbol{z}_{\mathrm{LC}} \times \boldsymbol{x}_{\mathrm{LC}}\right). \tag{C.7d}$$

Thus, the local coordinate system of the left clavicle ${}^{0}R_{LC}$ is determined as

$${}^{0}R_{\mathrm{LC}} = \begin{pmatrix} \boldsymbol{x}_{\mathrm{LC}} & \boldsymbol{y}_{\mathrm{LC}} & \boldsymbol{z}_{\mathrm{LC}} \end{pmatrix}, \tag{C.8}$$

and the joint angle of the left collar $\boldsymbol{q}_{\mathrm{LC}}$ is determined as

$$\boldsymbol{q}_{\mathrm{LC}} = \boldsymbol{q}\left({}^{0}R_{\mathrm{B}}^{T}\,{}^{0}R_{\mathrm{LC}}\right). \tag{C.9}$$

Left shoulder With regard to the left upper arm, first, the x-axis whose direction is from the left shoulder toward the left elbow is determined and fixed. The y-axis is independently determined and fixed in order that the left elbow joint retains 1 DOF, and finally the z-axis is determined:

$$\boldsymbol{x}_{\mathrm{LUA}} = \mathrm{Normalize}\left(\boldsymbol{p}_{\mathrm{LELB}} - \boldsymbol{p}_{\mathrm{LSHO}}\right), \tag{C.10a}$$

$$\boldsymbol{y}_{\mathrm{LUA}} = \mathrm{Normalize}\left(\boldsymbol{x}_{\mathrm{LUA}} \times \left(\frac{\boldsymbol{p}_{\mathrm{LWRA}} + \boldsymbol{p}_{\mathrm{LWRB}}}{2} - \boldsymbol{p}_{\mathrm{LELB}}\right)\right), \tag{C.10b}$$

$$\boldsymbol{z}_{\mathrm{LUA}} = \mathrm{Normalize}\left(\boldsymbol{x}_{\mathrm{LUA}} \times \boldsymbol{y}_{\mathrm{LUA}}\right), \tag{C.10c}$$

Thus, the local coordinate of the left upper arm ${}^0R_{\mathrm{LUA}}$ is determined as

$$ {}^0R_{\mathrm{LUA}} = \begin{pmatrix} \boldsymbol{x}_{\mathrm{LUA}} & \boldsymbol{y}_{\mathrm{LUA}} & \boldsymbol{z}_{\mathrm{LUA}} \end{pmatrix}, \tag{C.11} $$

and the joint angle of the left shoulder $\boldsymbol{q}_{\mathrm{LS}}$ is determined as

$$ \boldsymbol{q}_{\mathrm{LS}} = \boldsymbol{q}\left({}^0R_{\mathrm{LC}}^T \, {}^0R_{\mathrm{LUA}}\right). \tag{C.12} $$

Left elbow With regard to the left forearm, first, the x-axis whose direction is from the left elbow toward the left wrist is determined and fixed. The y-axis is independently determined and fixed in order that the left elbow joint retains 1 DOF, and finally the z-axis is determined:

$$ \boldsymbol{x}_{\mathrm{LFA}} = \mathrm{Normalize}\left(\frac{\boldsymbol{p}_{\mathrm{LWRA}} + \boldsymbol{p}_{\mathrm{LWRB}}}{2} - \boldsymbol{p}_{\mathrm{LELB}}\right), \tag{C.13a} $$

$$ \boldsymbol{y}_{\mathrm{LFA}} = \mathrm{Normalize}\left(\left(\boldsymbol{p}_{\mathrm{LELB}} - \boldsymbol{p}_{\mathrm{LSHO}}\right) \times \boldsymbol{x}_{\mathrm{LFA}}\right), \tag{C.13b} $$

$$ \boldsymbol{z}_{\mathrm{LFA}} = \mathrm{Normalize}\left(\boldsymbol{x}_{\mathrm{LFA}} \times \boldsymbol{y}_{\mathrm{LFA}}\right). \tag{C.13c} $$

Thus, the local coordinate system of the left forearm ${}^0R_{\mathrm{LFA}}$ is determined as

$$ {}^0R_{\mathrm{LFA}} = \begin{pmatrix} \boldsymbol{x}_{\mathrm{LFA}} & \boldsymbol{y}_{\mathrm{LFA}} & \boldsymbol{z}_{\mathrm{LFA}} \end{pmatrix}, \tag{C.14} $$

and the joint angle of the left elbow $\boldsymbol{q}_{\mathrm{LE}}$ is determined as

$$ \boldsymbol{q}_{\mathrm{LE}} = \boldsymbol{q}\left({}^0R_{\mathrm{LUA}}^T \, {}^0R_{\mathrm{LFA}}\right). \tag{C.15} $$

Left wrist With regard to the left hand, first, the x-axis whose direction is from the left wrist toward the left finger is determined and fixed, and then the y-axis is determined to be the direction perpendicular to the back of the left hand, and the z-axis is determined:

$$ \boldsymbol{x}_{\mathrm{LH}} = \mathrm{Normalize}\left(\boldsymbol{p}_{\mathrm{LFIN}} - \frac{\boldsymbol{p}_{\mathrm{LWRA}} + \boldsymbol{p}_{\mathrm{LWRB}}}{2}\right), \tag{C.16a} $$

$$ \boldsymbol{y}_{\mathrm{LH}} = \mathrm{Normalize}\left(\left(\boldsymbol{p}_{\mathrm{LWRA}} - \boldsymbol{p}_{\mathrm{LWRB}}\right) \times \boldsymbol{x}_{\mathrm{LH}}\right), \tag{C.16b} $$

$$ \boldsymbol{z}_{\mathrm{LH}} = \mathrm{Normalize}\left(\boldsymbol{x}_{\mathrm{LH}} \times \boldsymbol{y}_{\mathrm{LH}}\right). \tag{C.16c} $$

Thus, the local coordinate system of the left hand ${}^0R_{\mathrm{LH}}$ is determined as

$$ {}^0R_{\mathrm{LH}} = \begin{pmatrix} \boldsymbol{x}_{\mathrm{LH}} & \boldsymbol{y}_{\mathrm{LH}} & \boldsymbol{z}_{\mathrm{LH}} \end{pmatrix}, \tag{C.17} $$

and the joint angle of the left wrist $\boldsymbol{q}_{\mathrm{LW}}$ is determined as

$$ \boldsymbol{q}_{\mathrm{LW}} = \boldsymbol{q}\left({}^0R_{\mathrm{LFA}}^T \, {}^0R_{\mathrm{LW}}\right). \tag{C.18} $$

Right collar With regard to the right clavicle, first, the x-axis whose direction is from the right shoulder toward the breast is determined and fixed, and then the y- and z-axes are determined:

$$\boldsymbol{x}_{\mathrm{RC}} = \mathrm{Normalize}\left(\frac{\boldsymbol{p}_{\mathrm{CLAV}} + \boldsymbol{p}_{\mathrm{C7}}}{2} - \boldsymbol{p}_{\mathrm{RSHO}}\right), \tag{C.19a}$$

$$\boldsymbol{y}_{\mathrm{temp}} = \left(\frac{\boldsymbol{p}_{\mathrm{CLAV}} + \boldsymbol{p}_{\mathrm{C7}}}{2} - \boldsymbol{p}_{\mathrm{LSHO}}\right), \tag{C.19b}$$

$$\boldsymbol{z}_{\mathrm{RC}} = \mathrm{Normalize}\left(\boldsymbol{x}_{\mathrm{RC}} \times \boldsymbol{y}_{\mathrm{temp}}\right), \tag{C.19c}$$

$$\boldsymbol{y}_{\mathrm{RC}} = \mathrm{Normalize}\left(\boldsymbol{z}_{\mathrm{RC}} \times \boldsymbol{x}_{\mathrm{RC}}\right). \tag{C.19d}$$

Thus, the local coordinate system of the right clavicle ${}^{0}R_{\mathrm{RC}}$ is determined as

$${}^{0}R_{\mathrm{RC}} = \begin{pmatrix} \boldsymbol{x}_{\mathrm{RC}} & \boldsymbol{y}_{\mathrm{RC}} & \boldsymbol{z}_{\mathrm{RC}} \end{pmatrix}, \tag{C.20}$$

and the joint angle of the right collar $\boldsymbol{q}_{\mathrm{RC}}$ is determined as

$$\boldsymbol{q}_{\mathrm{RC}} = \boldsymbol{q}\left({}^{0}R_{\mathrm{B}}^{T\,0}R_{\mathrm{RC}}\right). \tag{C.21}$$

