AUDIOVISUAL ANALYSIS FOR LEARNING AND SYNTHESIS OF DANCE PERFORMANCES

by

Ferda Ofli

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in

Electrical & Electronics Engineering

Koç University

August, 2010

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a Ph.D. thesis by

Ferda Ofli

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Prof. A. Murat Tekalp

Assoc. Prof. Yücel Yemez

Assoc. Prof. Engin Erzin

Asst. Prof. Deniz Yüret

Prof. Lale Akarun

Date: _____

To my family

ABSTRACT

In this thesis, we propose an audiovisual framework for analysis of dance performances towards music-driven dance motion synthesis. The proposed framework first aims to extract elementary recurrent music and dance motion patterns. Then the analyses of the correlations between the elementary music and dance motion patterns are used to construct many-to-many statistical mappings from music to dance motions. These many-to-many mappings are then used for music-driven dance choreography synthesis and personalized dance performance animations. Based on this audiovisual framework, we first present a system that deals primarily with the unsupervised correlation analysis of elementary recurrent music and dance motion patterns. Later we present a second system that considers both analysis and synthesis parts of the proposed framework in a rather simplified context where a dance performance is assumed to have only a single dance motion pattern which is to be synchronized with the musical beat. Finally, we present a complete system for modeling, analysis, and synthesis of audiovisual dance performances that handles more complex and realistic scenarios. The third system automatically synthesizes a variety of synchronized dance performances that perceptually match the emotions and contents of the accompanying music; as if they were arranged by a choreographer. Experimental results for each system demonstrate that the proposed framework is able to extract and utilize from audiovisual correlations between music and dance motion patterns for synthesis of compelling music-driven dance performances.

ÖZETÇE

Bu tez çalışmasında müzikle sürülen dans hareketi sentezi için çeşitli dans performanslarını inceleyen işitsel-görsel bir çatı yapısı önerilmektedir. Önerilen çatı yapısı öncelikle yinelenen temel dans ve müzik örüntülerini çıkarmayı hedefler. Daha sonra çıkarılan dans ve müzik örüntüleri arasındaki ilintiler incelenerek müzikten dans hareketlerine giden çoktan çoğa bağıntılar oluşturulur. Bu bağıntılar ise müzikle sürülen dans koreografisi sentezinde ve kişiye özgü dans performansı animasyonu oluşturulmasında kullanılır. Bu çatı yapısını baz alarak ilk önce yinelenen temel dans ve müzik örüntülerinin güdümsüz çıkarımı ve ilinti analizini ele alan bir sistem sunmaktayız. Daha sonra ikinci bir sistemle önerilen çatı yapısının hem analiz hem de sentez kısımlarını görece basit bir senaryoda, dans performanslarının bir müzik için bir dans figüründen oluştuğu durumda, ele almaktayız. Son olarak, daha karmaşık bir senaryo için gerekli modelleme, analiz ve sentezi topyekün yapabilecek tam donanımlı bir sistem sunmaktayız. Bu sistem verilen müziğin içeriğine ve yapısına uygun alternatif dans koreografilerini otomatik olarak sentezlemektedir. Her bir sistem için deneysel sonuçlar göstermiştir ki; önerilen çatı yapısı müzik ve dans hareket örüntülerini belirlemede, belirlenen örüntüler arasında bağıntı modelleri oluşturmada, oluşturulan bağıntı modelleri ile müziğe uygun dans hareketleri sentezlemede başarılıdır.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor Prof. A. Murat Tekalp for being a great influence both in this research and personally with his incredible dedication, interest and advice. I also would like to thank my co-advisor Assoc. Prof. Yücel Yemez for providing me with constructive criticism and research freedom during my studies. I am grateful to Assoc. Prof. Engin Erzin for his inspiring enthusiasm and unreserved support at every stage of this thesis. They have been exceptional tutors from whom I have learned much more than just science. Without their unfailing guidance and encouragement, this thesis would not have been a success.

I would like to thank the other members of my thesis committee, Prof. Lale Akarun and Asst. Prof. Deniz Yüret, for their critical reading of this thesis and for their valuable comments. I also wish to thank Prof. A. Tanju Erdem, Assoc. Prof. Çağatay Başdoğan, Asst. Prof. T. Metin Sezgin, Assoc. Prof. Alper T. Erdoğan and Asst. Prof. S. Serdar Kozat for their generous help and comments that have expedited this thesis.

I owe my special thanks to Özgür Ülkem and Zeynep Yavuzcezzar who have shared time and expertise, given acquisitions for our audiovisual folk dance database.

I am indebted to my family, my parents Selma and Besim, and my sister Sevda, for enduring me all my life and for their infinite trust in me and encouragement to further my education.

Finally, my sincere appreciation goes to my friends who have been a family to me for the last ten years of my life at Koç University and for providing me the morale support that helped me in hard days of my research.

This thesis has been supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) and by the European Sixth Framework Programme Network of Excellence SIMILAR, and Network of Excellence 3DTV.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

CSCC	Chroma-Scale Cepstral Coefficient
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DTW	Dynamic Time Warping
FFT	Fast Fourier Transform
HCI	Human-Computer Interaction
HMM	Hidden Markov Model
MFCC	Mel-Frequency Cepstral Coefficient

Chapter 1

INTRODUCTION

Unimodal signal processing has reached high levels of sophistication to extract information individually from several modalities. However, we are still far away from the point where we exhaust all the information available in a modality. That is, valuable new information may appear when we put modalities together to seek for the hidden joint correlation between them. Even though it is in its infancy at the moment, multimodal signal processing is likely to play a significant role in computer vision research as multimodal environments are being introduced into several areas of this research.

Multimodality, or the reliance on more than one semiotic[1] channel for conveying communicative content, is inherent in the face-to-face communication in everyday interaction. People draw on a range of visual, verbal, paralinguistic, and other cues to make sense of each other. This ability to combine impressions from different senses enables humans to extract information from and understand complex environments. Similar examples can be seen in most human-computer interaction systems where speech, facial expression, gestures, tactile, etc., play a key role in establishing communicative interfaces. Multimodality provides us with the possibility and, in some cases, ease of digesting more information than a unimodal channel could provide in a human computer interaction. Therefore, the study of multimodal signal processing, analysis, and understanding deals with the challenge of handling several sources of information at the same time. While each of these modalities have been separately modeled at high levels of sophistication, multimodal modeling is still in its infancy.

Multimodal signal processing benefits from integrating different signals which are phys-

[1]Semiotics, also called semiotic studies or semiology, is the study of sign processes (semiosis), or signification and communication, signs and symbols, both individually and grouped into sign systems. It includes the study of how meaning is constructed and understood.

ically of different nature for exploring the underlying mutual relationship to learn and to present the uttermost information available to them in various contexts. Hence, multimodality is, in some ways, a reinforcement that offsets the weaknesses or insufficiencies of one modality by the strengths of another modality. This idea is the driving motive in most of the man-machine, brain-machine, and human-computer interfaces. For example, a framework based on joint analysis and modeling of brain signals and muscle movements enables monkeys to consciously control the movement of a robot arm in real time, using only signals from their brains and visual feedback on a video screen. This can be considered as an important step towards rehabilitation of people with brain and spinal cord damage from stroke, disease or trauma [1]. On the other hand, a technique that uses image processing capabilities in lip reading can be used to recognize undeterministic phones in speech recognition, or can be employed in speech-driven lip animation [2]. Further studying the relations between speech and facial gestures can lead to more realistic speech-driven talking face animations [3, 4]. Joint analysis of voice and face uncovers important clues that help improve the emotion recognition performance [5]. Combining tactile, i.e., haptics, with visual information creates the necessary grounds not only for several medical applications [6] but also for artwork such as haptic sculpturing [7]. Yet another example can be multimodal biometrics where a combination of different biometric recognition techniques is used in order to provide more-than-average security by integrating the evidence presented by multiple sources of information [8].

1.1 Scope

The goal of this research is to investigate methods to combine and fuse different modes of information for applications in human-computer interaction (HCI) with particular focus on visual and auditive modalities. Modeling the correlation between musical audio and body motion patterns is the main motive in this thesis. For this purpose, a multimodal framework is devised to analyze, learn and synthesize audiovisual data in particular for human body motions in the context of dance performances accompanied by music. The joint correlation model can be perceived as a mapping between music and dance motion patterns. This mapping, for instance, can be used to predict dance motion patterns from music patterns.

In that case, music-driven dance motion animation emerges as one of the several interesting multimodal signal processing applications on which there exists little work in the literature.

Human body motion can have many purposes: To go from one place to another, humans walk or run. On the other hand, some body motions express emotions. Dancing is a special type of body motion that has some predefined structure; as well as emotional aspects. Analysis of gestures in dance with the purpose of uncovering the conveyed emotions has been undertaken in recent studies [9]. There are several challenges involved in audio-driven human body motion analysis and synthesis: First, there does not exist a well-established set of elementary audio and motion patterns, unlike phonemes and visemes in speech articulation. Secondly, body motion patterns are person dependent and open to interpretation, and may exhibit variations in time even for the same person. Thirdly, audio and body motion are not physiologically coupled and the synchronicity in between them may exhibit variations. Moreover, motion patterns may span time intervals of different lengths with respect to its audio counterparts. The recent work by Sargin et al. address the challenges similar to those mentioned above in the context of prosody-driven head gesture synthesis in [10].

Dancing to music is an artistic skill and dancing with rhythm and gestures that matches the rhythm and content of the accompanying music requires experience up to some level. Professional dance performances, therefore, rely on a priori design of dance motions, i.e., choreographies. Choreography is the art of arranging dance movements for performance. Choreographers tailor the sequences of body movements to music in order to embody or express ideas and emotions in the form of a dance performance. Therefore, dance is closely bound to music in its structural course, artistic expression, and interpretation. Specifically, the rhythm and intensity of body movements in a dance performance are in synchrony with those of the music, and hence, the metric orders in the course of music and dance structure coincide, as Reynolds states in [11]. In order to successfully establish the contextual bond as well as the structural synchrony between dance motions and the accompanying music, choreographers tend to thoughtfully design dance motion sequence for a new piece of music by utilizing from the repertoire of choreographies that have been carefully planed in the past for similar musical pieces. Based on this common practice of choreographers, our ultimate goal in this thesis is to build a framework for automatic creation of dance performances in synchrony with the accompanying music; as if they were arranged by a choreographer,

through learning many-to-many statistical mappings from music to dance. Specifically, our final framework aims at automatic design of alternative dance choregoraphies that are coherent and compelling to audience. It is important to note at this point that the term *choreography* is generally used in the sense of spatial formation (circle, line, square, couples, etc.), of plastic aspects of movement (types of steps, gestures, posture, grasps, etc.), and of progression in space (floor patterns). However, in this thesis, we use the term *choreography* solely in the sense of composition, i.e., the arrangement of the dance motion sequence.

The organization of this dissertation is as follows. The multimodal signal analysis framework is explained in Chapter 2. Chapter 3 outlines music and dance motion feature extraction involved in several tasks throughout the thesis. Chapter 4 analyzes basically the correlations between elementary music and dance motion patterns. Chapter 5 elaborates more on the implementation of the overall audiovisual analysis-synthesis framework in the a simplified dance performance context and presents preliminary results for music-driven dance motion synthesis. Chapter 6 eventually describes the complete implementation of the proposed multimodal music-driven dance performance synthesis framework. Experimental results are presented at the end of each chapter. Discussions are provided in Chapter 7.

1.2 Related Work

Automatic dance analysis, annotation and synthesis have been studied extensively in the literature with emphasis on human body motion analysis/synthesis and dance music analysis whereas there is relatively little work on the open problem of music-driven automatic dance animation as we address in this thesis.

For human body motion analysis/synthesis problem, Bregler et al. [12] describes a body motion recognition approach that incorporates low-level probabilistic constraints extracted from image sequences of articulated gestures into high-level manifold and HMM-based representations. In order to synthesize data-driven body motion, Arikan and Forsyth [13], and Kovar et al. [14] propose motion graphs, representing allowable transitions between poses, to identify a sequence of smoothly transiting motion segments. Brand and Hertzmann [15] studies motion "style" transfer problem which involves intensive motion feature analysis and learning motion patterns from a highly varied set of motion capture sequences using

HMMs. In a recent study, Min et al. [16] presents a generative human motion model for synthesis of personalized human motion styles by constructing a multilinear motion model that provides explicit parameterized representation of human motion in terms of "style" and "identity" factors.

Dance music analysis, in general, includes beat and tempo tracking, measure analysis, and rhythm and melody detection. In [17], Gao and Lee propose an adaptive learning approach to analyze music tempo and beat based on maximum a posteriori (MAP) estimation. Ellis describes a dynamic programming solution for beat tracking by finding the best-scoring set of beat times that reflect the estimated global tempo of music [18]. An extensive evaluation of audio beat tracking and music tempo extraction algorithms, which were included in MIREX'06, can be found in [19]. There are also some recent studies on the open problem of automatic musical meter detection [20, 21]. In the last decade, chromatic scale features have become popular in musical audio analysis; especially in music information retrieval, since introduced by Fujishima [22]. Lee and Slaney [23] describe a method for automatic chord recognition from audio using HMMs with supervised learning over chroma features. Ellis and Poliner [24] propose a cross-correlation based cover song identification system with chroma features and dynamic programming beat tracking. Recently, Kim et al. [25] calculate the second order statistics to form dynamic chroma feature vectors in modeling harmony structures for classical music opus identification.

One of the early dance notation systems, known as Labanotation, defines a data format to record human dance figures with graphical symbols that provides a detailed sequence of changes in human posture during a dance figure [26]. In [27], Li et al. segment body motions into textons, each of which was modeled by a linear dynamic system, to synthesize human body motion in a manner statistically similar to the original motion capture data by considering the likelihood of switching from one texton to the next. In [28], Ruiz and Vachon perform analysis of dance figures in a chain of simple steps using HMMs to perform automatic recognition of basic movements in the contemporary dance.

Most of the studies in the context of multimodal music and dance analysis towards dance motion synthesis focuses solely on the synchronization aspect of the problem between an existing animation and a piece of music. Cardle et al. [29] implement a system for synchronizing motion to music by locally modifying motions using perceptual music cues. Lee

and Lee [30] employ dynamic programming to synchronize animation with its background music by changing the timing of both the music and the motion via time-scaling the music and time-warping the motion. Kim et. al. [31] and Alankus et al. [32] use transition graphs to synthesize new motion sequences from motion capture data using the results of motion rhythm analysis. Since all these methods consider only musical rhythm, they are most suitable for aligning dance motion with beats in a musical piece. In general, they lack multimodal correlation analysis of dance motion and music. Hence, they can hardly synthesize expressive dance motion.

Shiratori et. al. propose in [33] a technique to synthesize dance motion that is perceptually matched to music by using a mapping based on the rhythmic similarities between music and motion segments for synchronizing the animation with the song. Even though this approach is very similar to ours, their mapping is based on calculating deterministic distance metrics between motion features and music features while our mapping is based on statistical learning of recurring music and dance motion patterns.

Sauer and Yang design in [34] a music-driven character animation tool which extracts a set of features such as the beat and dynamics (louds and softs) of the music to build an animation from a dictionary of pre-built dance movements specified by the user through a script file. This tool requires programming several dance primitives in advance and its synthesis scheme is user-oriented. However, our final framework automatically synthesizes alternative dance choreographies that are coherent and compelling to audience.

Kim et al. [35] investigate the correlations between musical and motion features by designing a matching process to consider the correspondence of relative changes in both musical and motion feature spaces. They introduce similarity matrices to match the amount of relative changes in both feature spaces and use correlation coefficients to measure the strength of the correlation between each pair of the musical and motion features. Even though Kim et al. emphasize matching the progression of musical and motion patterns like we do, they lack to represent in their method the diversity of dance performance by finding only the optimal dance motion sequence for a particular musical piece.

1.3 Overview and Contributions

In this thesis, several audiovisual dance performance analysis-synthesis frameworks are designed to model the correlation between music and dance that will be exploited for predicting dance motion from music. Details are presented in respective chapters as briefly described in sequel.

Chapter 2 explains the general structure and main modules of the designed multimodal signal processing framework. This framework is designed for analyzing two modalities, i.e., auditive and visual modalities, to "learn" a mapping between them which will then be used to estimate and synthesize visual modality from auditive modality. The framework consists of two main blocks: analysis and synthesis. Analysis block is based on a two-stage strategy. In the first stage, input modalities are investigated in a unimodal sense to determine and model the elementary units, or building blocks, of the modalities. This operation corresponds to temporal segmentation and modeling of the low level feature streams of each modality to determine the recurrent elementary patterns that exist in each modality. In the second stage, the correlation between the elementary units of each modality is modeled as a mapping which can be used for estimation and synthesis of visual modality from auditive modality.

Chapter 3 outlines the necessary tools and techniques for extracting several different music and dance motion features that are used throughout the thesis. For musical audio analysis, we extract static features such as beat frequency as well as dynamic features such as mel-frequency cepstral coefficients (MFCCs) and chroma-scale cepstral coefficients (CSCCs). On the other hand, two types of dance motion features are extracted: 3D joint displacements and 3D joint angles. It is crucial to note that both sets of dance motion features entail an optical motion capture system for obtaining the optical motion capture data that basically represents the 3D dance motion trajectory of the dancer. The multicamera motion capture system built for this purpose is also explained in this chapter.

Chapters 4, 5 and 6 present applications of the multimodal signal processing framework proposed in Chapter 2 in the context of dance performances, where gestures and 3D movements of a dancer are mainly driven by musical piece and characterized by repetition of a set of dance figures. Several scenarios are examined in adaptation of the general framework

proposed in Chapter 2 to different applications that impose different application specific constraints to the general framework. Therefore, Chapters 4, 5 and 6 provide with application specific versions of the block diagram drawn for the general framework in Figure 2.1 in Chapter 2.