Right shoulder With regard to the right upper arm, first, the x-axis whose direction is from the right elbow toward the right shoulder is determined and fixed. The y-axis is independently determined and fixed in order that the right elbow joint retains 1 DOF, and finally the z-axis is determined:

$$\boldsymbol{x}_{\mathrm{RUA}} = \mathrm{Normalize}\left(\boldsymbol{p}_{\mathrm{RSHO}} - \boldsymbol{p}_{\mathrm{RELB}}\right), \tag{C.22a}$$

$$\boldsymbol{y}_{\mathrm{RUA}} = \mathrm{Normalize}\left(\boldsymbol{x}_{\mathrm{RUA}} \times \left(\boldsymbol{p}_{\mathrm{RELB}} - \frac{\boldsymbol{p}_{\mathrm{RWRA}} + \boldsymbol{p}_{\mathrm{RWRB}}}{2}\right)\right), \tag{C.22b}$$

$$\boldsymbol{z}_{\mathrm{RUA}} = \mathrm{Normalize}\left(\boldsymbol{x}_{\mathrm{RUA}} \times \boldsymbol{y}_{\mathrm{RUA}}\right). \tag{C.22c}$$

Thus, the local coordinate system of the right upper arm ${}^{0}R_{\mathrm{RUA}}$ is determined as

$${}^{0}R_{\mathrm{RUA}} = \begin{pmatrix} \boldsymbol{x}_{\mathrm{RUA}} & \boldsymbol{y}_{\mathrm{RUA}} & \boldsymbol{z}_{\mathrm{RUA}} \end{pmatrix}, \tag{C.23}$$

and the joint angle of the right shoulder $\boldsymbol{q}_{\mathrm{RS}}$ is determined as

$$\boldsymbol{q}_{\mathrm{RS}} = \boldsymbol{q}\left({}^{0}R_{\mathrm{RC}}^{T\,0}R_{\mathrm{RUA}}\right). \tag{C.24}$$

Right elbow With regard to the right forearm, first, the x-axis whose direction is from the right wrist toward the right elbow is determined and fixed. The y-axis is independently determined and fixed in order that the right elbow

joint retains 1 DOF, and finally the z-axis is determined:

$$\boldsymbol{x}_{\mathrm{RFA}} = \mathrm{Normalize}\left(\boldsymbol{p}_{\mathrm{RELB}} - \frac{\boldsymbol{p}_{\mathrm{RWRA}} + \boldsymbol{p}_{\mathrm{RWRB}}}{2}\right), \tag{C.25a}$$

$$\boldsymbol{y}_{\mathrm{RFA}} = \mathrm{Normalize}\left(\boldsymbol{x}_{\mathrm{RFA}} \times \left(\boldsymbol{p}_{\mathrm{RSHO}} - \boldsymbol{p}_{\mathrm{RELB}}\right)\right), \tag{C.25b}$$

$$\boldsymbol{z}_{\mathrm{RFA}} = \mathrm{Normalize}\left(\boldsymbol{x}_{\mathrm{RFA}} \times \boldsymbol{y}_{\mathrm{RFA}}\right). \tag{C.25c}$$

Thus, the local coordinate system of the right forearm ${}^0R_{\mathrm{RFA}}$ is determined as

$${}^0R_{\mathrm{RFA}} = \left(\boldsymbol{x}_{\mathrm{RFA}} \;\; \boldsymbol{y}_{\mathrm{RFA}} \;\; \boldsymbol{z}_{\mathrm{RFA}} \right), \tag{C.26}$$

and the joint angle of the right elbow $\boldsymbol{q}_{\mathrm{RE}}$ is determined as

$$\boldsymbol{q}_{\mathrm{RE}} = \boldsymbol{q}\left({}^0R_{\mathrm{RUA}}^T \, {}^0R_{\mathrm{RFA}}\right). \tag{C.27}$$

Right wrist With regard to the right hand, first, the x-axis whose direction is from the right finger to the right wrist is determined and fixed, and then the y-axis is determined to be the direction perpendicular to the back of the right hand, and the z-axis is determined:

$$\boldsymbol{x}_{\mathrm{RH}} = \mathrm{Normalize}\left(\frac{\boldsymbol{p}_{\mathrm{RWRA}} + \boldsymbol{p}_{\mathrm{RWRB}}}{2} - \boldsymbol{p}_{\mathrm{RFIN}}\right), \tag{C.28a}$$

$$\boldsymbol{y}_{\mathrm{RH}} = \mathrm{Normalize}\left(\boldsymbol{x}_{\mathrm{RH}} \times \left(\boldsymbol{p}_{\mathrm{RWRB}} - \boldsymbol{p}_{\mathrm{RWRA}}\right)\right), \tag{C.28b}$$

$$\boldsymbol{z}_{\mathrm{RH}} = \mathrm{Normalize}\left(\boldsymbol{x}_{\mathrm{RH}} \times \boldsymbol{y}_{\mathrm{RH}}\right). \tag{C.28c}$$

Thus, the local coordinate system of the right hand ${}^0R_{\mathrm{RH}}$ is determined as

$${}^0R_{\mathrm{RH}} = \left(\boldsymbol{x}_{\mathrm{RH}} \;\; \boldsymbol{y}_{\mathrm{RH}} \;\; \boldsymbol{z}_{\mathrm{RH}} \right), \tag{C.29}$$

and the joint angle of the right wrist $\boldsymbol{q}_{\mathrm{RW}}$ is determined as

$$\boldsymbol{q}_{\mathrm{RW}} = \boldsymbol{q}\left({}^0R_{\mathrm{RFA}}^T \, {}^0R_{\mathrm{RW}}\right). \tag{C.30}$$

C.2 Calculation of Joint Angles for the HRP-2

Each joint of the HRP-2 has a 1-DOF actuator; all the joint angles are represented in Roll-Pitch-Yaw format. We calculate each joint angle from a body link's direction in its local coordinate system. Each local coordinate of the HRP-2's arm joint is illustrated in Figure C.2. We use the function "atan2(x, y)" that returns $\tan^{-1}(x/y)$ within the range $(-\pi, \pi)$.

Local coordinate system of chest Since the HRP-2's torso consists of a rigid-body chest and a rigid-body lower chest, or loins, first the local coordinate system of the chest ${}^0R_1$ is determined as

$$\boldsymbol{y} = \text{Normalize}\left(\boldsymbol{p}_{\text{LSHO}} - \boldsymbol{p}_{\text{RSHO}}\right), \tag{C.31a}$$

$$\boldsymbol{z}_{\text{temp}} = \frac{\boldsymbol{p}_{\text{LSHO}} + \boldsymbol{p}_{\text{RSHO}}}{2} - \frac{\boldsymbol{p}_{\text{LFWT}} + \boldsymbol{p}_{\text{RFWT}} + \boldsymbol{p}_{\text{LBWT}} + \boldsymbol{p}_{\text{RBWT}}}{4}, \tag{C.31b}$$

$$\boldsymbol{x} = \text{Normalize}\left(\boldsymbol{y} \times \boldsymbol{z}_{\text{temp}}\right), \tag{C.31c}$$

$$\boldsymbol{z} = \text{Normalize}\left(\boldsymbol{x} \times \boldsymbol{y}\right), \tag{C.31d}$$

$${}^0R_1 = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}). \tag{C.31e}$$

Left shoulder pitch Rotation of the left shoulder in pitch is about the y-axis of the chest local coordinate system. The direction vector of the upper arm in the global coordinate system $\boldsymbol{v}_0$ is calculated as

$$\boldsymbol{v}_0 = \boldsymbol{p}_{\text{LELB}} - \boldsymbol{p}_{\text{LSHO}}\,, \tag{C.32}$$

and the direction in the chest local coordinate system $\boldsymbol{v}_1$ is calculated as

$$\boldsymbol{v}_1 = {}^0R_1^T\,\boldsymbol{v}_0\,. \tag{C.33}$$

The joint angle representing pitch rotation of the left shoulder θ_{LSP} is therefore determined as

$$\theta_{\text{LSP}} = \text{atan2}\left(-v_{1x}\,, -v_{1z}\right). \tag{C.34}$$

Left shoulder roll Rotation of the left shoulder in roll is about the x-axis of the local coordinate system of the left shoulder pitch. The direction vector of the upper arm in the local coordinate system of the left shoulder pitch $\boldsymbol{v}_2$ is calculated as

$$\boldsymbol{v}_2 = R_y\left(\theta_{\text{LSP}}\right)\,\boldsymbol{v}_1\,, \tag{C.35}$$

where $R_y\left(\theta_{\text{LSP}}\right)$ represents the matrix that rotates θ_{LSP} degrees around the y-axis of the chest local coordinate system. The joint angle representing roll rotation of the left shoulder θ_{LSR} is therefore determined as

$$\theta_{\text{LSR}} = \text{atan2}\left(v_{2y}\,, -v_{2z}\right). \tag{C.36}$$

Left shoulder yaw Rotation of the left shoulder in yaw is about the z-axis of the local coordinate system of the left shoulder roll. The direction vector of the forearm in the global coordinate system $\boldsymbol{v}_3'$ is calculated as

$$\boldsymbol{v}_3' = \frac{\boldsymbol{p}_{\text{LWRA}} + \boldsymbol{p}_{\text{LWRB}}}{2} - \boldsymbol{p}_{\text{LELB}}\,. \tag{C.37}$$

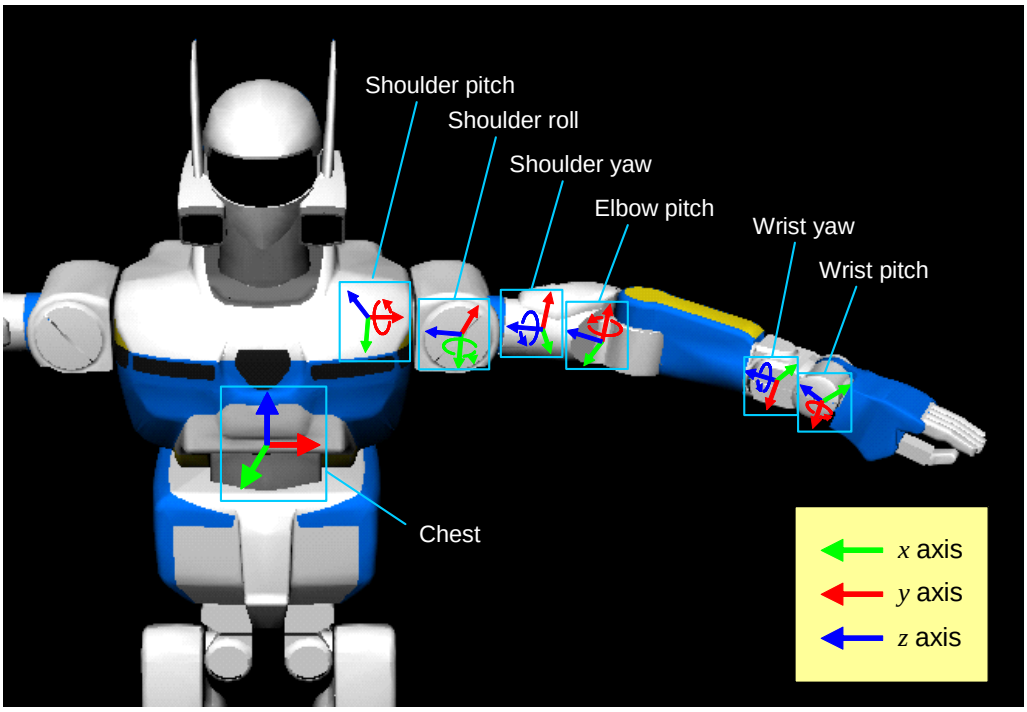


Figure C.2: Local coordinate systems of the HRP-2's left upper body. Axes surrounded by rotation arrows represent the central axes of the rotation relative to parent joints.

This vector is converted into the local coordinate system of the left shoulder roll as

$$v_3 = R_x(\theta_{\mathrm{LSR}})\, R_y(\theta_{\mathrm{LSP}})\, {}^0R_1^T\, v_3' \,, \tag{C.38}$$

where $R_x(\theta_{\mathrm{LSR}})$ represents the matrix that rotates θ_{LSR} degrees around the x-axis of the local coordinate system of the left shoulder roll. The joint angle representing yaw rotation of the left shoulder θ_{LSY} is therefore determined as

$$\theta_{\mathrm{LSY}} = \mathrm{atan2}\left(-v_{3y}\,, v_{3x}\right). \tag{C.39}$$

Note that when the arm is stretched all the way out, the atan2 function becomes singular so the joint angle cannot be computed. This problem does occur, but we solve it by assigning the answer as being the joint angle of the previous frame.

Left elbow pitch Rotation of the left elbow in pitch is about the y-axis of the local coordinate system of the left shoulder yaw. The direction vector of the forearm in the local coordinate system of the left shoulder yaw $\boldsymbol{v}_4$ is calculated as

$$\boldsymbol{v}_4 = R_z\left(\theta_{\mathrm{LSY}}\right)\boldsymbol{v}_3\,. \tag{C.40}$$

where $R_z\left(\theta_{\mathrm{LSY}}\right)$ represents the matrix that rotates θ_{LSY} degrees around the z-axis of the local coordinate system of the left shoulder yaw. The joint angle representing pitch rotation of the left should elbow θ_{LEP} is therefore determined as

$$\theta_{\mathrm{LEP}} = \mathrm{atan2}\left(-v_{4x}\,, v_{4z}\right). \tag{C.41}$$

Left wrist yaw Rotation of the left wrist in yaw is about the z-axis of the local coordinate system of the left elbow pitch. The sideward direction vector of the wrist in the global coordinate system $\boldsymbol{v}'_5$ is calculated as

$$\boldsymbol{v}'_5 = \boldsymbol{p}_{\mathrm{LWRA}} - \boldsymbol{p}_{\mathrm{LWRB}}. \tag{C.42}$$

This vector is converted into the local coordinate system of the left elbow pitch as

$$\boldsymbol{v}_5 = R_y\left(\theta_{\mathrm{LEP}}\right)R_z\left(\theta_{\mathrm{LSY}}\right)R_x\left(\theta_{\mathrm{LSR}}\right)R_y\left(\theta_{\mathrm{LSP}}\right){}^0R_1^T\,\boldsymbol{v}'_5, \tag{C.43}$$

where $R_y\left(\theta_{\mathrm{LEP}}\right)$ represents the matrix that rotates θ_{LEP} degrees around the y-axis of the local coordinate system of the left elbow pitch. The joint angle representing yaw rotation of the left wrist θ_{LWY} is therefore determined as

$$\theta_{\mathrm{LWY}} = \mathrm{atan2}\left(-v_{5x}\,, v_{5z}\right). \tag{C.44}$$

Left wrist pitch Rotation of the left wrist in pitch is about the y-axis of the local coordinate system of the left wrist yaw. The forward direction vector of the hand in the global coordinate system $\boldsymbol{v}'_6$ is calculated as

$$\boldsymbol{v}'_6 = \boldsymbol{p}_{\mathrm{LFIN}} - \frac{\boldsymbol{p}_{\mathrm{LWRA}} + \boldsymbol{p}_{\mathrm{LWRB}}}{2}. \tag{C.45}$$

This vector is converted into the local coordinate system of the left wrist yaw as

$$\boldsymbol{v}_6 = R_z\left(\theta_{\mathrm{LWY}}\right)R_y\left(\theta_{\mathrm{LEP}}\right)R_z\left(\theta_{\mathrm{LSY}}\right)R_x\left(\theta_{\mathrm{LSR}}\right)R_y\left(\theta_{\mathrm{LSP}}\right){}^0R_1^T\,\boldsymbol{v}'_6, \tag{C.46}$$

where $R_z\left(\theta_{\mathrm{LWY}}\right)$ represents the matrix that rotates θ_{LWY} degrees around the z-axis of the local coordinate system of the left elbow pitch. The joint angle representing pitch rotation of the left wrist θ_{LWP} is therefore determined as

$$\theta_{\mathrm{LWP}} = \mathrm{atan2}\left(-v_{6x}\,, -v_{6z}\right). \tag{C.47}$$

Joint angles of the right arm are determined in a similar way.

Appendix D

Quaternions for Rotation Representation

In Chapter 3, we used a quaternion to represent joint angle rotation. This appendix explains the use of quaternions for representation of the object rotation in detail.