Chapter 4 addresses the problem of multiview audiovisual *analysis* of dance figures to create a correlation model between body motion and music by *unsupervised* temporal segmentation of the recurrent elementary music and body motion patterns. This scenario assumes no prior information about the content of the audiovisual data and tries to automatically segment the modalities in order to build a correlation model between the modalities. Synthesis task is not considered in this scenario.

Chapter 5, on the other hand, considers an automatic *analysis-synthesis* scheme for music-driven dance animation based on *supervised* modeling of music and dance figures. This scenario is built upon a rather simplified setting, where a dance performance is assumed to have repetitions of only a single dance figure. The main focus of this scenario is to examine the synthesis problem of dance figures in synchrony with input audio.

Chapter 6 requires dealing with the complete framework for *modeling*, *analysis*, *annotation* and *synthesis* of multimodal dance performances, which can handle more complex and realistic scenarios with respect to the first two scenarios. In this scenario, we *analyze* correlations between music features and dance figure labels on some training dance videos in order to construct many-to-many statistical mappings from music measures (segments) to dance figures towards *generating* music-driven *personalized* dance performance animations.

The major contribution of this thesis is a multimodal signal processing framework for modeling, analysis, annotation and synthesis of different modalities, which

- handles the feature level analysis and synthesis of dance performances,
- attempts to create statistical models for one-to-many, many-to-many mappings between different modalities,
- describes techniques for automatic synthesis of one modality from the other one,
- provides several applications to animation and motion picture industry.

In addition, a list of minor contributions, which are necessary in fulfilling major contributions, can be given as,

- marker-based multicamera motion capture,
- chroma-scale cepstral coefficients,
- musical measure clustering.

Chapter 2

MULTIMODAL SIGNAL ANALYSIS AND SYNTHESIS FRAMEWORK

This chapter presents a general framework for analysis of multimodal signals, finding a mapping between the modalities and synthesis of one modality driven by the other modality. The framework is based on a two-stage method for joint analysis of the recurrent temporal unimodal patterns. Figure 2.1 describes the main blocks of the proposed multimodal signal analysis framework. The analysis method is used to "learn" correlations between the two modalities. The resulting mapping model is then employed to synthesize a sequence of unimodal patterns for one of the modalities using the learnings while the other modality serves as the driving input to the synthesis module.

The proposed framework is described up to a level that is detailed enough for general purpose modeling of any correlated (but not known or modeled, yet) modalities. Specific constraints can be imposed and necessary arrangements can be made according to the properties of the modalities under investigation. Since the scope of this thesis is confined to modeling the correlation between auditive and visual modalities in different scenarios, application specific constraints and arrangements will be introduced in the related chapters of the thesis. The details of the sub-blocks, given in Figure 2.1, are described in the following sections.

2.1 Unimodal Signal Analysis

In the first stage, unimodal analysis of the modalities includes two tasks: feature extraction and labeling/annotation. Then, hidden Markov model (HMM) based supervised/unsupervised temporal segmentation of modalities is employed independently to determine and learn the elementary recurrent patterns in both modalities.

One can consider a supervised approach for unimodal analysis of signals as long as there is information such as the knowledge of the elementary units that define the recurrent

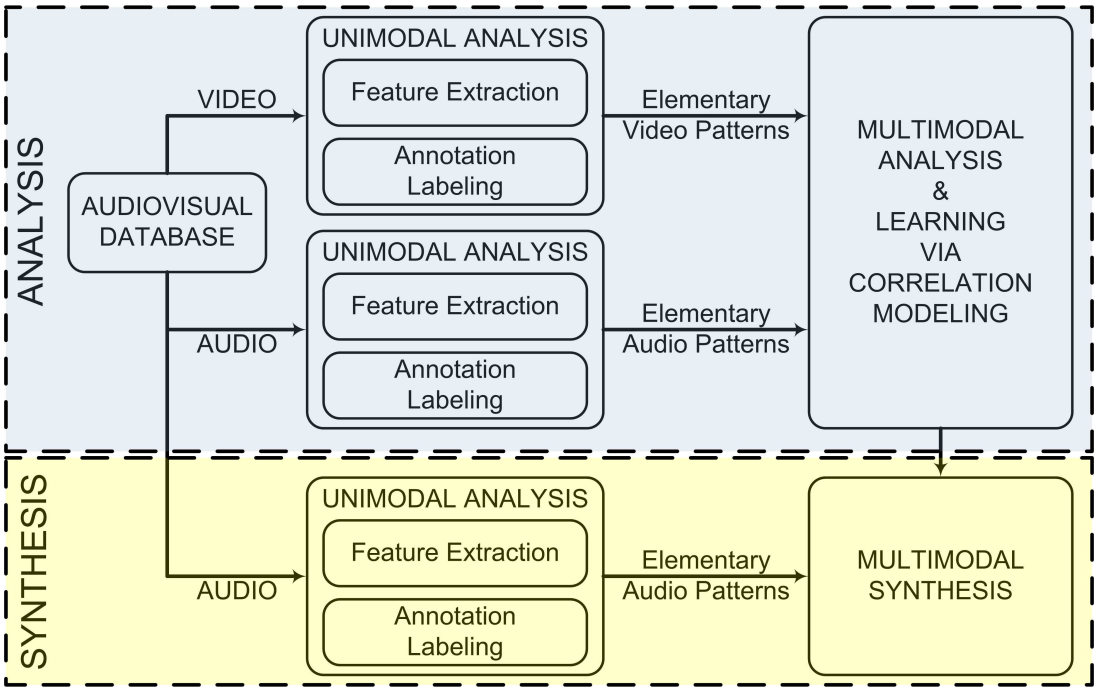


Figure 2.1: A general description of the proposed multimodal signal processing framework.

patterns in the modalities, which are also easy to manually segment and label. Then, supervised analysis will basically maximize the utilization of available information. In this case, an appropriate set of features will be used to train HMMs according to the known elementary unit labels.

The HMMs used for supervised modeling, in general, has a single branch left-to-right structure as shown in Figure 2.2. Even though the given HMM structure is simple, there are some design parameters that must be chosen according to the context of the problem. For instance, the number of states in an HMM structure, the number of Gaussian functions in a state that models the input parameters, or the type of the covariance matrix (whether diagonal or full) that models the relation within the feature set must be determined carefully to attain a successful model. Once necessary adjustments are accomplished, one HMM for each label is trained over the training data and the performance is measured by recognition rate of the trained models over the test data.

When there is not much information about the elementary units of the modalities, i.e.,

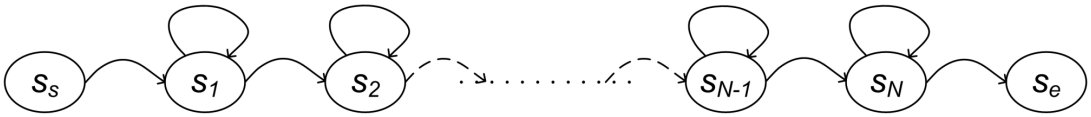


Figure 2.2: A simple left-to-right HMM structure with N states in between the start and end states.

labels or annotation scheme is not available, unsupervised analysis/segmentation comes into play to make sense of the available information to infer useful results that are not readily available. In other words, each modality has to be automatically segmented into its meaningful recurrent elementary patterns. In this case, individual feature streams corresponding to each modality are used to train separate parallel HMM structures, which provide probabilistic models for temporal recurrent patterns in respective modalities. Then, the segments corresponding to these patterns are detected and labeled automatically over the training data.

The parallel HMM structure used for unsupervised temporal segmentation has M parallel left-to-right branches and each branch is composed of N states as shown in Figure 2.3. In addition to the design parameters listed for single branch left-to-right HMM structure, the number of branches, in other terms, the number of temporal patterns must be determined prior to training of the parallel HMM structure.

2.2 Multimodal Signal Analysis and Learning via Correlation Modeling

In the second stage, correlations between the elementary recurrent patterns of the modalities is jointly analyzed to extract the recurrent co-occurring patterns. This joint correlation model can be based on simple but efficient analysis such as the co-occurrence matrix obtained from the co-occurring multimodal events; or, on a thorough analysis which uses multi-stream HMMs to determine a multimodal mapping model.

The multimodal analysis of the modalities by the use of multi-stream HMM structures is similar to the use of parallel HMM structures in unimodal analysis of the modalities explained in Section 2.1, except in this case, the input to the HMM structure is a multi-

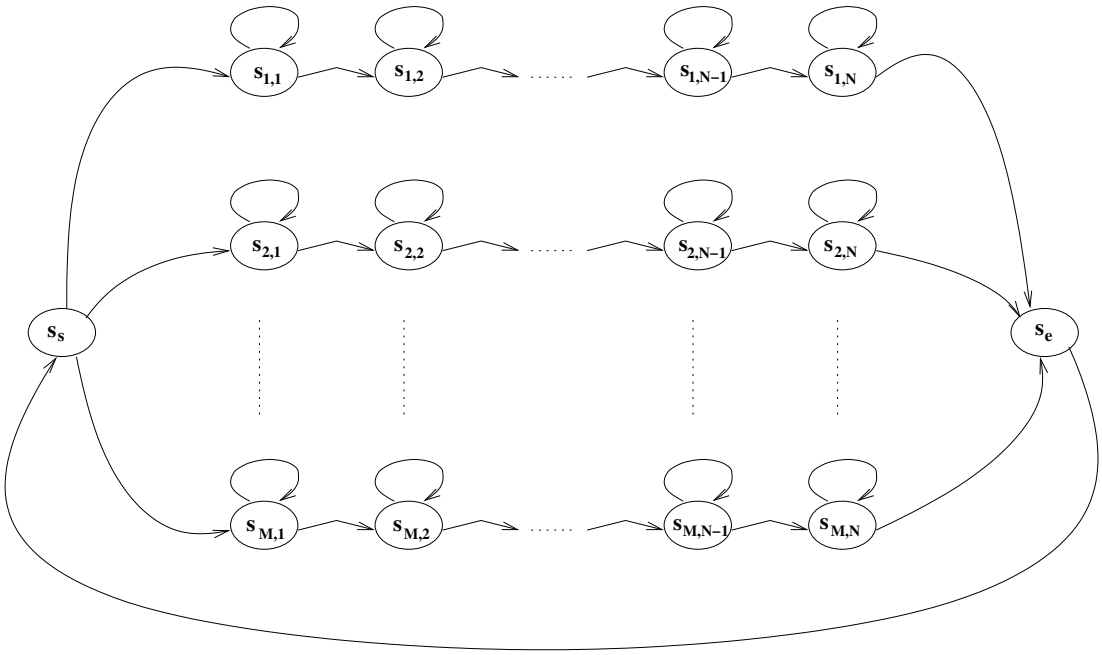


Figure 2.3: A parallel left-to-right HMM structure with M branches and N states at each branch.

stream discrete observation sequence. Therefore, the parallel HMM structure in Figure 2.3 is used with discrete multi-stream HMM branches for this task. In multi-stream HMMs, all streams share the same state transition structure however emission probabilities are determined independently for each stream.

It is important to note that if a multi-stream HMM structure is directly employed for joint analysis of the modalities over their individual feature streams, as commonly used for event detection [36], instead of the proposed two-stage analysis, the resulting joint feature segments would not necessarily correspond to independent meaningful elementary patterns for the modalities.

In either case, the resulting joint model stores the correlation between the two modalities. Specifically, this model "learns" the mapping between the elementary units of the modalities which enables us to infer one of the modality from the other.

2.3 Multimodal Signal Synthesis

In the synthesis stage, the multimodal mapping model is used to predict a sequence of elementary patterns for one of the modalities according to the elementary pattern sequence computed for the other modality. The set of parameters corresponding to the estimated sequence of elementary patterns are then synthesized using the HMMs obtained at the unimodal signal analysis stage and finally animated on an appropriate model.

Synthesis task requires a unimodal analysis of the input modality as suggested by Figure 2.1. The purpose of this initial step is to compute the sequence of elementary patterns for the test data of the input modality. Then, the computed sequence is used in conjunction with the learned multimodal mapping model to determine a sequence of elementary patterns for the output modality. The output, at this stage, is the list of labels that represent each of the elementary pattern that belongs to the output modality. However, it is further possible to compute the corresponding parameters (features) if the HMM related to each label is previously trained or somehow available. Once the parameters are generated, they can be used to animate an appropriate model to visualize the output of the synthesis. Visualization of the synthesis results plays an important role in subjective evaluation of the proposed framework.

2.4 Summary

In this chapter, we presented a general framework that performs temporal unimodal/multimodal signal analysis to learn elementary recurrent patterns, finds mappings between learned patterns of each modality, and synthesize patterns of one modality driven by the other modality. We described the basics of the proposed framework up to a level that is detailed enough for general purpose modeling of any correlated (but not known or modeled, yet) modalities. Specific constraints will be imposed and necessary arrangements will be made according to the properties of the modalities under investigation in the following chapters.

Chapter 3

AUDIOVISUAL FEATURE EXTRACTION

Modeling of temporal elementary recurrent music and dance motion patterns entails extraction of informative audio and video features. The quality of features extracted from each modality strongly effects the pattern modeling quality. Hence, the type of features to be used in modeling elementary recurrent music and dance motion patterns must be chosen carefully according to the problem statement. In this thesis, we extract dynamic music and dance motion features to account for the temporal statistics of the input music and dance motion as well as static music and dance motion features to tackle with the synchronization problem of the two modalities. For this purpose, we extract chroma- and mel-scale cepstral coefficients as dynamic music features whereas we extract 3D joint angles and displacements as dynamic dance motion features. On the other hand, musical beat frequency and measure detection is used as static music features. Before we elaborate on the details of several different music and dance motion feature extraction tasks, we will focus on multicamera motion capture system used in the audiovisual feature extraction. It is an integral part of dance motion feature extraction and the accuracy of motion capture process determines the quality of dance motion features. Therefore, an extensive analysis of the multicamera motion capture system is first provided in Section 3.1. Then, dance motion feature extraction and music feature extraction tasks are explained in Sections 3.2 and 3.3, respectively. Summary of the chapter is presented in Section 3.4.

3.1 Multicamera Motion Capture

Optical motion capture systems have continuously been evolving and there already exist various techniques and approaches in the literature, that can be distinguished mainly based on whether they make use of markers (active or passive), or fully rely on image features, and the type of motion analysis they employ (model-based or not). Aggarwal and Cai review the research progress on human motion analysis in [37] in detail and Gavrila provides an

in-depth survey in [38].

Marker-based systems rely on the contrast of the markers with the background to capture their motion. One can use active capture systems, such as LED markers that pulse in synchronization with the cameras' digital shutters, or passive systems, such as using strongly retro-reflective markers along with an illumination source co-located with each camera. These methods however can not acquire and capture the shape and texture properties of the subject, which could also give supplementary information about location of feature points. Hence, [39] proposes a motion capture algorithm based on the use of simple color-markers, aiming at a visually guided and more controllable 3D animation system. On the other hand, in [40], a vision-based full-body estimation and interaction system that uses a marker-less method is presented. It first extracts 2D blob features, and then estimates the 3D full-body parameters. Ricquebourg and Bouthemy in [41] develop a method to track the apparent contours of a moving articulated structure, avoiding the use of 3D models.

There exist a number of marker-based commercial systems as evaluated in [42, 43] for human motion capture but most of them rely on a high number of cameras to avoid occlusions, high frame rates or expensive hardware. In this work, we describe a low-cost method for multicamera marker-based body motion capture, that is accurate enough to train our analysis-synthesis system. Our method tracks the 3D positions of the joints of the body based on the markers' 2D projections on each camera's image plane. The proposed motion capture technique is based on 3D tracking of the markers attached to the person's body in the scene without need for an explicit 3D model (see Figure 3.1). We make use of the multistereo correspondence information from multiple cameras to obtain 3D positions of the markers. This provides us with a set of 3D point locations over time that expresses the alignment of the markers in 3D world. We employ Kalman filtering for smoothing out the observations and predicting the next target locations of the points in that point cloud in a similar fashion explained in [44]. This method also allows users to intervene into the tracking process, and therefore, has the benefit of producing accurate tracking results by letting users correct errors manually during the tracking process. However, the tracking process itself may become lengthy and cumbersome.

Figure 3.1: An example scene captured by the multicamera system available at Koç University. Markers are attached at or around the joints of the dancer's body.

3.1.1 Initialization

For a given frame in the video sequence, a set of N images are obtained from the N cameras. Each camera is modeled using a pinhole camera model based on perspective projection. Accurate calibration information is available. In order to estimate the 2D positions of the markers attached to the body of the dancer in the set of N images for a given frame, the original images are processed in the YCrCb color space which gives flexibility over intensity variations in the frames of a video as well as among the videos captured by the cameras from different views. In order to learn the chrominance information of the marker color, markers on the dancer are manually labeled in the first frame for all camera views. We assume that the distributions of Cr and Cb channel intensity values belonging to marker regions are Gaussian. Thus, we calculate the mean, μ, and the covariance, $\mathbf{\Sigma}$, over each marker region (a pixel neighborhood around the labeled point), where $\mu = [\mu_{Cr}, \mu_{Cb}]^T$ and $\mathbf{\Sigma} = (\mathbf{c} - \mu)(\mathbf{c} - \mu)^T$, $\mathbf{c}$ being $[c_{Cr}, c_{Cb}]^T$. Then, a threshold in the Mahalanobis sense with $(\mu, \mathbf{\Sigma})$ is applied to all images in order to detect marker locations. The number of detected markers in every image may vary due to occlusions. However, tracking information and redundancy among views allow us to overcome this problem.

3.1.2 3D Joint Position Tracking

The motion capture process in this case involves retrieving the body configuration in terms of its defining parameters, namely $\mathbf{P}_t = \{\mathbf{p}_0, \ldots, \mathbf{p}_{M-1}\}_t$, from the multiple video streams at a given time t. This set of parameters consists of the 3D positions of the markers located about the articulation points. The 3D position of each marker at each frame is determined via triangulation based on the observed 2D projections of the markers on each camera's image plane.

Let M be the number of markers on the dancer and $\mathbf{W}$ be the set of search windows, where $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$ such that each window $\mathbf{w}_m$ is centered around the location, $[x_m, y_m]^T$, of the corresponding marker. The set $\mathbf{W}$ is used to track markers over frames. Thus the center of each search window, $\mathbf{w}_m$, is initialized as the point manually labeled in the first frame and specifies the current position of the marker.

To track the marker positions through the incoming frames, we use the Mahalanobis distance from $\mathbf{c}$ to $(\mu, \mathbf{\Sigma})$ where $\mathbf{c}$ is a vector containing Cr and Cb channel intensity values $[c_{Cr}, c_{Cb}]^T$ of a point $\mathbf{x}_n \in \mathbf{w}_m$. Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L\}$ be the set of candidate pixels for which the chrominance distance is less than a certain threshold. If the number of these candidate pixels, L, is larger than a predefined value, then we label that marker as visible in the current camera view and update its position as the mean of the points in $\mathbf{X}$ for the current camera view. The same process is repeated for all marker points in all camera views. Hence, we have the visibility information of each marker from each camera, and for those that are visible, we have the list of 2D positions of the markers on that specific camera image plane.

Once we scan the current scene from all cameras and obtain the visibility information for all markers, we start calculating the 3D positions of the markers by back-projecting the set of 2D points which are visible in respective cameras, using triangulation method. Theoretically, it is sufficient to see a marker at least from two cameras to be able to compute its position in 3D world. If a marker is not visible at least from two cameras, then its current 3D position is estimated from the information in the previous frame.

The 3D positions of markers are tracked over frames by Kalman filtering where the filter states correspond to 3D position and velocity of each marker. The list of 3D points obtained

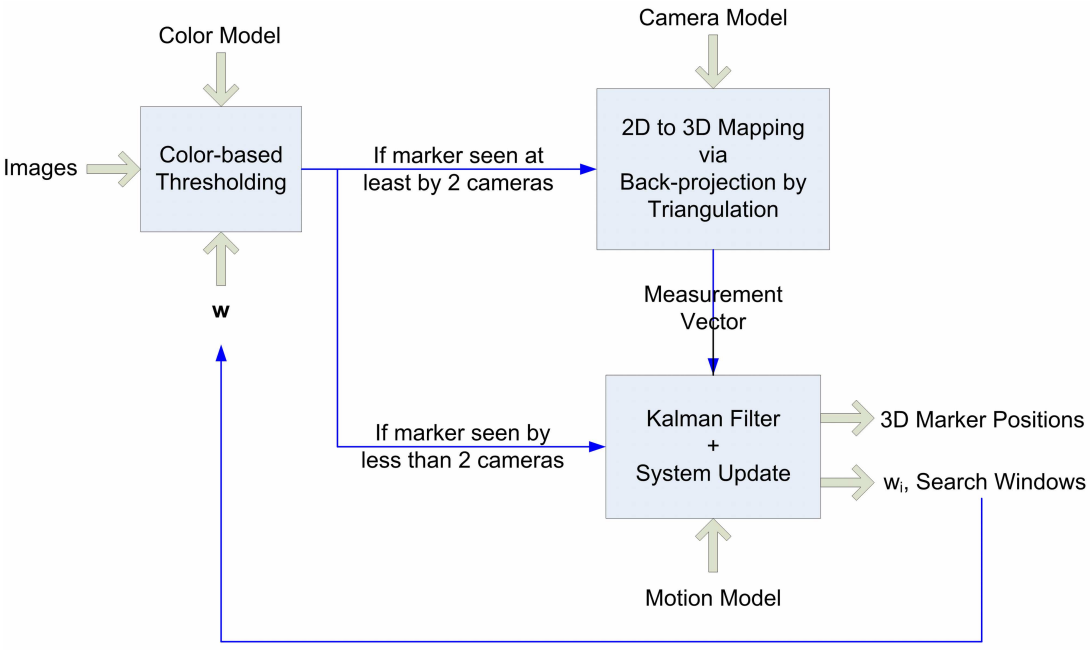


Figure 3.2: Block diagram of the proposed 3D joint positions tracking system.

by back-projection of visible 2D points in respective camera image planes constitutes the observations for this filter. This filtering operation has two purposes:

- to smooth out the measurements for marker locations in the current frame,
- to estimate the location of each marker in the next frame and to update the positioning of each search window, $\mathbf{w}_m$, on the corresponding image plane accordingly.

Fig. 3.2 summarizes the overall system for tracking 3D joint positions. Having updated the list of 3D joint positions for the current frame and estimated the location of the search windows for the next frame, we move on to the next frame and search the marker positions within the new search windows. This algorithm is repeated for the whole video. An instance of the 3D joint positions tracking process is shown in Fig. 3.3.

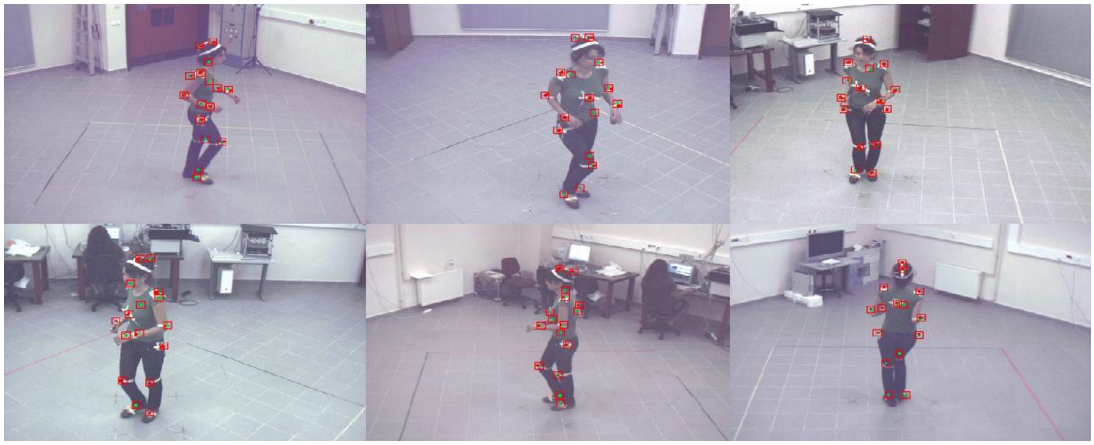


Figure 3.3: An example scene from the 3D joint positions tracking process. Red pixel regions in the red search windows represent the marker candidate pixels for the current frame. Green dots are the 2D projections of the 3D marker positions for the previous frame.

3.2 Extraction of Dance Motion Features

The motion capture process involves tracking a number of markers attached to the dancer's body as observed from multiple cameras and extraction of the corresponding motion features. Fig. 3.1 demonstrates our setting for this scenario. Markers in each video frame are tracked making use of their chrominance information. The 3D position of each marker at each frame is then determined via triangulation based on the observed projections of the markers on each camera's image plane. Therefore, the output of motion capture is a long list of 3D positions of each marker at each video frame. We consider two of the several ways of utilizing this output. In one way, we calculate 3D local displacements of each joint with respect to torso frame. In the other way, we compute the 3D joint angles at each frame from the global positions of the markers. The former set does not exactly create independent features whereas the latter set does. However, it is easier to algebraically calculate the 3D joint displacements than to calculate the 3D joint angles from 3D joint positions. On the other hand, most of the 3D character animations rely on joint angles rather than joint displacements.

3.2.1 3D Joint Displacements

3D joint displacements are extracted from 3D position vectors obtained by the proposed motion capture system. The displacement vectors are calculated relative to the reference frame after subtracting the rotational and translational motions which can be represented as a transformation matrix for the body as a whole. This transformation matrix is calculated using the torso which is composed of four points located on the hips, chest and back of the dancer. Points are defined in homogeneous coordinates such as $\mathbf{p}_i = [x_i \ y_i \ z_i \ 1]^T$. The transformation matrix is calculated relative to the first video frame. Let $\mathbf{M}^1 = [\mathbf{p}_1^1 \ \mathbf{p}_2^1 \ \mathbf{p}_2^1 \ \mathbf{p}_3^1]$ be 4×4 invertible matrix composed of initial locations of each torso joint. The locations of these points in the n^{th} frame can be given in a similar matrix format, $\mathbf{M}^n = [\mathbf{p}_1^n \ \mathbf{p}_2^n \ \mathbf{p}_2^n \ \mathbf{p}_3^n]$. The 4 transformation matrix $\mathbf{M}_p^n$ at n^{th} frame is calculated as $\mathbf{M}_p^n = (\mathbf{M}^n - \mathbf{m})(\mathbf{M}^1 - \mathbf{m})^{-1}$ where $\mathbf{m}$ is the mean of the points located on hips and shoulders in the first frame. Each initial point in the first frame is projected to the current frame by multiplying with the transformation matrix $\mathbf{M}_p^n$, and features are calculated as the differences of original point coordinates and the projected initial points, i.e., $\mathbf{f}_n^d = \mathbf{M}_p^n \times \mathbf{p}_i^n - \mathbf{p}_i^1$ where $\mathbf{p}_i^n$ and $\mathbf{p}_i^1$ are the locations of i^{th} point in n^{th} and initial frames, respectively. We also include the first and second differences of $\mathbf{f}_n^d$

$$\Delta\mathbf{f}_n^d = (1/2)\mathbf{f}_{n+1}^d + \mathbf{f}_n^d - (1/2)\mathbf{f}_{n-1}^d \tag{3.1}$$

$$\Delta^2\mathbf{f}_n^d = (1/2)\Delta\mathbf{f}_{n+1}^d + \Delta\mathbf{f}_n^d - (1/2)\Delta\mathbf{f}_{n-1}^d \tag{3.2}$$

to construct the desired dynamic 3D joint displacement feature vectors

$$\hat{\mathbf{f}}_n^d = [\mathbf{f}_n^{dT} \ \Delta\mathbf{f}_n^{dT} \ \Delta^2\mathbf{f}_n^{dT}]^T. \tag{3.3}$$

3.2.2 3D Joint Angles

In order to calculate 3D joint angles from 3D joint positions, we use a specialized commercial software package [45]. Our purpose is to fit a pre-designed 3D human body model to the set of 3D joint positions and let the aforementioned software calculate the desired set of 3D joint angles. The process of fitting a 3D human body model to the given set of 3D joint positions is outlined in Figure 3.4-(a). Usually, the process starts by manually fitting the character (actor) to a well-defined pose (ideally, a *T-pose*) to estimate dimensions at the

motion capture stage. Our original motion capture data did not include a T-pose, but we were still able to obtain acceptable results by using a similar pose selected from one of the output sequences. In order to determine how the motion capture data is to be interpreted, it is necessary to assign markers to *actor cells*. The set of assignments we have chosen to employ is depicted in Figure 3.4-(b), where the circles represent the cells. Some cells required more than one marker to behave properly during the animation. Once the body posture parameters, i.e., 3D joint positions, are successfully imported as motion capture data, we can extract the corresponding set of 3D joint Euler angles $\mathbf{f}_n^d$ for n^{th} motion frame from the model easily. However, angular features are generally discontinuous at boundary values due to their 2π-periodic nature and this situation causes a problem in training statistical models to capture the temporal dynamics of a sequence of angular features. Therefore, instead of using the static set of Euler angles $\mathbf{f}_n^d$, we just use their first and second differences as

$$\Delta\mathbf{f}_n^d = (1/2)\mathbf{f}_{n+1}^d + \mathbf{f}_n^d - (1/2)\mathbf{f}_{n-1}^d \tag{3.4}$$

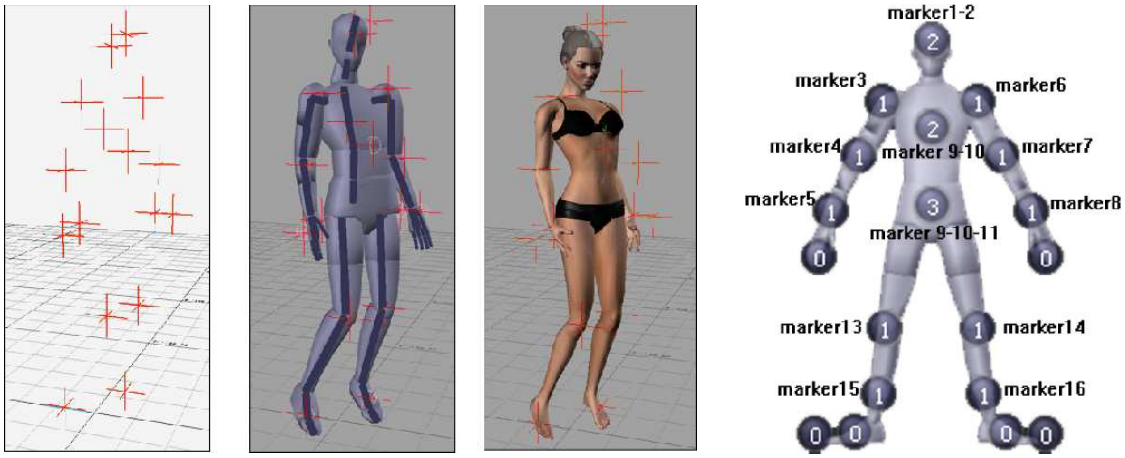
$$\Delta^2\mathbf{f}_n^d = (1/2)\Delta\mathbf{f}_{n+1}^d + \Delta\mathbf{f}_n^d - (1/2)\Delta\mathbf{f}_{n-1}^d \tag{3.5}$$

to form the set of dynamic 3D joint angle feature vectors

$$\hat{\mathbf{f}}_n^d = [\Delta\mathbf{f}_n^{dT} \ \Delta^2\mathbf{f}_n^{dT}]^T. \tag{3.6}$$

3.3 Extraction of Audio Features

One can consider the act of dancing as the natural response of the body to the rhythm of the music. Therefore, MFCCs are good choices for representing the music features since they perceive the sound as the human auditory system, which eventually shapes the movements of the body while dancing. Besides the MFCCs, chroma scale coefficients are also utilized to analyze the content of musical signals. On the other hand, it is crucial to note that among various features that characterize a musical audio signal, such as tonality, harmony or melody; tempo and measure are the ones that primarily drive and synchronize the dancing act. Hence, beat and measure analysis is also included in the audio feature extraction task.



(a) (b)

Figure 3.4: (a) Outline of the computation of 3D joint angles from motion data (b) Marker assignments

3.3.1 Mel-Frequency Cepstral Coefficients (MFCC)

In audio processing, the mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of an audio, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. Mel-frequency cepstral coefficients (MFCCs) are basically coefficients that collectively make up an MFC. The difference between the cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory perception system. This frequency warping can allow for better representation of sound, for example, in speech recognition and audio compression. We shall explain the stey-by-step computation of MFCC in this section.

1. Pre-emphasis: The speech signal $s(n)$ is sent to a high-pass filter:

$$s_2(n) = s(n) - as(n-1) \tag{3.7}$$

where $s_2(n)$ is the output signal and the value of a is usually between 0.9 and 1.0. The z-transform of the filter is

$$H(z) = 1 - az^{-1} \tag{3.8}$$

The goal of pre-emphasis is to compensate the high-frequency part that was suppressed during the sound production mechanism of humans. Moreover, it can also amplify the importance of high-frequency formants.

2. Frame Blocking: The input audio signal is segmented into frames of $20 \sim 30$ ms with optional overlap of $1/3 \sim 1/2$ of the audio frame size. Usually the frame size (in terms of sample points) is equal power of two in order to facilitate the use of fast Fourier transform (FFT). If this is not the case, we need to do zero padding to the nearest length of power of two. If the sample rate is 16 kHz and the frame size is 320 sample points, then the frame duration is 320/16000 = 0.02 sec = 20 ms. Additional, if the overlap is 160 points, then the frame rate is 16000/(320-160) = 100 frames per second.

3. Hamming Windowing: Each audio frame has to be multiplied with a hamming window in order to keep the continuity of the first and the last points in the frame (to be detailed in the next step). If the signal in a frame is denoted by $s(n),\ n = 0, \ldots, N-1$, then the signal after Hamming windowing is $s(n) * w(n)$, where $w(n)$ is the Hamming window defined by:

$$w(n, a) = (1 - \alpha) - \alpha \cos(2\pi n/(N-1)),\ 0 \leq n \leq N-1 \tag{3.9}$$

Different values of α corresponds to different curves for the Hamming windows and is typically set to 0.46.

4. Fast Fourier Transform (FFT): Spectral analysis shows that different timbres in audio signals corresponds to different energy distribution over frequencies. Therefore we usually perform FFT to obtain the magnitude frequency response of each frame. When we perform FFT on a frame, we assume that the signal within a frame is periodic, and continuous when wrapping around. If this is not the case, we can still perform FFT but the discontinuity at the frame's first and last points is likely to introduce undesirable effects in the frequency response. To deal with this problem, we have two strategies:

   - Multiply each frame by a Hamming window to increase its continuity at the first and last points.

- Take a frame of a variable size such that it always contains a integer multiple number of the fundamental periods of the speech signal.