D.1 Definition of a Quaternion

A quaternion is a non-commutative extension of a complex number. While a complex number consists of a real part and an imaginary part, a quaternion consists of a real part and three imaginary parts In terms of the elements i, j, and k, a quaternion $\boldsymbol{q}$ is represented as

$$\boldsymbol{q} = w + xi + yj + zk \tag{D.1}$$

$$= \begin{pmatrix} w \\ \boldsymbol{v} \end{pmatrix}, \tag{D.2}$$

where

$$\boldsymbol{v} \equiv (x, y, z)^T, \tag{D.3}$$

and w, x, y, and z are real numbers. In the following, we use Equation (D.2) to represent a quaternion. Its imaginary elements i, j, and k have the following fundamental relationships:

$$\begin{aligned} i^2 = j^2 = k^2 = ijk = -1, \\ ij = k, \quad ji = k, \\ jk = i, \quad kj = -i, \\ ki = j, \quad ik = -j. \end{aligned} \tag{D.4}$$

The conjugate quaternion $\boldsymbol{q}^*$ is defined as follows:

$$\boldsymbol{q}^* = \begin{pmatrix} w \\ -\boldsymbol{v} \end{pmatrix}. \tag{D.5}$$

D.2 Quaternion Operation

The addition of two quaternions $\boldsymbol{q}_1 = (w_1, \boldsymbol{v}_1)$ and $\boldsymbol{q}_2 = (w_2, \boldsymbol{v}_2)$ is equivalent to summing the elements together:

$$\boldsymbol{q}_1 + \boldsymbol{q}_2 = \begin{pmatrix} w_1 + w_2 \\ \boldsymbol{v}_1 + \boldsymbol{v}_2 \end{pmatrix}. \tag{D.6}$$

The subtraction of quaternions is defined as

$$\begin{aligned} \boldsymbol{q}_1 - \boldsymbol{q}_2 &= \boldsymbol{q}_1 + (-\boldsymbol{q}_2) \\ &= \begin{pmatrix} w_1 - w_2 \\ \boldsymbol{v}_1 - \boldsymbol{v}_2 \end{pmatrix}. \end{aligned} \tag{D.7}$$

The multiplication of quaternions, which is non-commutative, is termed the *Grassman product*:

$$\boldsymbol{q}_1\boldsymbol{q}_2 = \begin{pmatrix} w_1w_2 - \boldsymbol{v}_1 \cdot \boldsymbol{v}_2 \\ w_1\boldsymbol{v}_1 + w_2\boldsymbol{v}_2 + \boldsymbol{v}_1 \times \boldsymbol{v}_2 \end{pmatrix}. \tag{D.8}$$

The norm of a quaternion $|\boldsymbol{q}|$ is defined as

$$\begin{aligned} |\boldsymbol{q}| &= \sqrt{\boldsymbol{q}\boldsymbol{q}^*} \\ &= \sqrt{w^2 + \boldsymbol{v} \cdot \boldsymbol{v}}. \end{aligned} \tag{D.9}$$

Using the norm, an inverse quaternion $\boldsymbol{q}^{-1}$ is defined as

$$\boldsymbol{q}^{-1} = \frac{\boldsymbol{q}^*}{|\boldsymbol{q}|^2}. \tag{D.10}$$

D.3 Rotation Representation Using Quaternions

Consider the case when the 3D point $\boldsymbol{p}$ is rotated around an axis represented as a normalized vector $\boldsymbol{n} = (n_1, n_2, n_3)^T$ by θ radians, and is thereby moved to a new position $\boldsymbol{p}'$. In geometrical terms, the relationship between $\boldsymbol{p}$ and $\boldsymbol{p}'$ is represented as

$$\boldsymbol{p}' = (\boldsymbol{p} \cdot \boldsymbol{n})\boldsymbol{n} + (\boldsymbol{p} - (\boldsymbol{p} \cdot \boldsymbol{n})\boldsymbol{n}) \cos\theta + \boldsymbol{n} \times \boldsymbol{p} \sin\theta, \tag{D.11}$$

This can be written as a matrix equation:

$$\boldsymbol{p}' = (I\cos\theta + \boldsymbol{n}\boldsymbol{n}^T(1-\cos\theta) + A_{\boldsymbol{n}}\sin\theta)\boldsymbol{p}, \tag{D.12}$$

where

$$A = \begin{pmatrix} 0 & n_3 & -n_2 \\ -n_3 & 0 & n_1 \\ n_2 & -n_1 & 0 \end{pmatrix}, \tag{D.13}$$

and I is a 3D identity matrix.

Let $\boldsymbol{q} = (c, \boldsymbol{u})^T$ be a unit quaternion. We can define the following equation:

$$\begin{aligned} \boldsymbol{q}\begin{pmatrix}0\\ \boldsymbol{p}\end{pmatrix}\boldsymbol{q}^* &= \begin{pmatrix}c\\ \boldsymbol{u}\end{pmatrix}\begin{pmatrix}0\\ \boldsymbol{p}\end{pmatrix}\begin{pmatrix}c\\ -\boldsymbol{u}\end{pmatrix} \\ &= \begin{pmatrix}0\\ -\boldsymbol{u}\times(\boldsymbol{u}\times\boldsymbol{p}) + 2c(\boldsymbol{u}\times\boldsymbol{p}) + c^2\boldsymbol{p} + (\boldsymbol{p}\cdot\boldsymbol{u})\boldsymbol{u}\end{pmatrix} \\ &= \begin{pmatrix}0\\ (c^2 - \boldsymbol{u}\cdot\boldsymbol{u})\boldsymbol{p} + 2(\boldsymbol{p}\cdot\boldsymbol{u})\boldsymbol{u} + 2c(\boldsymbol{u}\times\boldsymbol{p})\end{pmatrix}. \end{aligned} \tag{D.14}$$

Because $\boldsymbol{q}$ is a unit quaternion, the following is true:

$$\boldsymbol{q} = \begin{pmatrix}\cos\theta\\ \boldsymbol{n}\sin\theta\end{pmatrix}, \tag{D.15}$$

and Equation (D.14) can therefore be modified as:

$$\boldsymbol{q}\begin{pmatrix}0\\ \boldsymbol{p}\end{pmatrix}\boldsymbol{q}^* = \begin{pmatrix}0\\ \boldsymbol{p}\cos 2\theta + (1-\cos 2\theta)(\boldsymbol{n}\cdot\boldsymbol{p})\boldsymbol{n} + (\boldsymbol{n}\times\boldsymbol{p})\sin 2\theta\end{pmatrix} \tag{D.16}$$

From Equation (D.12) and Equation (D.16), a unit quaternion represented as

$$\boldsymbol{q} = \begin{pmatrix}\cos(\theta/2)\\ \boldsymbol{n}\sin(\theta/2)\end{pmatrix} \tag{D.17}$$

can be interpreted as the rotation of angle θ around the axis $\boldsymbol{n}$. The following relationship between a rotation matrix and a quaternion can be also obtained:

$$R = \begin{pmatrix} s^2+u^2-w^2-v^2 & 2(uv-sw) & 2(sv+uw) \\ 2(sw+uv) & s^2+v^2-u^2-w^2 & 2(vw-su) \\ 2(uw-sv) & 2(su+vw) & s^2+w^2-v^2-u^2 \end{pmatrix}, \tag{D.18}$$

where a unit quaternion is represented as $(s, u, v, w)^T$.

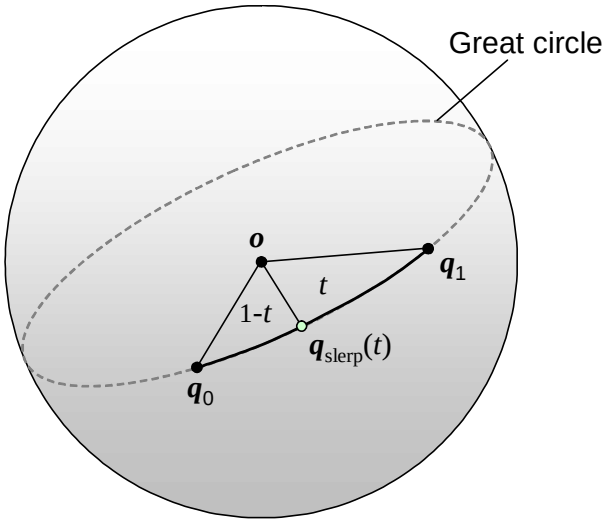


Figure D.1: Conceptual illustration of SLERP calculation.

Because Equation (D.17) is quite similar to Euler's formula for a complex number:

$$\exp(i\theta) = \cos\theta + i\sin\theta, \tag{D.19}$$

we can obtain the following relationship by extending Euler's formula to a quaternion:

$$\exp(\boldsymbol{n}\theta) = \cos\theta + \boldsymbol{n}\sin\theta. \tag{D.20}$$

Therefore, we can also define the following relationship:

$$\log(\cos\theta + \boldsymbol{n}\sin\theta) = \boldsymbol{n}\theta \in \mathbb{R}^3. \tag{D.21}$$

D.4 Spherical Linear Interpolation

Spherical linear interpolation, called *SLERP*, is a famous interpolation technique in computer graphics. SLERP in quaternion space was introduced by Shoemake [Sho85] for the purpose of animating 3D rotation; it calculates the interpolated position between two points that are on a 4D unit sphere along a great circle arc of the sphere.

Let $\boldsymbol{q}_0$ and $\boldsymbol{q}_1$ be quaternions located on the 4D unit sphere, and $t \in [0, 1]$ be an interpolation parameter. SLERP is defined as

$$SLERP\left(\boldsymbol{q}_0, \boldsymbol{q}_1; t\right) = \boldsymbol{q}_0 \left(\boldsymbol{q}_0^{-1} \boldsymbol{q}_1\right)^t . \tag{D.22}$$

Using Equation (D.21), Equation (D.22) is referred to as

$$\begin{aligned} \log(\boldsymbol{q}_{\text{slerp}}(t)) &= \log\left(\boldsymbol{q}_0 \left(\boldsymbol{q}_0^{-1} \boldsymbol{q}_1\right)^t\right) \\ &= \log\left(\exp\left(\theta_0 \boldsymbol{v}_0\right) \left(\exp\left(-\theta_0 \boldsymbol{v}_0\right) \exp\left(\theta_1 \boldsymbol{v}_1\right)\right)^t\right) \\ &= \log\left(\exp\left((1-t)\theta_0 \boldsymbol{v}_0\right) \exp\left(t\theta_1 \boldsymbol{v}_1\right)\right) \\ &= (1-t)\left(\theta_0 \boldsymbol{v}_0\right) + t\left(\theta_1 \boldsymbol{v}_1\right). \end{aligned} \tag{D.23}$$

where $\boldsymbol{q}_0 = \exp\left(\theta_0 \boldsymbol{v}_0\right)$ and $\boldsymbol{q}_1 = \exp\left(\theta_1 \boldsymbol{v}_1\right)$. Accordingly, SLERP in quaternion space is simply calculated as the simple linear interpolation between two 3D vectors.

References

[ABB05] Alankus G., Bayazit A. A., Bayazit O. B.: Automated motion synthesis for dancing characters. *Computer Animation and Virtual Worlds Vol. 16*, No. 3-4 (2005), 259–271.

[ACC05] Assa J., Caspi Y., Cohen-Or D.: Action synopsis: Pose selection and illustration. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2005) Vol. 24*, No. 3 (2005), 667–676.

[AF02] Arikan O., Forsyth D. A.: Interactive motion generation from examples. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2002) Vol. 21*, No. 3 (2002), 483–490.

[AFO03] Arikan O., Forsyth D. A., O' Brien J. F.: Motion synthesis from annotations. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2003) Vol. 22*, No. 3 (2003), 402–408.

[ALP06] Abe Y., Liu C. K., Popović Z.: Momentum-based parameterization of dynamic character motion. *Graphical Models Vol. 68*, No. 2 (2006), 194–211.

[Ari06] Arikan O.: Compression of motion capture databases. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2006) Vol. 24*, No. 3 (2006), 890–897.

[Auta] Autodesk Maya: http://www.autodesk.com/maya.

[Autb] Autodesk MotionBuilder: http://www.autodesk.com/motionbuilder.

[BA83] Burt P. J., Adelson E. H.: The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications COM-31*, No. 4 (1983), 532–540.

[BAHH92] Bergen J. R., Anandan P., Hanna K. J., Hingorani R.: Hierarchical model-based motion estimation. In *Proceedings of European Conference on Computer Vision* (1992), pp. 237–252. Lecture Notes in Computer Science, Vol. 588.

[BFB94] Barron J. L., Fleet D. J., Beauchemin S. S.: Performance of optical flow techniques. *International Journal of Computer Vision Vol. 12*, No. 1 (1994), 43–77.

[BH00] Brand M. E., Hertzmann A.: Style machines. In *Proceedings of ACM SIGGRAPH 2000* (2000), pp. 402–408.

[Bre90] Bregman A. S.: *Auditory Scene Analysis: The Perceptual Organization of sound*. The MIT Press, 1990.

[Bro90] Brown J. C.: Calculation of a constant Q spectral transform. *Journal of Acoustic Society of America Vol. 89*, No. 1 (1990), 425–434.

[BSP*04] Barbič J., Safonova A., Pan J., Faloutsos C., Hodgins J. K., Pollard N. S.: Segmenting motion capture data into distinct behaviors. In *Proceedings of Graphics Interface* (2004), pp. 185–194.

[BW95] Bruderlin A., Williams L.: Motion signal processing. In *Proceedings of ACM SIGGRAPH 95* (1995), pp. 97–104.

[CAMG06] Caspi Y., Axelrod A., Matsushita Y., Gamliel A.: Dynamic stills and clip trailers. *The Visual Computer (Pacific Graphics 2006 Conference Proceedings) Vol. 22*, No. 9 (2006), 642–652.

[CB93] Cooke M., Brown G.: Computational auditory scene analysis: Exploiting principles of perceived continuity. *Speech Communication Vol. 13* (1993), 391–399.

[CCYL04] Chao S., Chiu C., Yang S., Lin T.: Tai chi synthesizer: A motion synthesis framework based on key-postures and motion instructions. *Computer Animation and Virtual Worlds Vol. 15*, No. 3-4 (2004), 259–268.

[CCZB00] Chi D., Costa M., Zhao L., Badler N.: The EMOTE model for Effort and Shape. In *Proceedings of ACM SIGGRAPH 2000* (2000), pp. 173–182.

[CGM*06] Chalodhorn R., Grimes D. B., Maganis G. Y., Rao R. P. N., Asada M.: Learning humanoid motion dynamics through sensory-motor mapping in reduced dimensional spaces. In *Proceedings of IEEE International Conference on Robotics and Automation* (2006), pp. 3693–3698.

[CH05] Chai J., Hodgins J. K.: Performance animation from low-dimensional control signals. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2005) Vol. 24*, No. 3 (2005), 686–696.

[Che01] Chew E.: Modeling tonality: Applications to music cognition. In *Proceedings of Annual Conference of the Cognitive Science Society* (2001), pp. 206–211.

[CKDH01] Cemgil A. T., Kappen B., Desian P., Honing H.: On tempo tracking: Tempogram representation and Kalman filtering. *Journal of New Music Research Vol. 29*, No. 4 (2001), 259–273.

[CMU] CMU Graphics Lab Motion Capture Database: http://mocap.cs.cmu.edu/.

[CON05] Chen B., Ono Y., Nishita T.: Character animation creation using hand-drawn sketches. *The Visual Computer (Pacific Graphics 2005 Conference Proceedings) Vol. 21*, No. 8-10 (2005), 551–558.

[DH89] Desain P., Honing H.: The quantization of musical time: A connectionist approach. *Computer Music Journal Vol. 13*, No. 3 (1989), 56–66.

[DH94] Desain P., Honing H.: Advanced issues in beat induction modeling: Syncopation, tempo and timing. In *Proceedings of International Computer Music Conference* (1994), pp. 92–94.

[DSJQ04] Dyaberi V. M., Sundaram H., James J., Qian G.: Phrase structure detection in dance. In *Proceedings of ACM Multimedia* (2004), pp. 332–335.

[FF05] Forbes K., Fiume E.: An efficient search algorithm for motion data using weighted PCA. In *Proceedings of ACM SIGGRAPH/ Eurographics Symposium on Computer Animation* (2005), pp. 67–76.

[FH85] Flash T., Hogan H.: The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience Vol. 5* (1985), 1688–170.

[FvdPT01] Faloutsos P., van de Panne M., Terzopoulos D.: Composable controllers for physics-based character animation. In *Proceedings of ACM SIGGRAPH 2001* (2001), pp. 251–260.

[GBT04] Glardon P., Boulic R., Thalmann D.: PCA-based walking engine using motion capture data. In *Proceedings of Computer Graphics International* (2004), pp. 292–298.