The second strategy encounters difficulty in practice since the identification of the fundamental period is not a trivial problem. Moreover, unvoiced sounds do not have a fundamental period at all. Consequently, we usually adopt the first strategy to multiply the frame by a Hamming window before performing FFT.

5. Triangular Bandpass Filters: We multiple the magnitude frequency response by a set of 20 triangular bandpass filters to get the log energy of each triangular bandpass filter. The positions of these filters are equally spaced along the Mel-frequency, which is related to the common linear frequency f by the following equation:

$$mel(f) = 1125 \ln(1 + f/700) \tag{3.10}$$

Mel-frequency is proportional to the logarithm of the linear frequency, reflecting similar effects in the human's subjective aural perception.

6. Discrete Cosine Transform (DCT): In this step, we apply DCT on the 20 log energy $\mathrm{E_k}$ obtained from the triangular bandpass filters to have L mel-scale cepstral coefficients according to the following formula.

$$\mathrm{C_m} = \sum_{k=1}^{N} \cos\left[m\left(k - 0.5\right)\pi/N\right] \mathrm{E_k},\ m = 1, 2, \ldots, L \tag{3.11}$$

where N is the number of triangular bandpass filters, L is the number of mel-scale cepstral coefficients that form the coefficient vector $\mathbf{f}_n^m$, which denotes the static MFCCs for the n^{th} musical audio frame. Usually we set $N = 20$ and $L = 12$. Since we have performed FFT, DCT transforms the frequency domain into a time-like domain called quefrency domain. The obtained features are similar to cepstrum, thus it is referred to as the mel-scale cepstral coefficients, or MFCC. For better performance, we add the log energy and perform delta operation, as explained in the next two steps.

7. Log Energy: The energy within a frame is also an important feature that can be easily obtained. Hence we usually add the log energy as the 13^{th} feature to MFCC.

8. Delta Cepstrum: It is also advantageous to have the first and second time derivatives of the resulting coefficient vector $\mathbf{f}_n^m$, using the following regression formulas:

$$\Delta \mathbf{f}_n^m = \frac{\sum_{r=-2}^{2} r \mathbf{f}_{n+r}^m}{\sum_{r=-2}^{2} r^2} \tag{3.12}$$

$$\Delta^2 \mathbf{f}_n^m = \frac{\sum_{r=-2}^{2} r \Delta \mathbf{f}_{n+r}^m}{\sum_{r=-2}^{2} r^2} \tag{3.13}$$

We finally form the dynamic musical audio feature vector

$$\hat{\mathbf{f}}_n^m = [\mathbf{f}_n^{mT} \ \Delta \mathbf{f}_n^{mT} \ \Delta^2 \mathbf{f}_n^{mT}]^T \tag{3.14}$$

that also includes the first and second time derivatives.

3.3.2 Chroma-Scale Cepstral Coefficients (CSCC)

Unlike speech, music consists of a sequence of tones whose frequencies are already defined. Moreover, musical melody is a rhythmical succession of single tones in different patterns. Besides MFCCs, we extract chroma-based features that can be better used for modeling the melodic pattern in a measure segment with tone related features using temporal statistical models, HMMs. In order to represent musical scale, we project the entire spectrum onto 12 bins corresponding to the 12 distinct semi-tones of the musical octave. Theoretically, the frequency of the k-th note in the n-th octave is calculated as

$$f_k^n = f_0^0 2^{n+k/12} \tag{3.15}$$

where $f_0^0 = 16.35$ Hz, the pitch of the C0 note, and $n, k \in \mathrm{Z},\ 0 \leq k \leq 11$, based on Shepard's helix model [47]. In this study, we extract chroma features of 60 semi-tones for $4 \leq n \leq 8$ (over 5 octaves from the C4 note to the B8 note).

Rather than performing a short-time power spectrum analysis that has been commonly used for chroma feature extraction in [24] and [25], we prefer melscale analysis of musical audio signal for chroma feature extraction due to the logarithmic nature of the semi-tone frequencies. We extract chroma features by following an approach which is very similar to the MFCC calculation, explained in previous section. The difference is in how we choose the triangular overlapping windows while calculating the chromatic scale cepstral coefficients (CSCC) from the magnitude spectrum of DFT of the audio signal. We basically center

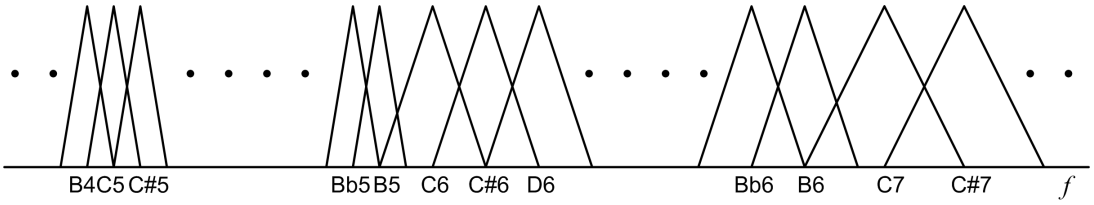


Figure 3.5: Triangular overlapping windows centered at the locations of semi-tone frequencies at different octaves during chroma features extraction.

the triangular weight windows at the locations of semi-tone frequencies at different octaves, given by equation (3.15) for $0 \leq k \leq 11$ and $4 \leq n \leq 8$, as shown in Figure 3.5. Then, we take log-average of the harmonics of the calculated semi-tone coefficients, that gives us the 12 CSCC features $\mathbf{f}_n^m$, representing the energy of each tone extracted from the musical audio frame n. We also compute the first and second time derivatives of these 12 CSCC features using the following regression formulas in equations (3.12) and (3.13) to form the dynamic music feature vector in equation (3.14).

3.3.3 Beat, Tempo and Measure

We estimate the tempo in terms of beats per minute (BPM) using the algorithms suggested in [48, 49]. Tempo estimation involves three basic tasks: onset detection, periodicity estimation and beat location estimation. Onset detection aims to point out where musical notes start, and tempo is established by the periodicity of the detected onsets. Beat location is computed directly from periodicity estimation.

First, onsets are detected based on the spectral energy flux of the input audio signal, that signifies one of the most salient features. Onset detection is determining, since beat tends to occur at onsets. Next, the periodicity is estimated from the detected onsets using an autocorrelation based method. Once the periodicity is determined, the tempo can be calculated in terms of BPM. Finally, beat locations are estimated by generating an artificial pulse train with the estimated periodicity and by cross-correlating it with the onset sequence. Maximum values of this function marks the starting of a beat location. See Figure 3.6 for an example of this process.

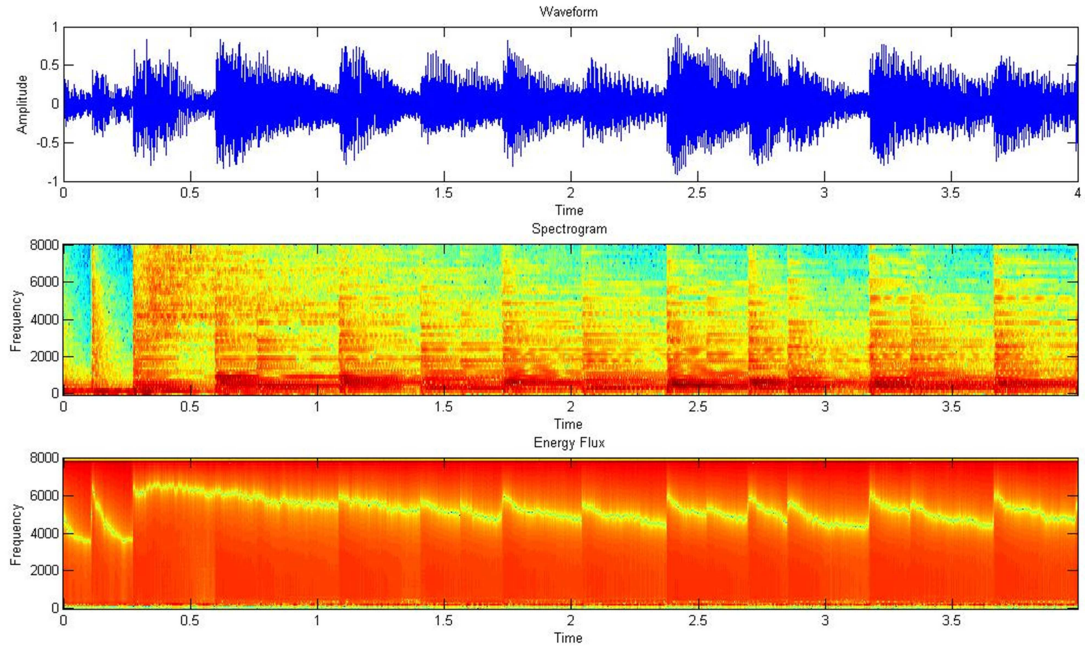


Figure 3.6: Beat detection example: time waveform, spectrogram and spectral energy flux of a sample 4-second music segment computed with 50% overlap analysis window.

Measure estimation gives an inference about the number of beats existing in between a measure in musical excerpt. Different rhythmic audio patterns are assumed to exist in each musical measure. More importantly, measure boundaries in general match the boundaries of dance figures.

3.4 Summary

This chapter introduced different sets of dynamic and static music and dance motion features that are used throughout this thesis. Specifically, we extracted chroma- and mel-scale cepstral coefficients as dynamic music features whereas we extract 3D joint angles and displacements as dynamic dance motion features. On the other hand, musical beat frequency and measure detection is used as static music features. In the following chapters, we will attack at the problem of multimodal dance motion analysis-synthesis from different perspectives. Each time, we will choose from different choices of feature types introduced in this chapter.

Chapter 4

UNSUPERVISED CORRELATION ANALYSIS OF MUSIC AND DANCE MOTION PATTERNS

Automatic dance analysis, annotation and synthesis have been studied extensively in the literature with emphasis on human body motion analysis-synthesis and dance music analysis. However, the correlation between the recurrent elementary dance motion and music patterns has attracted little attention. This chapter describes our first attempt to realize partially the multimodal framework explained in Chapter 2. That is, we will focus solely on the correlations between automatically extracted music and dance motion patterns [50, 51], using only the multimodal signal analysis part of the overall multimodal signal processing framework described in Chapter 2. We use 3D joint displacements (explained in Section 3.2.1) as our motion features and MFCC features (explained in Section 3.3.1) as our music features. In the multimodal analysis, we first perform an HMM based unsupervised temporal segmentation of the music and dance motion features to determine the recurrent elementary music and dance motion patterns. Then, we investigate the correlations between the resulting dance motion and music patterns to create a model that can be used towards estimation and synthesis of realistic music-driven dance animation. Section 4.1 describes the unsupervised temporal segmentation scheme and Section 4.2 outlines the multimodal correlation analysis approach. Experiments and results are demonstrated in Section 4.3. Section 4.4 summarizes the chapter.

4.1 Unsupervised Temporal Segmentation

We follow the basic approach introduced in Section 2.1 for unimodal unsupervised temporal segmentation of musical audio and dance motion into their meaningful elementary recurrent patterns. That is, we train a separate parallel HMM structure for each feature stream corresponding to each modality. The HMM structure $\mathbf{\Lambda}$ has M parallel branches and N states as displayed in Figure 2.3. The parallel HMM $\mathbf{\Lambda}$ is composed of M parallel left-to-right

HMMs, $\{\lambda_1, \lambda_2, \ldots, \lambda_M\}$, where each λ_m is composed of N states, $\{s_{m,1}, s_{m,2}, \ldots, s_{m,N}\}$. The state transition matrix $\mathbf{A}_{\lambda_m}$ of each λ_m is associated with a sub-diagonal matrix of $\mathbf{A}_\Lambda$. The feature stream is a sequence of feature vectors, $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_T\}$, where $\mathbf{f}_t$ denotes the generic feature vector at frame t. Unsupervised temporal segmentation using HMM model $\mathbf{\Lambda}$ yields L number of segments $\varepsilon = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_L\}$. The l^{th} temporal segment is associated with the following sequence of feature vectors,

$$\varepsilon_l = \{\mathbf{f}_{t_l}, \mathbf{f}_{t_l+1}, \ldots, \mathbf{f}_{t_{l+1}-1}\} \quad l = 1, 2, \ldots, L \tag{4.1}$$

where $\mathbf{f}_{t_1}$ is the first feature vector $\mathbf{f}_1$ and $\mathbf{f}_{t_{L+1}-1}$ is the last feature vector $\mathbf{f}_T$. The segmentation of the feature stream is performed using Viterbi decoding to maximize the probability of model match, which is the probability of feature sequence $\mathbf{F}$ given the trained parallel HMM $\mathbf{\Lambda}$,

$$\begin{aligned} \mathrm{P}(\mathbf{F}|\mathbf{\Lambda}) &= \max_{t_l, m_l} \prod_{l=1}^{L} \mathrm{P}(\{\mathbf{f}_{t_l}, \mathbf{f}_{t_l+1}, \ldots, \mathbf{f}_{t_{l+1}-1}\}|\lambda_{m_l}) \\ &= \max_{\varepsilon_l, m_l} \prod_{l=1}^{L} \mathrm{P}(\varepsilon_l|\lambda_{m_l}) \end{aligned} \tag{4.2}$$

where ε_l is the l^{th} temporal segment, which is modeled by the m_t-th branch of the parallel HMM $\mathbf{\Lambda}$. One can show that λ_{m_l} is the best match for the feature sequence ε_l, that is,

$$m_l = \operatorname*{argmax}_m \mathrm{P}(\varepsilon_l|\lambda_m). \tag{4.3}$$

Since the temporal segment ε_l from frame t_l to $(t_{l+1} - 1)$ is associated with segment label m_l, we define the sequence of frame labels based on this association as,

$$\ell_t = m_l \qquad \text{for } t = t_l, t_l + 1, \ldots, t_{l+1} - 1 \tag{4.4}$$

where ℓ_t is the label of the t^{th} frame and we have a label sequence $\ell = \{\ell_1, \ell_2, \ldots, \ell_T\}$ corresponding to the feature sequence $\mathbf{F}$. The first stage analysis extracts the frame label sequences ℓ^d and ℓ^m given the dance motion and musical audio feature streams $\mathbf{F}^d$ and $\mathbf{F}^m$, respectively.

The parallel HMM structure has two important parameters to set before the training of the model $\mathbf{\Lambda}$. The first parameter is the number of states in each branch, N. It should be selected by considering the average duration of temporal patterns. N is selected to be

$N_{\Lambda_d} = 10$, assuming minimum motion pattern duration is $\frac{1}{3}$ sec (10 frames). On the other hand, the number of temporal patterns for musical audio is set to $N_{\Lambda_m} = 5$ states in each branch of the musical audio HMM model $\mathbf{\Lambda}_m$ to model musical audio patterns.

The second parameter is the number of temporal patterns with the notation M. In order to find an optimum value for M, two fitness measures are checked where the first fitness measure, α, is the probability of model match and the second, β, is the average statistical separation between two similar temporal patterns. The value determined for M would be helpful for modeling the dance motion patterns. Therefore, the total number of temporal patterns, M, can be selected in the vicinity of the intersection of the normalized α and β measures. The definitions for these two measures are given below in equations.

$$\alpha = \frac{1}{T} \log(\mathrm{P}(\mathbf{F}|\mathbf{\Lambda})) \tag{4.5}$$

$$\beta = \frac{1}{T} \sum_{l=1}^{L} \log(\frac{\mathrm{P}(\varepsilon_l|\lambda_{m_l})}{\mathrm{P}(\varepsilon_l|\lambda_{m_l^*})}) \tag{4.6}$$

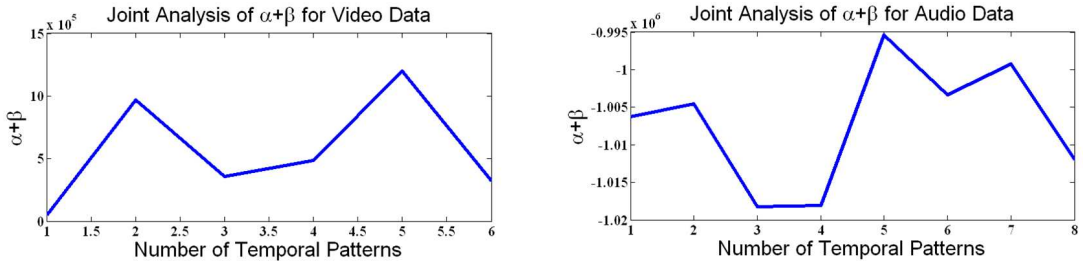
where $\lambda_{m_l^*}$ is the second best match for the temporal segment ε_l, that is given as,

$$m_l^* = \underset{\forall m \neq m_l}{\operatorname{argmax}} \mathrm{P}(\varepsilon_l|\lambda_m) \tag{4.7}$$

4.2 Multimodal Correlation Analysis

The first stage analysis defines elementary recurrent dance motion patterns for separate body parts using unsupervised temporal clustering over individual feature streams. The dance motion feature streams $\mathbf{F}^b$ are used to train HMM structure $\mathbf{\Lambda}_d$ that captures recurrent dance motion patterns ε^d. Musical audio feature streams $\mathbf{F}^m$ are similarly used to train HMM structure $\mathbf{\Lambda}_m$ to capture recurrent musical audio patterns ε^m. For ease of notation, we use a generic notation to represent the HMM structure which is identical for dance motion and musical audio streams.

In the second stage, we perform a joint analysis of dance motion-musical audio patterns and extract recurrent co-occurring patterns. This joint correlation analysis will be based on the co-occurrence matrix obtained from the co-occurring dance motion-musical audio events.



(a) (b)

Figure 4.1: Results of iterative approach for selection of the branch number,M, for (a) video and (b) audio.

4.3 Experiments and Results

Our training dataset includes multiview video recordings of only one dance performance, *zeybek*, with a duration of approximately 5 minutes. We take into account two different cases to analyze the dance motion events.

The first case considers the dance motion as a whole and models the movements of all parts of the body with a single HMM. That is, there are exactly two parallel HMMs to be trained: one for dance motion patterns and the other for musical audio patterns. Figure 4.1 shows that $M = 5$ maximizes α and β measures jointly. Hence, the parallel HMM models for dance motion and musical audio pattern analysis consist of 5 branches each.

Table 4.1 demonstrates the co-occurrence relation between the dance motion and musical audio patterns obtained as a result of our first stage analysis. Each row in the table displays the co-occurrence percentages of different musical audio patterns with dance motion patterns over the whole video. According to this co-occurrence matrix, the dance motion pattern V_e is the most repetitive one in our audiovisual data. Nevertheless, when we look at the co-occurrence relation of the first musical audio pattern, i.e. A_a, we see that it is also highly correlated with the dance motion patterns V_a. On the other hand, A_a never co-occurs with the dance motion patterns V_c and V_d.

The second case treats the whole dance motion as a union of motion of separate body parts, such as motion of right leg, motion of left arm, etc., and train models for each body part separately to determine the recurrent dance motion patterns in the first stage where

Table 4.1: Co-occurrence matrix for dance motion-musical audio events.

	V_a	V_b	V_c	V_d	V_e
A_a	40.43	8.51	0.00	0.00	51.06
A_b	5.49	12.09	13.19	6.59	62.63
A_c	10.99	2.20	0.00	4.95	81.86
A_d	0.00	2.94	0.00	2.94	94.12
A_e	22.22	8.55	28.21	4.27	36.75

the analysis process for musical audio remains the same as previous scenario.

Figure 4.2 shows the plots obtained for α and β measures of different body segments. For video, M is set as 3 which is in the vicinity of the intersection of the normalized α and β measures for separate dance motion patterns. Hence, our HMMs for dance motion pattern analysis consist of 3 branches each. On the other hand, Figure 4.3 shows us that $M = 6$ jointly maximizes α and β measures for the analysis of musical audio data.

Table 4.2 demonstrates the co-occurrence percentages between the left arm and the right arm motion patterns obtained as a result of our first stage analysis. Each row in the table displays the co-occurrence rates of different left arm motion patterns with right arm motion patterns over the whole video. According to this co-occurrence matrix, the left arm motion pattern L_a, L_b and L_c highly co-occurs with R_a, R_b and R_c, respectively. The dance figures related with both arm are labeled with same labels for similar figures where label a represents raising the arms up and then lowering them down, b occurs as holding the arms above the shoulder and c is observed as swinging arms forward and backward below shoulder.

Table 4.3 demonstrates the co-occurrence percentages between the left leg and right leg motion patterns obtained as a result of our first stage analysis. Similarly we can see that left and right arm are highly correlated and labels for similar figures are the same. Label

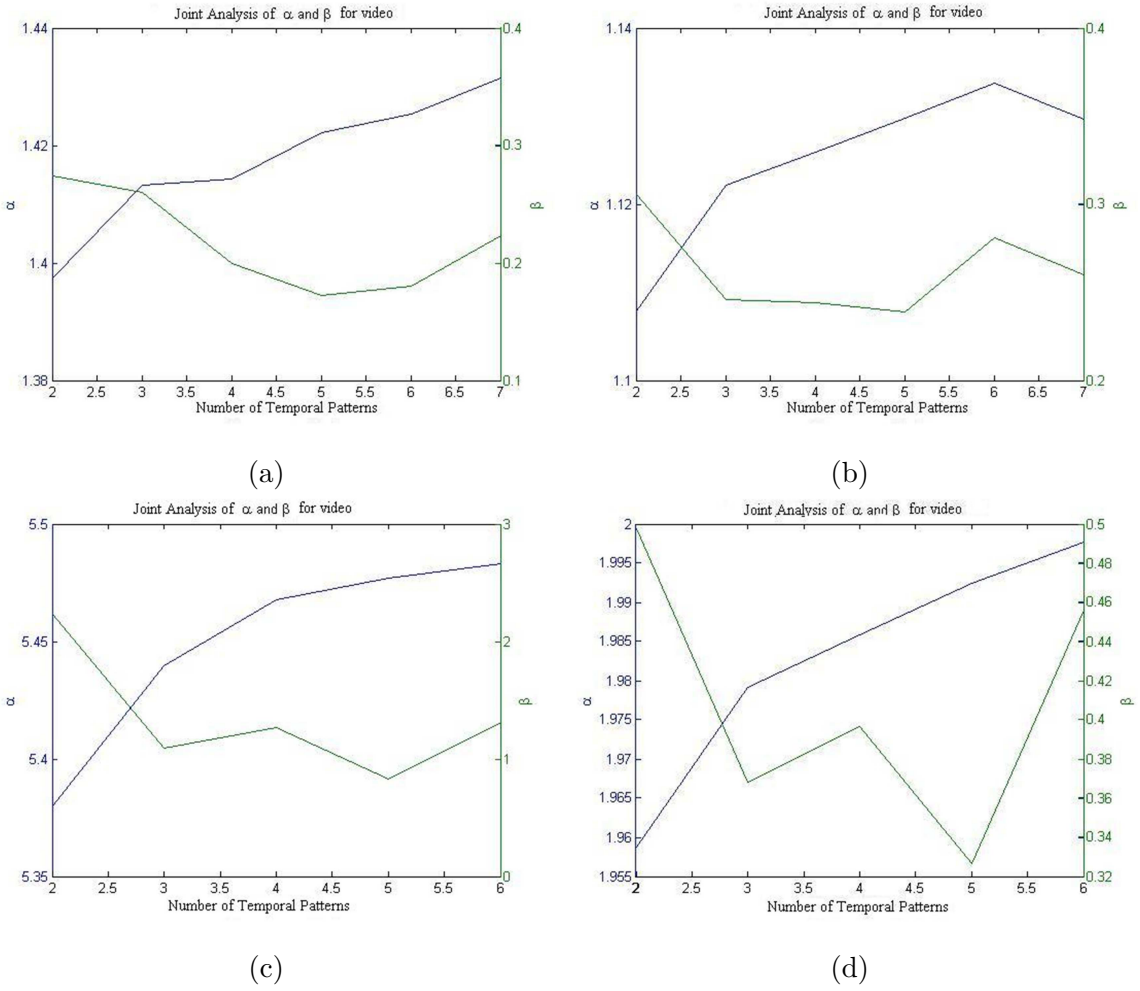


Figure 4.2: Results of iterative approach for selection of M for the dance motion patterns, upper left graphics is for for left leg and the upper right positioned graphics for right leg, left below graphics represents α and β measure for left arm and the graphics located right below represents for right arm.

a represents the act of standing at the same place with little bumps of legs, b occurs as pulling the legs up with big steps and c is observed as walking slowly. We can see from Table 4.4 that left leg and left arm has highly correlated patterns that co-occurs frequently. Nevertheless, we observe in Table 4.5 that right leg and right arm has highly correlated patterns that co-occurs frequently.

As a result of second stage analysis we investigated the correlation between dance motion

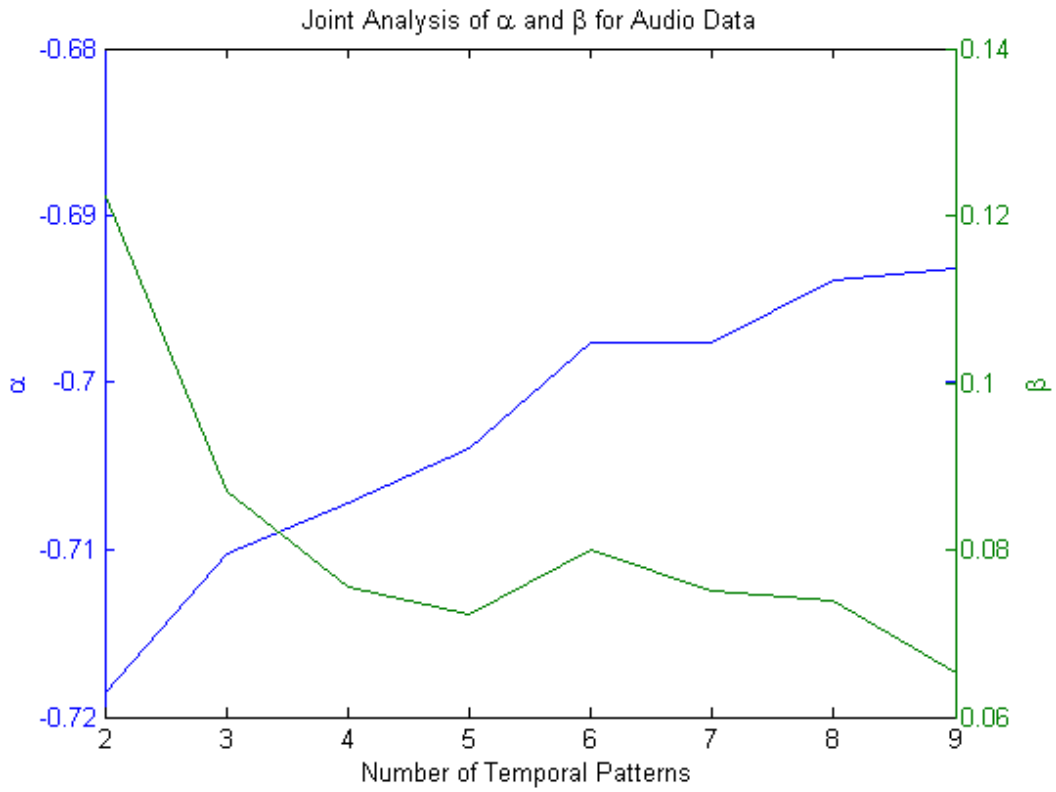


Figure 4.3: Results of iterative approach for selection of M for the musical audio data.

patterns and musical audio patterns. Table 4.6 gives the co-occurrence percentages of right leg and musical audio data patterns. Some motion patterns are highly correlated with musical audio patterns for instance $RArm_c$ highly co-occurs with musical audio pattern A_a where A_f is co-occurred with a small percentages with the same pattern.

4.4 Summary

Results of our analysis indicate that certain motion patterns are highly correlated with the musical audio channel. The co-occurrence tables tell us that arms are jointly correlated, legs are jointly correlated and arms and legs are correlated jointly, as well. The temporal patterns of correlated visual motion and audio should prove useful for synthetic agents and/or robots to learn dance figures from musical audio.

Table 4.2: Co-occurrence matrix for Left Arm-Right Arm events in percentages.

	$LArm_a$	$LArm_b$	$LArm_c$
$RArm_a$	95.65	0	4.35
$RArm_b$	0	100	0
$RArm_c$	16.67	8.33	75

Table 4.3: Co-occurrence matrix for Left Leg-Right Leg events in percentages.

	$LLeg_a$	$LLeg_b$	$LLeg_c$
$RLeg_a$	100	0	0
$RLeg_b$	0	100	0
$RLeg_c$	0	0	100

Table 4.4: Co-occurrence matrix for Left Arm-Left Leg events in percentages.

	$LLeg_a$	$LLeg_b$	$LLeg_c$
$LArm_a$	94.6	2.7	2.7
$LArm_b$	0	100	0
$LArm_c$	0	0	100

Table 4.5: Co-occurrence matrix for Right Arm-Right Leg events in percentages.

	$RLeg_a$	$RLeg_b$	$RLeg_c$
$RArm_a$	93.33	3.335	3.335
$RArm_b$	0	100	0
$RArm_c$	0	0	100

Table 4.6: Co-occurrence matrix for Left-Arm and musical audio patterns in percentages.

	A_a	A_b	A_c	A_d	A_e	A_f
$RArm_a$	10.64	25.53	19.86	12.06	9.22	26.69
$RArm_b$	21.13	19.01	24.29	11.97	6.69	16.90
$RArm_c$	38.71	10.11	2.81	4.93	8.45	0.35

Chapter 5

SUPERVISED AUDIOVISUAL ANALYSIS OF DANCE PERFORMANCES TOWARDS MUSIC-DRIVEN DANCE MOTION SYNTHESIS

This chapter describes our second attempt to realize the overall multimodal framework explained in Chapter 2, this time in its entirety. The automatic music-driven dance animation scheme is based on supervised modeling of music and dance motions [52, 53]. In this scheme, we consider a simplified dance scenario, where a dance performance is assumed to have only a single dance motion pattern, i.e., a dance figure, which is to be synchronized with the musical beat. Each dance figure in the training database is modeled in a supervised manner with a set of left-to-right HMM structures (see Figure 2.2) and the associated beat frequency. In the synthesis phase, an audio signal of unknown musical type is first classified, within a time interval, into one of the genres that have been learned in the analysis phase, based on MFCCs. The motion parameters of the corresponding dance figures are then synthesized via the trained HMM structures in synchrony with the audio signal based on the estimated tempo information. Finally, the generated motion parameters, i.e., 3D joint displacements, are animated along with the musical audio. The particular block diagram for the proposed scheme is depicted in Figure 5.1 and comprises three modules: multimodal analysis (training), audio-driven body motion synthesis and animation. The block diagram in Figure 5.1 is basically an extended and detailed version of the general multimodal framework presented in Figure 2.1 in Chapter 2. Section 5.1 outlines the dance motion and music analysis tasks, whereas music-driven dance motion synthesis is presented in Section 5.2. Experiments and results are demonstrated in Section 5.3. Finally, Section 5.4 summarizes the chapter.

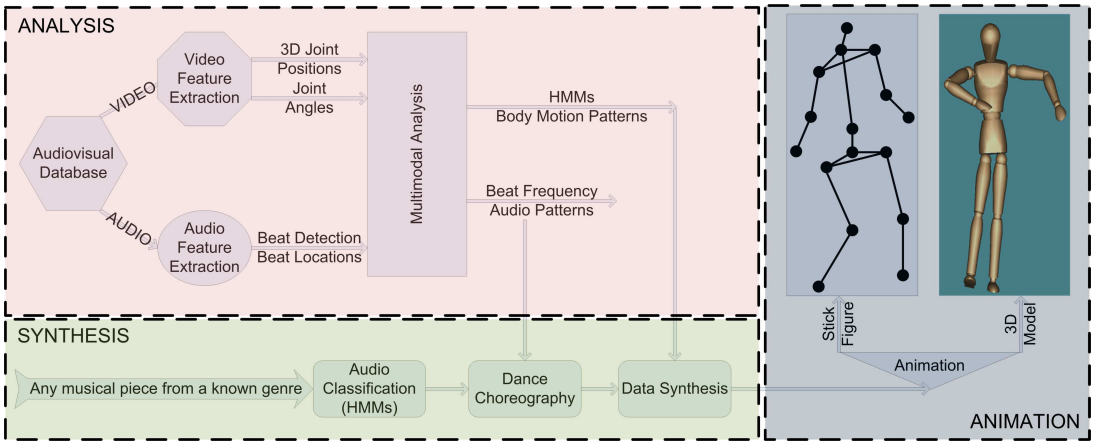


Figure 5.1: Block diagram of the supervised analysis-synthesis system.

5.1 Multimodal Dance Performance Analysis

5.1.1 Dance Motion Analysis

In the analysis block, multiview video sequences are analyzed in order to capture the time-varying posture of the dancer's body while audio is processed to extract beat information. The multiview videos are manually segmented into semantic recurring motion patterns: the dance figures. The corresponding body posture parameters are then used to train a set of HMMs, each of them modeling a different dance figure.

Human body motion analysis is tackled through HMMs. Dance motion is addressed by analyzing patterns that are repeated sequentially by the dancer and a set of HMMs is trained separately for each dance figure. Data employed to train the HMMs are the normalized 3D joint displacements which are extracted as explained in Section 3.2.1. For each figure, two sub-HMMs are defined to better capture the dynamics behavior of the upper and lower part of the body. The HMM modeling the upper part of the body addresses the arms movement (described by the (x, y, z) positions of the six landmarks placed in shoulders, elbows and wrists) while the other HMMs accounts for the legs (described by the (x, y, z) position for the six landmarks placed in hips, knees and ankles) (Figure 5.2).

To start evaluating the performance of the system presented in this report, a simple HMM is adopted. Typically, dance figures always contain a very concrete sequence of move-

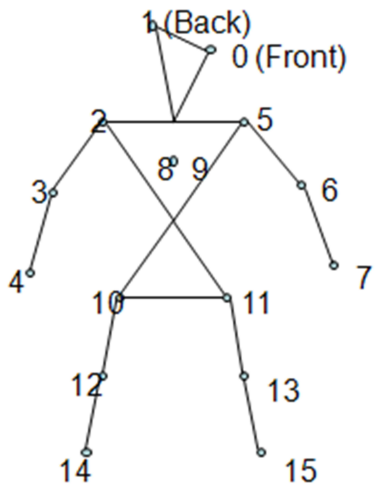


Figure 5.2: Markers positions (10 to 15 for lower body, 2 to 7 for upper body).

ments hence a left-right HMM structure is employed (Figure 2.2). Each of the parameters is represented by a single Gaussian function and one full covariance matrix is computed for each state. This rather simple scheme leads to satisfactory results hence no further complexity is added to the simple.

5.1.2 Audio Analysis

Since the audio and video sequences are synchronized, each repetition of a dance figure determines a time segment from which the beat frequency associated with the figure can be estimated. Analyzing the results from labeling of the dance figures in the video frames, we simply counted the number of beats per figure to estimate the beat frequency. We perform beat extraction task as explained in Section 3.3.3. We use this information during the synthesis to determine the beginning and ending frames of a dance figure.