[GH04] Goto M., Hirata K.: Recent studies on music information processing. *Acoustical Science and Technology Vol. 25*, No. 6 (2004), 419–425.

[Gle98] Gleicher M.: Retargetting motion to new characters. In *Proceedings of ACM SIGGRAPH 98* (1998), pp. 33–42.

[GMHP04] Grochow K., Martin S. L., Hertzmann A., Popović Z.: Style-based inverse kinematics. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2004) 23*, 3 (2004), 522–531.

[Got01] Goto M.: An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research Vol. 30*, No. 2 (2001), 159–171.

[GW02] Gonzalez R. C., Woods R. E.: *Digital Image Processing*. Prentice Hall, 2002.

[Har78] Harris F. J.: On the use of windows for harmonic analysis with discrete Fourier transform. *Proceedings of IEEE Bd. 66* (1978), 51–83.

[HB95] Heeger D. J., Bergen J. R.: Pyramid-based texture analysis/synthesis. In *Proceedings of ACM SIGGRAPH 1995* (1995), pp. 229–238.

[HG06] Heck R., Gleicher M.: Parametric motion graphs. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation (Posters and Demos)* (2006), pp. 19–20.

[HGP04] Hsu E., Gentry S., Popović J.: Example-based control of human motion. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), pp. 69–77.

[HKG06] Heck R., Kovar L., Gleicher M.: Splicing upper-body actions with locomotion. *Computer Graphics Forum (Proceedings of Eurographics 2006) Vol. 25*, No. 3 (2006), 219–227.

[HN01] Hachimura K., Nakamura M.: Method of generating coded description of human body motion from motion-captured data. In *Proceedings of IEEE International Workshop on Robot and Human Interactive Communication* (2001), pp. 122–127.

[HPP05] Hsu E., Pulli K., Popović J.: Style translation for human motion. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2005) Vol. 24*, No. 3 (2005), 1082–1089.

[Hut77] Hutchinson A.: *Labanotation*. Theater Arts Book, 1977.

[IF04] Ikemoto L., Forsyth D. A.: Enriching a motion collection by transplanting limbs. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), pp. 99–108.

[IMH05] Igarashi T., Moscovich T., Hughes J. F.: Spatial keyframing for performance-driven animation. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), pp. 107–115.

[ITTN04] Inamura T., Toshima I., Tanie H., Nakamura Y.: Embodied symbol emergence based on mimesis theory. *International Journal of Robotics Research Vol. 23*, No. 4 (2004), 363–377.

[JM02] Jenkins O. C., Matarić M. J.: Deriving action and behavior primitives from human motion data. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (2002), pp. 2551–2556.

[Kai67] Kailath T.: The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology COM-15* (1967), 52–60.

[KG02] Kovar L., Gleicher M.: Footskate cleanup for motion capture editing. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2002), pp. 97–104.

[KG03] Kovar L., Gleicher M.: Flexible automatic motion blending with registration curves. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), pp. 214–224.

[KG04] Kovar L., Gleicher M.: Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2004) Vol. 23*, No. 3 (2004), 559–568.

[KGP02] Kovar L., Gleicher M., Pighin F.: Motion graphs. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2002) Vol. 21*, No. 3 (2002), 473–482.

[KHN02] Kojima K., Hachimura K., Nakamura M.: Labaneditor: Graphical editor for dance notation. In *Proceedings of IEEE International Workshop on Robot and Human Interactive Communication* (2002), pp. 59–64.

[KI93] Kang S. B., Ikeuchi K.: Toward automatic robot instruction from perception – recognizing a grasp from observation. *IEEE Transactions on Robotics and Automation Vol. 9*, No. 4 (1993), 432–443.

[KII88] Katayose H., Imai M., Inokuchi S.: Sentiment extraction in music. In *Proceedings of International Conference on Pattern Recognition* (1988), vol. 2, pp. 1083–1087.

[KKK*02] Kaneko K., Kanehiro F., Kajita S., Yokoyama K., Akachi K., Kawasaki T., Ota S., Isozumi T.: Design of prototype humanoid robotics platform for HRP. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (2002), pp. 2431–2436.

[KPS03] Kim T., Park S. I., Shin S. Y.: Rhythmic-motion synthesis based on motion-beat analysis. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2003) 22*, 3 (2003), 392–401.

[KTP03] Kahol K., Tripathi P., Panchanathan S.: Gesture segmentation in complex motion sequences. In *Proceedings of IEEE International Conference on Image Processing* (2003), vol. 2, pp. 105–108.

[KTP04] Kahol K., Tripathi P., Panchanathan S.: Automated gesture segmentation from dance sequences. In *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition* (2004), pp. 883–888.

[KTP06] Kahol K., Tripathi P., Panchanathan S.: Documenting motion sequences: Development of a personalized annotation system. *IEEE Multimedia Magazine Vol. 13*, No. 1 (2006), 37–45.

[LC00] Logan B., Chu S.: Music summarization using key phrases. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing* (2000), pp. II–749 – II–752.

[LCF05] Lai Y., Chenney S., Fan S.: Group motion graphs. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), pp. 281–290.

[LCL06] Lee K. H., Choi M. G., Lee J.: Motion patches: Building blocks for virtual environments annotated with motion data. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2006) Vol. 25*, No. 3 (2006), 898–906.

[LCR*02] Lee J., Chai J., Reitsma P. S. A., Hodgins J. K., Pollard N. S.: Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2002) Vol. 21*, No. 3 (2002), 491–500.

[LH74] Lawson C. L., Hanson R. J.: *Solving Least Squares Problems*. Prentice Hall, 1974.

[LK94] Large E. W., Kolen J. F.: Resonance and the perception of musical meter. *Connection Science Vol. 6*, No. 1 (1994), 64–76.

[LL05] Lee H.-C., Lee I.-K.: Automatic synchronization of background music and motion in computer animation. *Computer Graphics Forum (Proceedings of Eurographics 2005) Vol. 24*, No. 3 (2005), 353–361.

[LLZ06] Lu L., Liu D., Zhang H.: Automatic mood detection and tracking of music audio signals. *IEEE Transactions on Audio, Speech and Language Processing Vol. 14*, No. 1 (2006), 5–18.

[LM06] Liu G., McMillan L.: Segment-based human motion compression. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), pp. 127–135.

[LS99] Lee J., Shin S. Y.: A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of ACM SIGGRAPH 99* (1999), pp. 39–48.

[LS01] Lee J., Shin S. Y.: A coordinate-invariant approach to multiresolution motion analysis. *Graphical Models Vol. 63*, No. 2 (2001), 87–105.

[LU60] Laban R., Ullmann L.: *Mastery of Movement*. Princeton Book Company Publishers, 1960.

[LWS97] Lee S., Wolberg G., Shin S. Y.: Scattered data interpolation with multilevel B-splines. *IEEE Transactions on Visualization and Computer Graphics Vol. 3*, No. 3 (1997), 228–244.

[LWS02] Li Y., Wang T., Shum H.-Y.: Motion texture: a two-level statistical model for character motion synthesis. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2002) Vol. 21*, No. 3 (2002), 465–472.

[LZ03] Lu L., Zhang H.-J.: Automated extraction of music snippets. In *Proceedings of ACM Multimedia* (2003), pp. 140–147.

[LZWM06] Liu G., Zhang J., Wang W., McMillan L.: Human motion estimation from a reduced marker set. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2006), pp. 35–42.

[MBSL99] Malik J., Belongie S., Shi J., Leung T.: Texton, contours and regions: Cue integration in imagesegmentation. In *Proceedings of IEEE International Conference on Computer Vision* (1999), pp. 918–925.

[Mit98] Mitra S. K.: *Digital Signal Processing*. McGraw Hill, 1998.

[MK05] Mukai T., Kuriyama S.: Geostatistical motion interpolation. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2005) Vol. 24*, No. 3 (2005), 1062–1070.

[MK06] Mukai T., Kuriyama S.: Multilinear motion synthesis using geostatistics. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation (Posters and Demos)* (2006), pp. 21–22.

[MPS06] McCann J., Pollard N. S., Srinivasa S.: Physics-based motion retiming. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), pp. 205–214.

[MR06] Müller M., Röder T.: Motion templates for automatic classification and retrieval of motion capture data. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), pp. 137–146.

[MRC05] Müller M., Röder T., Clausen M.: Efficient content-based retrieval of motion capture data. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2005) 24*, 3 (2005), 677–685.

[MZF06] Majkowska A., Zordan V. B., Faloutsos P.: Automatic splicing for hand and body animations. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), pp. 309–316.