5.2 Music-Driven Dance Motion Synthesis

The goal of the synthesis block, depicted in Figure 5.1, is to generate the corresponding body posture parameters synchronized with a test musical audio signal. The given musical audio signal is first classified, within a time interval, into one of the genres that have been

learned in the analysis part. For genre classification, we rely on MFCCs and employ the HMM-based classification technique described in [54]. The classified audio tracks are then analyzed to extract the beat and tempo information via the method explained in Section 3.3.3. The genre of the audio track determines the dance figure to be synthesized (recall that in the current scenario we are looking if there is only one single figure associated with each genre) whereas the beat locations and the tempo information determine the duration and location of the figure. We note that the beat frequency for the same dance figure may vary within a musical audio signal or from one piece to another.

5.2.1 Audio Classification

This part is a simple music genre classification problem. We have two types of music audio files where one is *salsa* and the other is *belly* dance. We use supervised HMMs and the MFCCs to discriminate between the input musical pieces. Use of MFCCs as the only musical audio feature set is sufficient for the classification problem, since we have only two kinds of audio files. For the extraction of parameters and classification steps, we use HTK toolkit [55].

Using the HMMs generated in the analysis step we first classify the input music audio files as *salsa* or *belly* dance as depicted in Figure 5.3, below. Then, we estimate the beat signal for the detected music audio file following the steps onset detection, periodicity estimation and beat location. Next, we identify the beat segmentation times in the music audio and determine the duration (in terms of frame numbers) of figures to be performed during the animation. Precalculated beats per frame information that we got in the analysis section is used for this purpose. For example, for *salsa*, each figure corresponds to a time segment of eight beats, so by multiplying the start and end time of the each segment with the number of frames per second (30 in our case), we simply get the beginning and ending frame numbers for *salsa* dance figures.

5.2.2 Body Motion Parameter Generation

Once we have the list of durations and types of consecutive dance figures in a file, we can use that file to generate the appropriate values for the animation parameters according to the

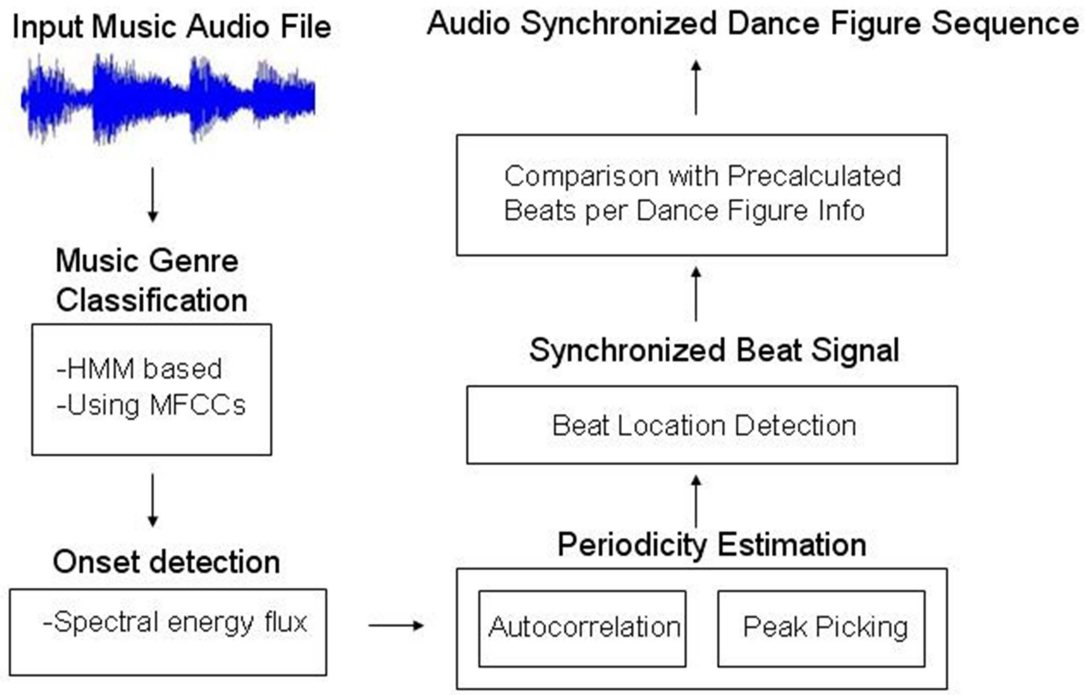


Figure 5.3: Audio processing steps in the synthesis part.

mean and standard deviation values of the corresponding HMM states. This file basically determines how much time each dance figure takes in the sequence. This helps us to allocate exactly the necessary amount of time to perform each dance figure.

Having the state sequences and the observation probabilities that are modeled as Gaussian distributions, the body posture parameters are generated along the state sequences associated with the corresponding Gaussian distribution at each state. The dance figure boundaries are overlapped and averaged in order to generate smoother figure-to-figure transitions. Finally, the generated body posture parameters are smoothed using median filtering followed by a Gaussian low-pass filter to remove motion jerkiness within a state and in the transition from one state to another.

It is crucial to note that the use of HMMs for dance figure synthesis provides us with the ability of introducing random variations in the synthesized body motion patterns for each dance figure. These variations make the synthesis results look more natural due to the fact that humans perform slightly varying dance figures at different times for the same

musical piece. Another important thing is that the use of HMMs for synthesis enables us to generate dance figures with varying durations in accordance with the beat information of the given musical audio signal.

5.2.3 Animation

We have designed a stick figure animation tool to visualize the output of the analysis-synthesis system. The stick figure animation is developed as an OpenGL based console application that is capable of animating a given set of point coordinates in 3D. The application can generate an animation of moving vertices without connecting them to each other. When the hierarchical connectivity information of the input point coordinates is available, the program generates the stick figure representation by connecting the neighboring vertices with edges. It also provides basic functionalities such as rotation, zooming in/out and panning the stick figure on the screen as well as capturing a single frame as an image or a sequence of frames as a video file. Despite depending on a simple idea, this tool proves to be useful when one wants to observe the success of the analysis-synthesis process, quickly and easily.

5.3 Experiments and Results

Our training dataset includes multiview video recordings of two dance performances, one for *salsa* and one for *belly*, each with a duration of approximately 5 minutes. The performances are recorded synchronously from 6 cameras at 30 fps. Each video recording consists of one single dance figure repeated successively during the whole performance.

For motion analysis, we manually label the start and end frames of each dance figure throughout the entire dance recordings. Recall that we have used 2 HMMs for training the 3D joint positions. These HMM models of each dance figure are trained in a supervised manner with the body posture parameters captured from the manually labeled segments.

In order to determine the optimal number of states for each of the HMMs, we train each HMM with different number of states (varying from 2 to 19). By computing the average logarithmic probability of the model match for each value, we examine the progression of the learning process and the accuracy of the trained model. The evolution of this parameter

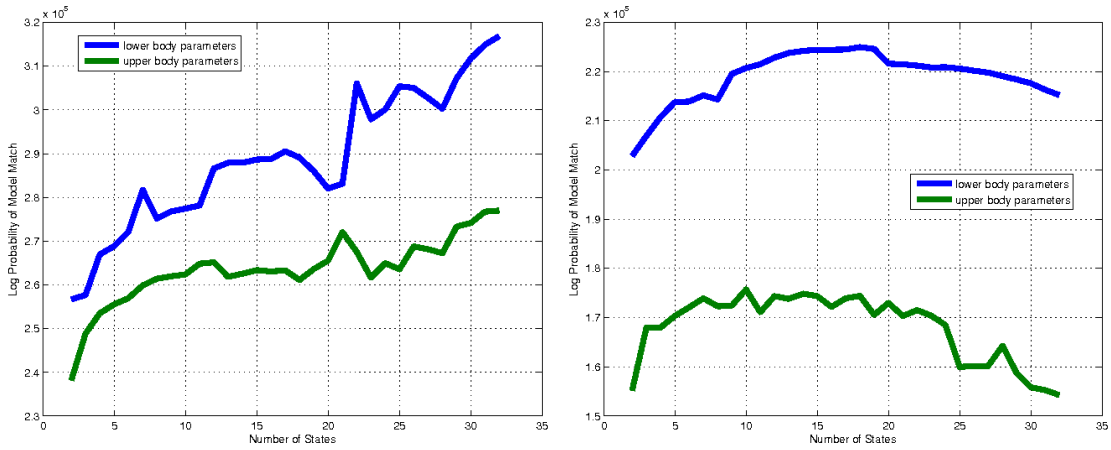


Figure 5.4: Evolution of the logarithmic probability of the model match with varying number of states for the 4 HMM structures in the case of 3D joint positions (two for *salsa* on the left and two for *belly* on the right).

for the totality of the 4 HMM structures that we trained is displayed in Fig. 5.4. We observe that the optimal number of states is related to the complexity of the dance figure. In the case of the *salsa* figure, which is more complicated than the *belly*, the optimal numbers are greater than those for the *belly* figure. To determine the optimal number of states, we basically search for the peak in the plot, or the point where the plots start to saturate since we also want to keep the number of states, and hence the model complexity, as low as possible.

In order to verify that the posture parameters are correctly modeled with the resulting HMMs, in Fig. 5.5 and Fig. 5.6, we compare, for some of the parameters, the evolution of the means of their Gaussian distributions over the HMM states with the evolution of the same parameters through the realizations of the corresponding dance figures in the training data set. The shapes of the evolution are clearly observed to be similar, even for the parameters which show significant variations from one realization to another in the training set and are thus difficult to model.

The musical audio signals are recorded at 16 kHz as 16 bit mono PCM wavefiles. The signals are analyzed over a 25 ms Hamming window at every 10 ms. The set of 13 MFC coefficients along with their first and second derivatives, adding up to a total of 39 features,

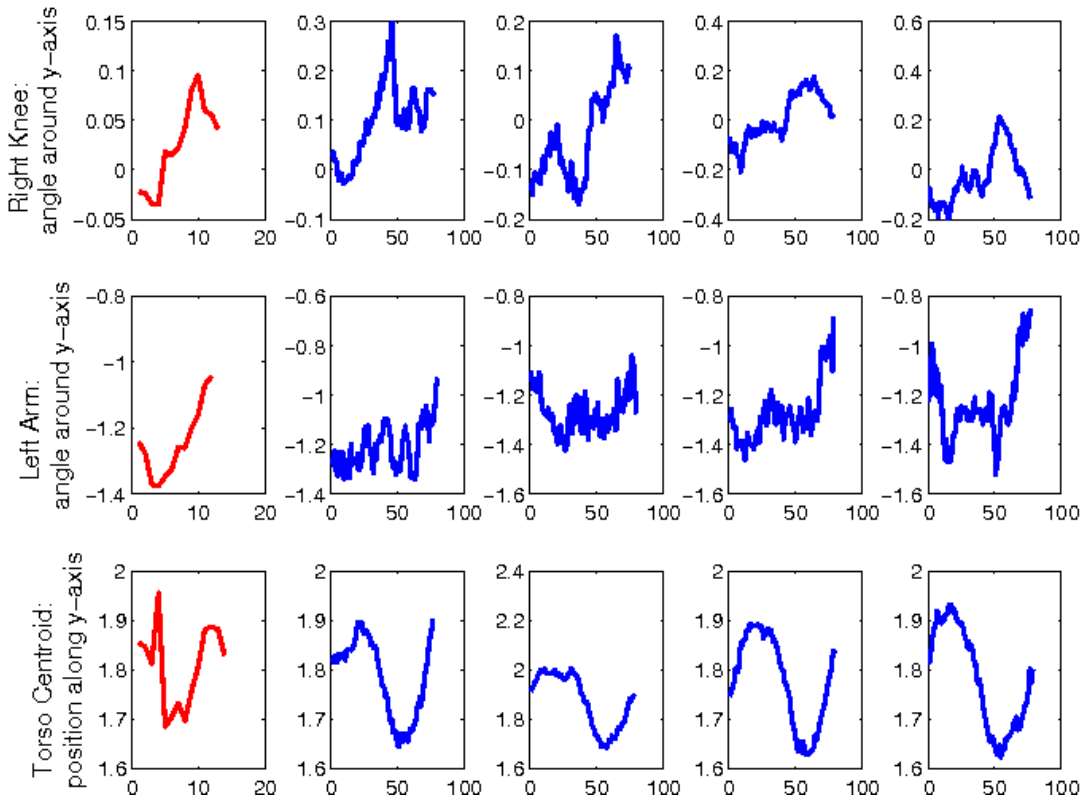


Figure 5.5: For the *salsa* figure, variation of the means of three parameters over the HMM states (plotted in red) and evolution of the same three parameters during four different realizations sampled from the training video (plotted in blue).

forms the audio feature vector for the genre classification task. Using MFCCs as the only audio feature set becomes sufficient for the classification problem in our case, since we have only two types of musical audio, *salsa* and *belly*. On the other hand, we concluded that each *salsa* figure corresponds to 8 beats in the *salsa* music audio file and each *belly* dance figure corresponds to 3 beats in the *belly* dance music.

We have considered several animation scenarios for demonstration of our dancing avatar. In the first scenario, we mix two audio tracks of different genres, *salsa* and *belly*, and use this mixed track as the animation audio to show that the avatar can successfully recognize the changing audio and synthesize the correct dance figures. In the second scenario, we first slow down and then speed up the audio track to demonstrate that the avatar can

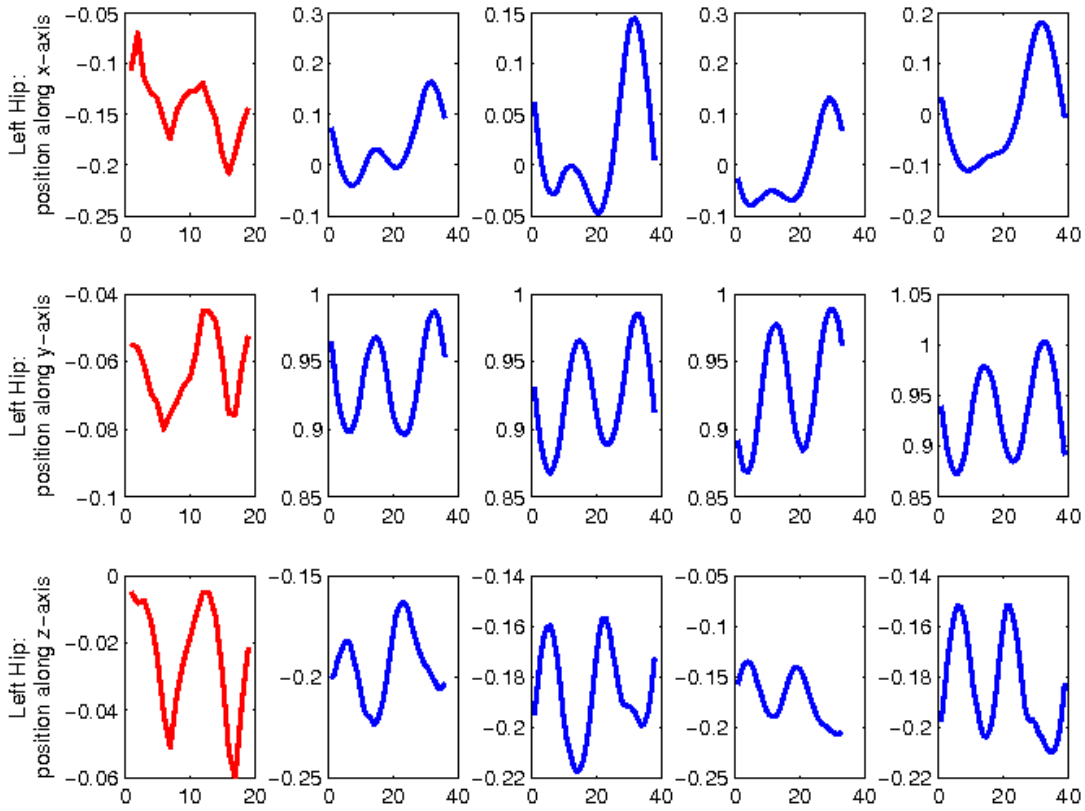


Figure 5.6: For the *belly* figure, variation of the means of three parameters over the HMM states (plotted in red) and evolution of the same three parameters during four different realizations sampled from the training video (plotted in blue).

keep track of the changing beat information and adjust the speed of the dance movements accordingly. In the final scenario, we take an arbitrary audio which is neither *salsa* nor *belly* to see how the avatar adapts itself to a different genre that it has not been trained for. We applied these three scenarios on analysis-synthesis results of the 3D joint positions parameter set. Demo videos of these scenarios are available online at *http://mvgl.ku.edu.tr/bodymotionanalysis/jmui/*.

5.4 Summary

It is crucial to note that the use of HMMs for dance figure synthesis provides us with the ability of introducing random variations in the synthesized body motion patterns for each

dance figure. These variations make the synthesis results look more natural due to the fact that humans perform slightly varying dance figures at different times for the same musical piece. Another important thing is that the use of HMMs for synthesis enables us to generate dance figures with varying durations in accordance with the beat and measure information of the given musical audio signal.

Results of our analysis-synthesis study shows that our system can successfully recognize the genre changes in a given audio track and synthesize the correct dance figures in a very realistic manner. It can also keep track of the changing beat information and adjust the speed of the dance movements accordingly.

Chapter 6

LEARNING STATISTICAL MUSIC-TO-DANCE MAPPINGS FOR CHOREOGRAPHY SYNTHESIS

In this chapter, a rather complete framework is proposed for modeling, analysis, annotation and synthesis of multimodal dance performances, which can handle more complex and realistic scenarios. The main objective is to automatically create a variety of synchronized dance performances that perceptually match the emotions and contents of the accompanying music; as if they were arranged by a choreographer. The proposed framework is based on learning many-to-many statistical mappings from musical measures (music segments) to dance figures (dance segments) towards generating plausible music-driven dance choreographies [56]. We assume that dance figures (dance segment boundaries) coincide with musical measures (music segment boundaries). For each training video, figure segments are manually labeled by an expert to indicate the type of dance motion. Motion trajectory of each dance figure is learned via hidden Markov model (HMM) based on 3D joint angles of the dancer's body for use in dance motion synthesis. Chroma features of each measure are used for music analysis. We model temporal statistics of such chroma features corresponding to each dance figure label to identify different harmonic (melodic) musical measure patterns for that dance figure. We employ a modified Viterbi algorithm for statistical music-driven choreography synthesis based on the correlations between dance figures and musical measures, as well as, the correlations between consecutive dance figures learned from the training dance video. The motion parameters of the dance figures in the synthesized choreography are then computed using the trained dance figure models. Finally, the generated motion parameters are animated synchronously with the musical audio using a 3D character model.

6.1 System Overview and Feature Extraction

The overall framework, as depicted in Figure 6.1, comprises of three parts: analysis, synthesis, and animation. The analysis part includes feature extraction and modeling modules

besides the data preparation module. In the data preparation module, input music stream is segmented by an expert into its units, i.e., musical measures. We use m_t to denote the measure segment at measure frame t. Measure segment boundaries are then used by the expert to define the motion units, i.e., dance figures. We use d_t to denote the dance figure segment corresponding to measure at frame t. The expert also assigns each dance figure d_t a figure label l_j to indicate the type of the dance motion. The collection of l_j forms the set of candidate dance figures, i.e., $\mathcal{L} = \{l_j | 1 \leq j \leq N\}$ where N is the number of distinct dance figure labels that exist in the training audiovisual dance database. Feature extraction modules compute the dance motion features $\mathbf{F}^{d_t}$ and music chroma features $\mathbf{F}^{m_t}$ for each d_t and m_t, respectively. The dance motion features $\mathbf{F}^{d_t}$ are used to train a hidden Markov model h_j^{d} for each dance figure label l_j in order to construct the set of dance figure models $\mathcal{H}^{\mathrm{d}}$. On the other hand, music chroma features $\mathbf{F}^{m_t}$ are used to train a hidden Markov model h_j^{m} for each dance figure label l_j in order to create the set of musical measure models $\mathcal{H}^{\mathrm{m}}$. Music chroma features $\mathbf{F}^{m_t}$ are also used to cluster measure segments m_t according to the harmonic similarity between different measure segments. Based on these measure clusters, we determine the group of dance figures that are accompanied by the musical measures with similar harmonic content. We then create the exchangeable figures model $\mathcal{X}$ based on such dance figure groups. In the meantime, the intrinsic dependencies of the dance figures l_j are captured by the choreography model $\mathcal{C}$.

The synthesis part makes use of the three models; namely, $\mathcal{X}$, $\mathcal{C}$, and $\mathcal{H}^{\mathrm{m}}$, to determine the output dance figure sequence $\tilde{\mathbf{r}}$ (i.e., choreography), from music chroma features which are extracted for a test input music. Here, $\tilde{\mathbf{r}} = \{\tilde{r}_t\}_{t=1}^{t=T}$, where $\tilde{r}_t \in \mathcal{L}$ and T is the number of musical measure segments. Specifically, the choreography synthesis module employs a modified Viterbi algorithm to determine the sequence of dance figures $\tilde{\mathbf{r}}$ subject to the exchangeable figures model $\mathcal{X}$ and the choreography model $\mathcal{C}$. Finally, body posture parameters corresponding to each dance figure in the synthesized choreography $\tilde{\mathbf{r}}$ are generated using the dance figure models $\mathcal{H}^{\mathrm{d}}$ (obtained in the analysis part) to animate a 3D character. The details about the multimodal dance figure analysis are presented in Section 6.2, whereas the music-driven dance choreography synthesis is explained in detail in Section 6.3. Section 6.4 presents the experiments and results; and finally, Section 6.5 outlines concluding remarks for the proposed framework.

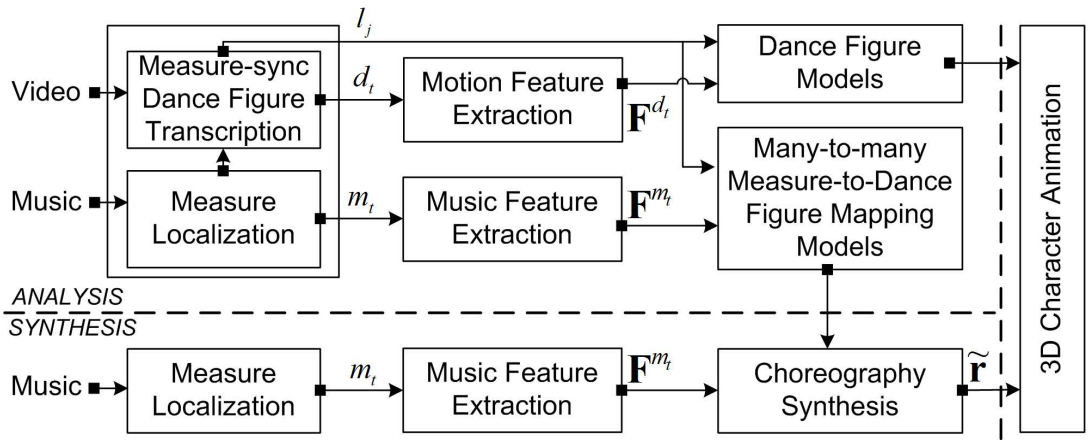


Figure 6.1: Block diagram of the overall multimodal dance performance analysis-synthesis framework.

6.1.1 Data Preparation

An audiovisual dance database can be pictured as a collection of measures and dance figures that are aligned in two parallel streams: music stream and dance stream. Figure 6.2 demonstrates a short excerpt from a musical piece as a collection of measures. In data preparation, a dance expert segments the input music and video streams into their units, i.e., measures m_t and dance figures f_t, respectively. Even though manual segmentation of video into its units (i.e., dance figures) is often more intuitive than manual segmentation of audio into its units (i.e., measures), we argue to do the opposite; that is, we perform a musical beat analysis based segmentation of music and video. For this purpose, we first make use of one of the recent automatic beat extraction algorithms proposed by Davies and Plumbley in [57] to the help the expert easily locate the measure boundaries, based on the extracted beat positions in a musical piece. Then, the expert checks the accuracy of manually marked measure boundary locations by analyzing whether the corresponding video segment is an acceptable dance figure, i.e., a meaningful compositional unit of dance. The expert also assigns each one of the resulting dance figure segments a label l_j to indicate the type of the dance movement. We regard the resulting sequence of dance figure labels l_j as the original (reference) choreography, i.e., $\mathbf{r} = \{r_t\}_{t=1}^{t=T}$, where $r_t \in \mathcal{L}$ and T is the

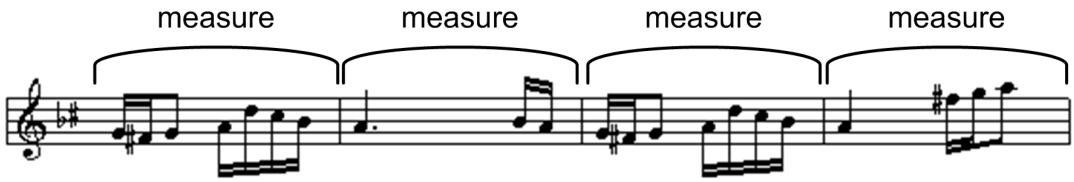


Figure 6.2: A musical piece is a collection of measures each of which has a different combination of musical notes.

number of musical measure segments. Reader should note that segmentation of a musical piece into its units (i.e., measures) is common to analysis and synthesis parts.

6.1.2 Music Features

We extract chroma features $\hat{\mathbf{f}}_n^m$ as explained in Section 3.3.2 to characterize the melodic or harmonic content of music. Each $\mathbf{F}^{m_t}$, therefore, corresponds to the set of music feature vectors $\hat{\mathbf{f}}_n^m$ that fall into the measure segment m_t. Specifically, $\mathbf{F}^{m_t}$ is a matrix of CSCC feature values in the form

$$\mathbf{F}^{m_t} = [\hat{\mathbf{f}}_1^m \ \hat{\mathbf{f}}_2^m \ \ldots \ \hat{\mathbf{f}}_{N_{m_t}}^m], \tag{6.1}$$

where N_{m_t} is the number of audio frames in measure segment m_t.

6.1.3 Dance Motion Features

We use the set of 3D joint angles extracted as explained in Section 3.2.2. We prefer joint angles as our dance motion features due to their widespread usage in human body motion analysis-synthesis and 3D character animation literature. We compute 60 angular values associated with 25 key joints of the body as well as 6 values for the global rotation and translation of the body, which leads to a dance motion feature vector $\hat{\mathbf{f}}_n^d$ of length 132 for each dance motion frame n. Moreover, each $\mathbf{F}^{d_t}$ is a collection of motion feature vectors $\hat{\mathbf{f}}_n^d$ that fall into dance figure segment d_t while training temporal models of motion trajectories associated with each dance figure label l_j. That is, $\mathbf{F}^{d_t}$ is a matrix of body motion feature values in the form

$$\mathbf{F}^{d_t} = [\hat{\mathbf{f}}_1^d \ \hat{\mathbf{f}}_2^d \ \ldots \ \hat{\mathbf{f}}_{N_{d_t}}^d], \tag{6.2}$$

where N_{d_t} is the number of dance motion frames within dance motion segment d_t. We also calculate the *mean trajectory* for each dance figure label l_j, namely μ_{l_j}, by calculating for each motion feature an average value over all instances (realizations) of the dance figures, which are labeled as l_j. These *mean trajectories* (μ_{l_j}) are required later in choreography animation since each dance figure model h_j^{d} capture only the temporal dynamics of the first and second differences of the Euler angles of the key joints associated with the dance figure label l_j.

6.2 Multimodal Dance Performance Analysis

In this section, we provide a detailed description of each model used in the proposed choreography analysis-synthesis framework. These models are: (i) Dance figure models $\mathcal{H}^{\mathrm{d}}$, which are used for parameter generation in choreography animation; (ii) Musical measure models $\mathcal{H}^{\mathrm{m}}$, which capture the many-to-one mappings from musical measures to dance figures using HMMs; (iii) Choreography model $\mathcal{C}$, which captures the intrinsic dependencies of dance figures; and (iv) Exchangeable figures model $\mathcal{X}$, which captures the one-to-many mapping from musical measures to dance figures, and hence, represents the subjective nature of the dance choreography with possibilities in the choice of dance figures and in their organization. Reader should note that the last three models are represented in a single block as "many-to-many measure-to-dance figure mapping models" in Figure 6.1.

6.2.1 Dance Figure Models ($\mathcal{H}^{\mathrm{d}}$)

The way a dancer performs a particular dance figure may exhibit variations in time in a dance performance. Therefore, it is important to model temporal statistics of each dance figure to capture the variations in the dance performance. Note that these models will also capture the personalized dance figure patterns of a dancer. We use the set of motion features $\mathbf{F}^{d_t}$ to train an HMM h_j^{d} for each dance figure label l_j to capture the dynamic behavior of the dancing body. Since a dance figure contains typically a well-defined sequence of body movements, we employ a left-to-right HMM structure (i.e., $a_{ik}^j \neq 0$ for $k = i, i+1$ where a_{ik}^j is the transition probability from state q_i^j to state q_k^j in h_j^{d}) to model each dance figure. Each motion parameter is represented by a single Gaussian function and one full covariance

matrix is computed for each h_j^{d}.

6.2.2 Musical Measure Models ($\mathcal{H}^{\mathrm{m}}$)

In a dance performance, musical measures that correspond to the same dance figure may exhibit variations and are usually a collection of different melodic patterns. That is, different melodic patterns can accompany the same dance figure, displaying a many-to-one mapping relation from musical measures to dance figures. We capture this many-to-one mapping by employing hidden Markov models (HMMs) to identify and model the melodic patterns corresponding to each dance figure. Specifically, we train an HMM h_j^{m} over the collection of measures co-occurring with the dance figure l_j. Here, musical measure features (i.e., chroma features) $\mathbf{F}^{m_t}$ are the observations and the dance figure labels l_j are the classes of the trained HMMs. Hence, we train as many HMMs as the number of different dance figures that exist in the dance performance. We define left-to-right HMM structures with $a_{ik}^j \neq 0$ for $k = i, i+1, i+2$ where a_{ik}^j is the transition probability from state q_i^j to state q_k^j in h_j^{m} for training models for the collection of measures. The transitions from state q_i^j to q_{i+2}^j account for the differences in measure durations. We use mixtures of Gaussians to model each parameter in chroma-based music feature vector and one diagonal covariance matrix is computed for each h_j^{m}. Using mixtures of Gaussians enables us to capture in a single model the different melodic patterns that correspond to a particular dance figure. We denote the collection of musical measure models as $\mathcal{H}^{\mathrm{m}}$, i.e., $\mathcal{H}^{\mathrm{m}} = \{h_j^{\mathrm{m}} | 1 \leq j \leq N\}$. Musical measure models $\mathcal{H}^{\mathrm{m}}$ provide us a tool to capture the many-to-one part of the many-to-many musical measure to dance figure mapping problem.

6.2.3 Choreography Model ($\mathcal{C}$)

Choreography model is built to capture the intrinsic dependencies of the dance figure sequences within the context of dance choreographies. Choreography model has two main contributors: i) figure-to-figure transition probabilities, and ii) probability of observing a musical measure feature sequence given a specific dance figure. The figure-to-figure transition probability density functions are modeled in n-gram language models, where the probability of the dance figure l_j at d_t given the dance figure sequence $i_1, i_2, ..., i_{n-1}$ at d_{t-1},

d_{t-2}, ..., d_{t-n+1}, i.e., $P(d_t = l_j | d_{t-1} = i_1, \ldots, d_{t-n+1} = i_{n-1})$, defines the n-gram dance language model. This model provides us a number of rules that specify the structure of a dance choreography. For instance, a dance figure that never appears after a particular sequence of $n-1$ dance figures in the training video does not appear in the synthesized choreography, either. We can also enforce a dance figure to always follow a particular sequence of $n-1$ dance figures if it is also the case in the training video with the help of the n-gram dance language model. The second contributor to the choreography model, i.e. the probability of music feature sequence $\mathbf{F}^{m_t}$ given a specific dance figure h_j^{m}, $P(\mathbf{F}^{m_t} | h_j^{\mathrm{m}})$, can be computed with the musical measure models $\mathcal{H}^{\mathrm{m}}$.

We define our choreography model $\mathcal{C}$ as a discrete HMM by taking the dance language model as a *bigram* model. The choreography model $\mathcal{C} = (\mathcal{A}, \mathcal{B}, \pi)$ can be described with the following parameters:

1. T, the number of time frames (measure segments). For each time frame (measure), the choreography synthesis process outputs exactly one dance figure label. Recall that we denote the individual dance figures as d_t and individual measures as m_t for $1 \leq t \leq T$.

2. N, the number of distinct dance figure labels, i.e., l_j where $1 \leq j \leq N$. Dance figure labels are the physical output of the process being modeled.

3. $\mathcal{A} = \{a_{ij}\}$, the dance figure transition probability distribution where

$$a_{ij} = P(d_t = l_j | d_{t-1} = l_i), \quad 1 \leq i, j \leq N \tag{6.3}$$

$$a_{ij} \geq 0, \quad 1 \leq i, j \leq N \tag{6.4}$$

$$\sum_{j=1}^{N} a_{ij} = 1, \quad 1 \leq i \leq N \tag{6.5}$$

4. $\mathcal{B} = \{b_t(j)\}$, the dance figure probability distribution for measure m_t where

$$b_t(j) = P(\mathbf{F}^{m_t} | h_j^{\mathrm{m}}), \quad 1 \leq j \leq N \tag{6.6}$$

$$1 \leq t \leq T \tag{6.7}$$

5. $\pi = \{\pi_i\}$, the initial dance figure distribution where

$$\pi_i = P(d_1 = l_i), \quad 1 \leq i \leq N \tag{6.8}$$

The choreography model $\mathcal{C}$ is the core of our choreography synthesis task and will be further investigated in Section 6.3.1. Note that, dance choreography synthesis can be performed based on Viterbi decoding on the choreography model $\mathcal{C}$. Furthermore, the token-passing algorithm [58] can be utilized for the choreography synthesis for higher order n-gram dance language models.

6.2.4 Exchangeable Figures Model ($\mathcal{X}$)

It is also possible in a dance performance that several distinct dance figures can be performed equally well along with a particular musical measure pattern, exhibiting a one-to-many mapping relation from musical measures to dance figures [11]. To capture this one-to-many mapping relation from musical measures to dance figures via learning exchangeable figure groups, we first compute the melodic similarity score s_{ij} between two different measure segments m_i and m_j as the local match score obtained from dynamic time warping (DTW) [59] of the chroma-based feature matrices $\mathbf{F}^{m_i}$ and $\mathbf{F}^{m_j}$, corresponding to m_i and m_j, respectively, in the musical piece S_k. Then, based on the melodic similarity scores between pairs of musical measure segments in S_k, we form an affinity matrix $\mathbf{Y}_k = \left(y_{ij}^k\right)_{i,j=1,\ldots,N}$ where $y_{ij}^k = \exp(-s_{ij})$ if $i \neq j$, and $y_{ii}^k = 0$. Finally, we apply the spectral clustering algorithm described in [60] over $\mathbf{Y}_k$ to cluster the measure segments in S_k. The spectral clustering algorithm in [60] assumes the number of clusters is known a priori and employs k-means clustering algorithm [61]. Since we do not know the number of clusters a priori, we measure the "quality" of the partition in the resulting clusters using the internal indexes, *silhouettes* [62], to determine the appropriate number of clusters. The silhouette value for each point is a measure of how similar that point is to points in its own cluster compared to points in other clusters, and ranges from -1 to +1. Averaging over all the silhouette values, we compute the overall quality of the clustering for a range of cluster numbers and pick the one that results in the highest silhouette.