[NMS02] Nakata T., Mori T., Sato T.: Analysis of impression of robot bodily expression. *Journal of Robotics and Mechatronics Vol. 14*, No. 1 (2002), 27–36.

[NNI03] Nakazawa A., Nakaoka S., Ikeuchi K.: Synthesize stylistic human motion from examples. In *Proceedings of IEEE International Conference on Robotics and Automation* (2003), pp. 3905–3910.

[NNI04] Nakazawa A., Nakaoka S., Ikeuchi K.: Matching and blending human motions using temporal scalable dynamic programming. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (2004), pp. 287–294.

[NNIY02] Nakazawa A., Nakaoka S., Ikeuch K., Yokoi K.: Imitating human dance motions through motion structure analysis. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (2002), pp. 2539–2544.

[NNK*05] Nakaoka S., Nakazawa A., Kanahiro F., Kaneko K., Morisawa M., Ikeuchi K.: Task model of lower body motion for a biped humanoid robot to imitate human dances. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (2005), pp. 3157–3162.

[NT04] Nava G. P., Tanaka H.: Finding music beats and tempo by using an image processing technique. In *Proceedings of International Conference on Information Technology for Application* (2004).

[Oga01] Ogawara K.: *Robot instruction of everyday manipulation tasks from human demonstration*. PhD thesis, The University of Tokyo, 2001. [in Japanese].

[OSU00] Osaki R., Shimada M., Uehara K.: Extraction of primitive motions by using clustering and segmentation of motion-captured data. *Journal of Japanese Society for Artificial Intelligence Vol. 15*, No. 5 (2000), 878–886. [in Japanese].

[OTI*00] Ogawara K., Takamatsu J., Iba S., Tanuki T., Kimura H., Ikeuchi K.: Acquiring hand-action models in task and behavior levels by a learning robot through observing human demonstrations. In *Proceedings of IEEE-RAS International Conference on Humanoid Robots* (2000).

[PB02] Pullen K., Bregler C.: Motion capture assisted animation: Texturing and synthesis. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2002) Vol. 21*, No. 3 (2002), 501–508.

[PHRA02] Pollard N. S., Hodgins J. K., Riley M. J., Atkenson C. G.: Adapting human motion for the control of a humanoid robot. In *Proceedings of IEEE International Conference on Robotics and Automation* (2002), pp. 1390–1397.

[PO03] Peters C., O' Sullivan C.: Bottom-up visual attention for virtual human animation. In *Proceedings of International Conference on Computer Animation and Social Agents* (2003), pp. 111–117.

[RBC98] Rose C. F., Bodenheimer B., Cohen M. F.: Verbs and adverbs: Multidimensional motion interpolation using radial basis function. *IEEE Transactions on Computer Graphics and Applications Vol. 18*, No. 5 (1998), 32–40.

[RL00] Rusinliwicz S., Levoy M.: QSplat: A multiresolution point rendering system for large meshes. In *Proceedings of ACM SIGGRAPH 2000* (2000), ACM Press, pp. 343–352.

[RNKI06] Ruchanurucks M., Nakaoka S., Kudoh S., Ikeuchi K.: Humanoid robot motion generation with sequential physical constraints. In *Proceedings of IEEE International Conference on Robotics and Automation* (2006), pp. 2649–2654.

[Roa96] Roads C.: *The Computer Music Tutorial*. The MIT Press, 1996.

[Ros92a] Rosenthal D.: Emulation of human rhythm perception. *Computer Music Journal Vol. 16*, No. 1 (1992), 64–76.

[Ros92b] Rosenthal D.: *Machine Rhythm: Computer Emulation of Human Rhythm Perception*. PhD thesis, Massachusetts Institute of Technology, 1992.

[RP03] Reitsma P. S. A., Pollard N. S.: Perceptual metrics for character animation: sensitivity to errors in ballistic motion. *ACM Transactions on Graphics Vol. 22*, No. 3 (2003), 537–542.

[RP04] Reitsma P. S. A., Pollard N. S.: Evaluating motion graphs for character navigation. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), pp. 89–98.

[RPE*05] Ren L., Patrick A., Efros A. A., Hodgins J. K., Rehg J. M.: A data-driven approach to quantifying natural human motion. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2005) Vol. 24*, No. 3 (2005), 1090–1097.

[RSC01] Rose C. F., Sloan P.-P. J., Cohen M. F.: Artist-directed inverse-kinematics using radial basis function interpolation. *Computer Graphics Forum (Proceedings of Eurographics 2001) Vol. 21*, No. 3 (2001), 239–250.

[SCF06] Shapiro A., Cao Y., Faloutsos P.: Style components. In *Proceedings of Graphics Interface* (2006).

[Sch98] Scheirer E. D.: Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustic Society of America Vol. 103*, No. 1 (1998), 588–601.

[SDO*04] Stone M., DeCarlo D., Oh I., Rodriguez C., Stere A., Lees A., Bregler C.: Speaking with hands: Creating animated conversational characters from recordings of human performance. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2004) Vol. 23*, No. 3 (2004), 506–513.

[SH05] Safonova A., Hodgins J. K.: Analyzing the physical correctness of interpolated human motion. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), pp. 171–180.

[Sho85] Shoemake K.: Animating rotation with quaternion curves. In *Proceedings of ACM SIGGRAPH 1985* (1985), ACM Press, pp. 245–254.

[SHP04] Safonova A., Hodgins J. K., Pollard N. S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2004) Vol. 23*, No. 3 (2004), 514–521.

[SKK04] Sakamoto Y., Kuriyama S., Kaneko T.: Motion map: Image-based retrieval and segmentation of motion data. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), ACM Press, pp. 259–266.

[SLSG01] Shin H. J., Lee J., Shin S. Y., Gleicher M.: Computer puppetry: An importance-based approach. *ACM Transactions on Graphics Vol. 20*, No. 2 (2001), 67–94.

[SLY*05] Shen X., Li Q., Yu T., Geng W., Lau N.: Mocap data editing via movement notations. In *Proceedings of International Conference on Computer Aided Design and Computer Graphics* (2005).

[SMK05] Sakuma T., Mukai T., Kuriyama S.: Psychological model for animating crowded pedestrians. *Computer Animation and Virtual Worlds Vol. 16*, No. 3-4 (2005), 343–351.

[SMKT06] Shiratori T., Matsushita Y., Kang S. B., Tang X.: Video completion by motion field transfer. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2006), vol. 1, pp. 411–418.

[SMS05] Sethares W. A., Morris R. D., Sethares J. C.: Beat tracking of musical performance using low-level audio features. *IEEE Transactions on Speech and Audio Processing Vol. 13*, No. 2 (2005).

[SO06] Shin H. J., Oh H. S.: Fat graphs: Constructing an interactive character with continuous controls. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006).

[SXWK04] Shao X., Xu C., Wang Y., Kankanhalli M. S.: Automatic music summarization in compressed domain. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing* (2004), pp. IV–261 – IV–264.

[Tod94] Todd N. P. M.: The auditory primal sketch: A multiscale model of rhythmic group. *Journal of New Music Research Vol. 23*, No. 1 (1994), 25–70.

[TSK00] Tak S., Song O.-Y., Ko H.-S.: Motion balance filtering. *Computer Graphics Forum (Proceedings of Eurographics 2000) Vol. 19*, No. 3 (2000), 437–446.

[TTO*00] Takamatsu J., Tominaga H., Ogawara K., Kimura H., Ikeuchi K.: Symbolic representation of trajectories for skill generation. In *Proceedings of IEEE International Conference on Robotics and Automation* (2000), pp. 4077–4082.

[UAT95] Unuma M., Anjyo K., Takeuchi R.: Fourier principles for emotion-based human figure animation. In *Proceedings of ACM SIGGRAPH 1995* (1995), pp. 91–96.

[UGB*04] Urtasan R., Glardon P., Boulic R., Thalmann D., Fua P.: Style-based motion synthesis. *Computer Graphics Forum Vol. 23*, No. 4 (2004), 799–812.

[VBSS90] Vukobratović M., Borovac B., Suria D., Stokic D.: *Biped Locomotion: Dynamics, Stability, Control and Application*. Vol. 7 of Scientific Fundamentals of Robotics, Springer-Verlag, 1990.

[VJ69] Vukobratović M., Juricic D.: Contribution to the synthesis of biped gait. *IEEE Transactions on Biomedical Engineering Vol. 16*, No. 1 (1969), 1–6.

[War] Warabi-za: http://www.warabi.or.jp/. [in Japanese].

[WDAC06] Wang J., Drucker S. M., Agrawala M., Cohen M. F.: The cartoon animation filter. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2006) Vol. 25*, No. 3 (2006), 1169–1173.

[WH97] Wiley D. J., Hahn J. K.: Interpolation synthesis of articulated figure motion. *IEEE Transactions on Computer Graphics and Applications Vol. 17*, No. 6 (1997), 39–45.

[WK88] Witkin A., Kass M.: Spacetime constraints. In *Proceedings of ACM SIGGRAPH 88* (1988), pp. 105–108.

[WLZ04] Wang M., Lu L., Zhang H.-J.: Repeating pattern discovery from acoustic musical signals. In *Proceedings of IEEE International Conference on Multimedia and Expo* (2004), pp. 2019–2022.

[WP95] Witkin A., Popović Z.: Motion warping. In *Proceedings of ACM SIGGRAPH 95* (1995), pp. 105–108.

[WSI04] Wexler Y., Shechtman E., Irani M.: Space–time video completion. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2004), pp. 120–127.

[YSLG05] Yu T., Shen X., Li Q., Geng W.: Motion retrieval based on movement notation language. *Computer Animation and Virtual Worlds Vol. 16*, No. 3-4 (2005), 273–282.

List of Publications

Journal Papers

1. 白鳥貴亮，中澤篤志，池内克史，“音楽特徴を考慮した舞踊動作の自動生成”，電子情報通信学会論文誌 D-II (条件付採録)

2. Takaaki Shiratori, Atsushi Nakazawa, Katsushi Ikeuchi, “Dancing-to-Music Character Animation”, *Computer Graphics Forum*, Vol. 25, No. 3, pp. 449-458, 2006.9

3. 池内克史，中澤篤志，工藤俊亮，中岡慎一郎，白鳥貴亮，“観察学習パラダイムに基づく二足歩行ヒューマノイドロボットによる舞踊動作の再現”，バイオメカニクス研究, Vol. 10, No. 3, pp. 190-202, 2006.9

4. 白鳥貴亮, 中澤篤志, 池内克史, “モーションキャプチャと音楽情報を用いた舞踊動作解析手法”, 電子情報通信学会論文誌 D-II, Vol. J88-D-II, No. 8, pp. 1583-1590, 2005.8

5. 池内克史，中澤篤志，小川原光一，高松淳，工藤俊亮，中岡慎一郎，白鳥貴亮，“民俗芸能のデジタルアーカイブとロボットによる動作提示”， 日本バーチャルリアリティ学会学会誌, Vol. 9, No. 2, pp. 14-20, 2004.6

6. Atsushi Nakazawa, Shinichiro Nakaoka, Takaaki Shiratori, Katsushi Ikeuchi, “Analysis and Synthesis of Human Motions using Motion Capture”, *Journal of Three Dimensional Images*, Vol. 17, No. 4, pp. 77-84, 2003.12

Commentary Articles

1. 白鳥貴亮，中澤篤志，池内克史, “人の動きのデジタル化とその応用 -モーションキャプチャと音楽情報を利用した舞踊動作の解析と生成-”, 月刊画像ラボ 3 月号, pp. 1-5, 2006.3

International Conferences

1. Miti Ruchanurucks, Shunsuke Kudoh, Koichi Ogawara, Takaaki Shiratori, Katsushi Ikeuchi, "Humanoid Robot Painter: Visual Perception and High-Level Planning", In *Proc. IEEE International Conference on Robotics and Automation* (ICRA2007), 2007.4 (to appear)

2. Daisuke Miyazaki, Mawo Kamakura, Tomoaki Higo, Yasuhide Okamoto, Rei Kawakami, Takaaki Shiratori, Akifumi Ikari, Shintaro Ono, Yoshihiro Sato, Mina Oya, Masayuki Tanaka, Katsushi Ikeuchi, Masanori Aoyagi, "3D Digital Archive of the Burghers of Calais," In *Proc. International Conference on Virtual Systems and Multimedia* (VSMM2006), Lecture Notes in Computer Science (LNCS), 2006.10

3. Takaaki Shiratori, Atsushi Nakazawa, Katsushi Ikeuchi, "Dancing-to-Music Character Animation", In *Proc. Eurographics 2006*, 2006.9

4. Manoj Perera, Takaaki Shiratori, Shunsuke Kudoh, Atsushi Nakazawa, Katsushi Ikeuchi, "Task Recognition and Style Analysis in Dance Sequences", In *Proc. IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems* (MFI2006), 2006.9

5. Takaaki Shiratori, Yasuyuki Matsushita, Sing Bing Kang, Xiaoou Tang, "Video Completion by Motion Field Transfer", In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR2006), 2006.6

6. Takaaki Shiratori, Atsushi Nakazawa, Katsushi Ikeuchi, "Synthesizing Dance Performance Using Musical and Motion Features", In *Proc. IEEE International Conference on Robotics and Automation* (ICRA2006), 2006.5

7. Takaaki Shiratori, Atsushi Nakazawa, Katsushi Ikeuchi, "Detecting Dance Motion Structure using Motion Capture and Musical Information", In *Proc. International Conference on Virtual Systems and Multimedia* (VSMM2004), 2004.11

8. Takaaki Shiratori, Atsushi Nakazawa, Katsushi Ikeuchi, "Detecting Dance Motion Structure through Music Analysis", In *Proc. IEEE International Conference on Automatic Face and Gesture Recognition* (FGR2004), 2004.5

9. Atsushi Nakazawa, Shinichiro Nakaoka, Takaaki Shiratori, Katsushi Ikeuchi, "Analysis and Synthesis of Human Motions using Motion Capture", In *Proc. International Conference on Humans and Computers* (HC2003), 2003.8

10. Takaaki Shiratori, Atsushi Nakazawa, Katsushi Ikeuchi, "Rhythmic Motion Analysis using Motion Capture and Musical Information", In *Proc. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* (MFI2003), 2003.7

11. Atsushi Nakazawa, Shinichiro Nakaoka, Takaaki Shiratori, Katsushi Ikeuchi, "Analysis and Synthesis of Human Dance Motions", In *Proc. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* (MFI2003), 2003.7

Domestic Conferences

1. 白鳥貴亮, 松下康之, Sing Bing Kang, Xiaoou Tang, "Video Completion by Motion Field Transfer", マイクロソフト産学連携研究機構第 2 回シンポジウム, 2006.11

2. 白鳥貴亮, 中澤篤志, 池内克史, "音楽情景を考慮した舞踊動作", 画像の認識・理解シンポジウム (MIRU2006), 2006.7

3. 白鳥貴亮, 中澤篤志, 池内克史, "音楽情景に基づいた舞踊動作の自動生成", Visual Computing / グラフィクスと CAD 合同シンポジウム 2006, 2006.6 (**invited talk**)

4. 白鳥貴亮, 中澤篤志, 池内克史, "音楽およびモーションキャプチャデータからの舞踊動作生成手法", 第 23 回日本ロボット学会学術講演会, 2005.9

5. 中澤篤志, 白鳥貴亮, 池内克史, "観察に基づく音楽およびモーションキャプチャデータからの舞踊動作生成手法", 画像の認識・理解シンポジウム (MIRU2005), 2005.7

6. 中澤篤志, 中岡慎一郎, 白鳥貴亮, 工藤俊亮, 池内克史, "モーションキャプチャによる全身運動解析と模倣ロボット　ー「じょんがら」節を HRP-1S に踊らせるー", 情報処理学会 コンピュータビジョンとイメージメディア研究報告 (CVIM), 2004.11

7. 白鳥貴亮, 中澤篤志, 池内克史, "モーションキャプチャと音楽情報を利用した舞踊動作の構造解析", 画像の認識・理解シンポジウム (MIRU2004), 2004.7

8. 白鳥貴亮, 中澤篤志, 池内克史, "楽音のリズムを考慮した舞踊動作の解析", 第 21 回日本ロボット学会学術講演会, 2003.9

Patents

1. Takaaki Shiratori, Yasuyuki Matsushita, Sing Bing Kang, Xiaoou Tang, "Video Completion by Motion Field Transfer", Pending in USA

2. 池内克史, 中澤篤志, 白鳥貴亮, "モーション作成装置およびモーション作成方法並びにこれらに用いるプログラム", 特願 2005-200922，Pending in Japan