We perform separate clustering for each musical piece S_k in order to increase the accuracy of musical measure clustering since similar measure patterns are likely to occur in the same musical piece rather than spread among different musical pieces. Once we obtain clusters of measures in all musical pieces, we can then use all of the measure clusters in all musical pieces to determine the exchangeable figures group $\mathcal{G}_j$ for each dance figure l_j by collecting the

dance figure labels that appear in the same cluster with l_j. Based on the exchangeable figure groups $\mathcal{G}_j$, we define the exchangeable figures model x_j as an indicator random variable:

$$x_j(i) = \mathrm{I}(l_i) = \begin{cases} 1, & \text{if } l_i \in \mathcal{G}_j \\ 0, & \text{otherwise} \end{cases} \tag{6.9}$$

where $\mathcal{G}_j$ is the exchangeable figure group associated with the dance figure l_j. The collection of x_j for all dance figure labels in $\mathcal{L}$ gives us the exchangeable figures model $\mathcal{X}$.

The notion of exchangeable figures is the key to reflect the subjective nature of the dance choreography with possibilities in the choice of dance figures and their organization throughout the choreography estimation process. The use of exchangeable figures model allows us to create a different artistic dance performance content each time we estimate a dance choreography.

6.3 Music-Driven Dance Choreography Synthesis

In this section, we address music-driven choreography synthesis and animation using the proposed multimodal dance figure models. The system takes music as input and produces first a sequence of dance figure labels, and then, generates the corresponding sequence of dance motion features, i.e., joint angle vectors, which are used to animate a 3D character model.

6.3.1 Choreography Synthesis

We formulate the choreography synthesis problem as estimating a dance figure sequence based on the choreography model $\mathcal{C}$ (described in Section 6.2.3). The lattice structure, say $\mathbf{M}$, of the discrete HMM that defines the choreography model $\mathcal{C}$ has the following properties:

- the vertical dimension represents the distinct dance figures labels,
- the horizontal dimension represents the time frames of music (i.e.,measures),
- vertices of the lattice are the acoustic scores associated with each dance figure label, based on the musical measure models $\mathcal{H}^{\mathrm{m}}$, i.e., $b_t(j) = P(\mathbf{F}^{m_t}|h_j^{\mathrm{m}})$,

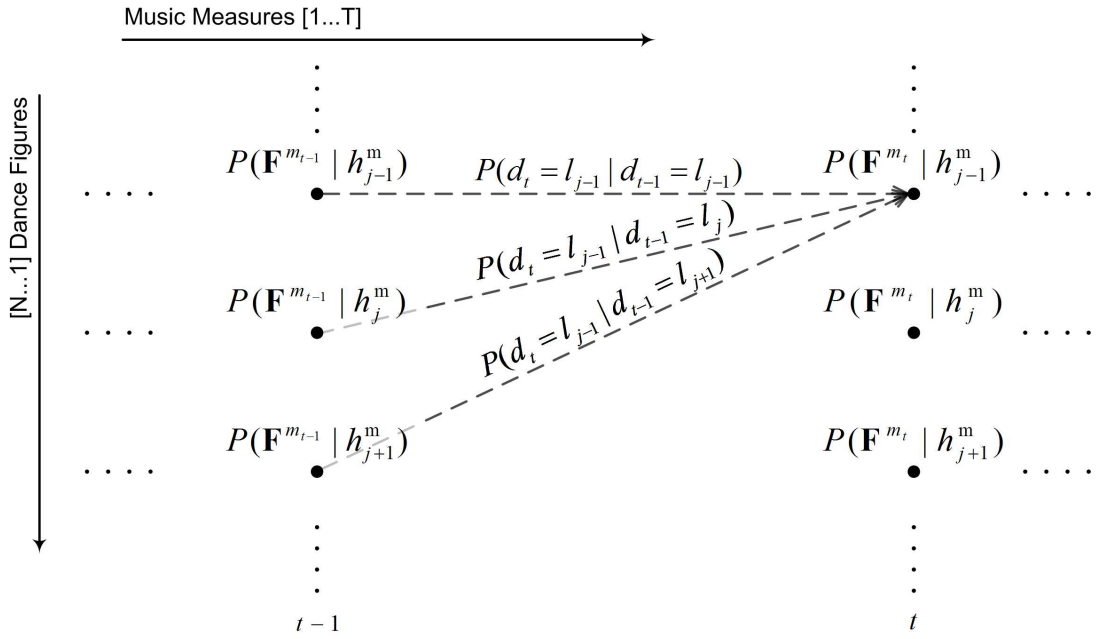


Figure 6.3: Lattice structure $\mathbf{M}$ of the choreography model $\mathcal{C}$.

- edges of the lattice are the figure-to-figure transition probabilities, based on the dance language model $\mathcal{A}$, i.e., $a_{ij} = P(d_t = l_j | d_{t-1} = l_i)$.

Figure 6.3 visualizes the lattice structure $\mathbf{M}$ of the choreography model $\mathcal{C}$.

At this point, the choreography synthesis problem can be seen as finding a path through the lattice $\mathbf{M}$. Therefore, assuming a uniform distribution for π, we employ a modified Viterbi algorithm to traverse through the lattice $\mathbf{M}$ to estimate an output dance figure sequence $\tilde{\mathbf{r}}$ subject to the exchangeable figures model $\mathcal{X}$ by finding a path along $\mathbf{M}$ in three different ways. In the first scenario, we follow the *single best path* along $\mathbf{M}$, i.e., the label sequence that has the maximum total likelihood.

Let $\phi_j(t)$ represent the partial likelihood score of performing the dance figure l_j at frame t along a single path that accounts for the highest partial likelihood from frame 1 to frame t. This partial likelihood can be computed efficiently using the following recursion:

$$\phi_j(t) = \max_i \{\phi_i(t-1)a_{ij}\} b_t(j). \tag{6.10}$$

At frame t, each partial likelihood score $\phi_j(t-1)$ is known for all dance figures l_j, hence (6.10) can be used to compute $\phi_j(t)$ thereby extending the partial paths by one time frame. Since the direct computation of likelihoods leads to underflow, we use log-likelihoods and rewrite the recursion in (6.10) as

$$\phi_j(t) = \max_i\{\phi_i(t-1) + \log(a_{ij})\} + \log(b_t(j)). \tag{6.11}$$

We also define a structure $\psi_j(t)$ to keep track of the argument which maximizes (6.11), for each j and t, in order to retrieve the dance figure sequence. The overall algorithm for finding the single best dance figure sequence can be summarized as follows:

1. Initialization:

$$\begin{aligned} \phi_j(1) &= b_1(j), \quad 1 \le j \le N \\ \psi_j(1) &= 0, \qquad 1 \le j \le N \end{aligned} \tag{6.12}$$

2. Recursion:

$$\begin{aligned} \phi_j(t) &= \max_i\{\phi_i(t-1) + \log(a_{ij})\} + \log(b_t(j)), \quad 2 \le t \le T \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad 1 \le j \le N \\ \psi_j(t) &= \operatorname{argmax}_i\{\phi_i(t-1) + \log(a_{ij})\}, \qquad\quad 2 \le t \le T \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad 1 \le j \le N \end{aligned} \tag{6.13}$$

3. Termination:

$$\begin{aligned} \Phi &= \max_i\{\phi_i(T)\} \\ \Psi(T) &= \operatorname{argmax}_i\{\phi_i(T)\} \end{aligned} \tag{6.14}$$

4. Path (dance figure sequence) backtracking:

$$\Psi(t) = \psi_{\Psi(t+1)}(t+1), \quad t = T-1, T-2, \ldots, 1 \tag{6.15}$$

$\Psi(t)$ in (6.15) stores the resulting dance figure label sequence as the desired output choreography $\tilde{\mathbf{r}}$. The resulting dance choreography $\tilde{\mathbf{r}}$ is expected to be unique in the first synthesis scenario since the proposed algorithm finds the optimal path along the lattice $\mathbf{M}$.

In the second synthesis scenario, we find a *likely path* along $\mathbf{M}$ in which we follow one of the *likely* partial paths in lieu of following the partial path that has the highest partial likelihood score at each time frame. For this purpose, we modify the algorithm described in the first synthesis scenario for finding the *single best path* along $\mathbf{M}$. Instead of picking the maximum in (6.11), we pick a partial path (indexed as i^*) among the top two "candidate" partial path transition scores (i.e., $\phi_i(t-1)+\log(a_{ij})$ over all i) by coin flipping to compute $\phi_j(t)$. Updating also the recurrence relation for $\psi_j(t)$ accordingly, the recursions in (6.11) become:

$$\begin{aligned} \phi_j(t) &= \phi_{i^*}(t-1) + \log(a_{i^*j}) + \log(b_t(j)), \\ \psi_j(t) &= i^*. \end{aligned} \tag{6.16}$$

The second synthesis scenario is expected to yield different dance choreographies due to sampling the distribution of partial path transition scores at each time frame, introducing variation into the choreography synthesis process.

The third synthesis scenario, on the other hand, requires some additional work based on the output of the first synthesis scenario for replacing each dance figure with a different choice from its exchangeable figure group (including itself) according to the exchangeable figures model $\mathcal{X}$ while also ensuring the optimality of the path (referred to as *exchangeable path*). Specifically, we go over the output of the first synthesis scenario figure by figure; and at each figure (i.e., time frame),

(i) we replace the figure with another one from its exchangeable figure group including itself, according to the distribution of acoustic scores $P(\mathbf{F}^{m_t}|h_j^{\mathrm{m}})$ of the dance figures $l_j \in \mathcal{G}_j$,

(ii) we update the rest of the figure sequence by determining a new single best path for the remaining time frames using the Viterbi algorithm.

We repeat these two steps until we reach the end of the dance figure sequence. In contrast to previous scenario, we constrain the collection of "candidate" dance figures that can replace

a particular dance figure in the choreography, say l_j, to those dance figures for which the exchangeable figures model x_j yields 1. It is possible to say that the third synthesis scenario is based on a smarter strategy than the second synthesis scenario to introduce variation into the choreography synthesis process.

Besides these three synthesis scenarios, we synthesize two more dance choreographies: one using only the musical measure models $\mathcal{H}^{\mathrm{m}}$ for identifying each measure segment in the test musical piece with a dance figure label (which we refer to as *only-acoustic* choreogprahy), and another one using only the n-gram figure-to-figure transition probabilities (which we refer to as *only-transition* choreography). The *only-acoustic* choreography corresponds to a synthesis scenario in which only the correlations between musical measures and dance figure labels are considered (but the correlations between consecutive figures are ignored). This is indeed the standard application of HMMs for musical measure identification. In contrast to *only-acoustic* choreography, the *only-transition* choreography corresponds to a synthesis scenario in which the dance figure for the next measure segment is predicted according to the distribution of the bigram transition probabilities associated with the dance figure at the present measure segment. The dance figure sequences resulting from these two scenarios constitute reference choreographies that help us investigate the benefits of incorporating the choreography model $\mathcal{C}$ and exchangeable figures model $\mathcal{X}$ into the choreography synthesis process, and evaluate the quality of the output dance choreographies resulting from the first three synthesis scenarios.

6.3.2 Character Animation

The synthesized choreography $\tilde{\mathbf{r}}$, (i.e., $\{\tilde{r}_t\}_{t=1}^{t=T}$), specifies the label sequence of dance figures to be performed with each measure segment whose duration is known beforehand in the proposed framework. Body posture parameters corresponding to each dance figure in the synthesized choreography $\{\tilde{r}_t\}_{t=1}^{t=T}$ are then generated such that they fit to the statistical dance figure models $\mathcal{H}^{\mathrm{d}}$, learned in the dance motion analysis part as explained in Section 6.2.1.

To generate body posture parameters using the dance figure model h_j^{d} for the dance figure l_j, we first determine the number of dance motion frames L required for the given segment duration. Next, we distribute the required number of motion frames L among the

states of the dance figure model h_j^{d} according to the expected state occupancy duration:

$$o_i^j = \frac{1}{1 - a_{ii}^j}, \qquad 1 \leq i \leq P \tag{6.17}$$

where o_i^j is the expected duration in state q_i^j, a_{ii}^j is the self-state-transition probability for state q_i^j (assuming $a_{ii}^j \neq 0$), and P is the number of states in the HMM h_j^{d}.

In order to avoid generation of noisy parameters, we first increase the time resolution of the dance motion by oversampling the dance motion model. That is, we generate parameters for a multiple of L, say KL where K is an integer scale factor, instead of L number of motion frames. Then, we generate the body motion parameters along the states of h_j^{d} according to the distribution of KL motion frames to these states, using the corresponding Gaussian distribution at each state. To reverse the effect of oversampling, we perform a downsampling by K that eventually yields smoother state transitions, and hence, more realistic parameter generation that avoid motion jerkiness.

As we mentioned in Section 6.1.3, the dance figure models h_j^{d} are trained over the first and second differences of the Euler angles of the joints. Thus, the generated parameters are basically the first and second differences of the Euler angles of the joints. Therefore, we need to simply sum the generated first differences with the *mean trajectory* associated with the dance figure l_j, i.e., μ_j, to obtain the final set of body posture parameters for l_j. Figure 6.4 depicts a synthesized trajectory against a sample trajectory from the database and the *mean trajectory* besides the expected state duration boundaries and the state means for one of the dance figures in the audiovisual dance database.

After repeating the described procedure for each dance figure in the synthesized choreography, the body posture parameters at the dance figure boundaries are smoothed via cubic interpolation within a Δ-neighborhood of each dance figure boundary in order to generate smoother figure-to-figure transitions.

It is crucial to note that the use of HMMs for dance figure synthesis provides us with the ability of introducing random variations in the synthesized body motion patterns for each dance figure. These variations make the synthesis results look more natural due to the fact that humans perform slightly varying dance figures at different times for the same musical piece.

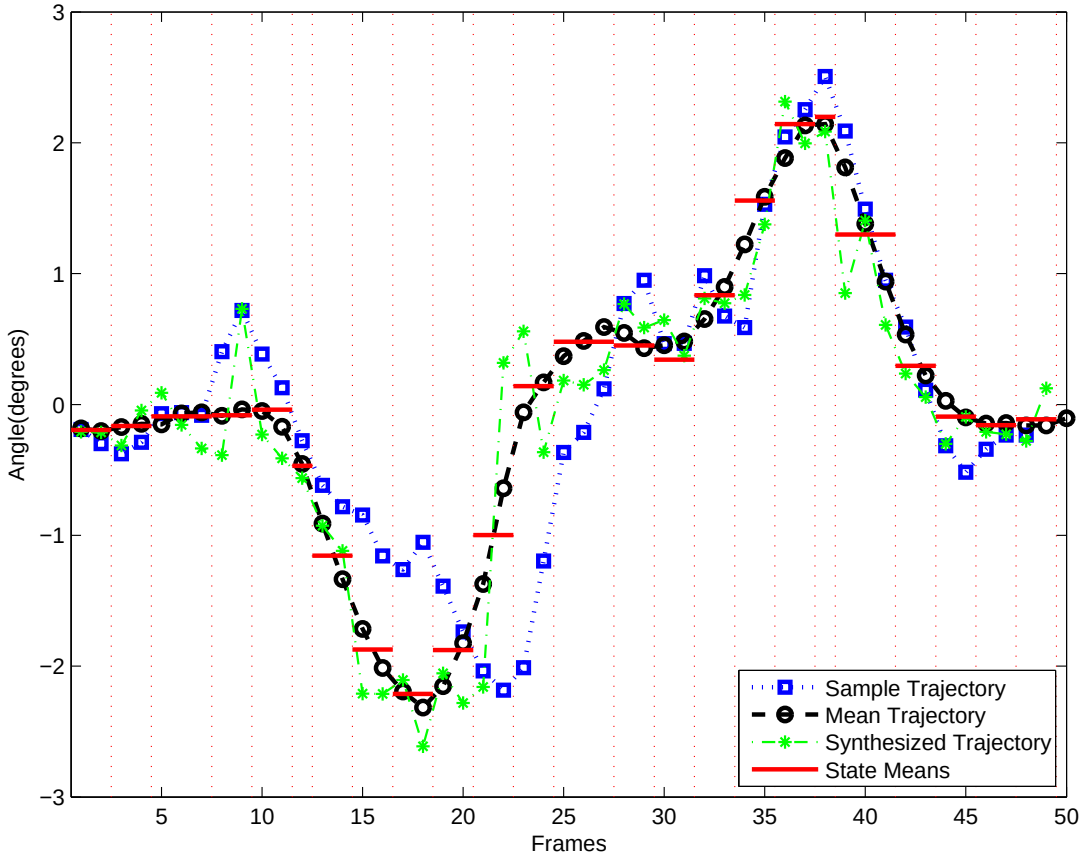


Figure 6.4: A synthesized trajectory is compared with a sample trajectory from the database and the *mean trajectory* as well as the expected state duration boundaries and the state means; all associated with the same dance figure.

6.4 Experiments and Results

In this study, we investigate the Turkish folk dance, *kasik*[1]. Our audiovisual database is 36 minutes long and consists of 20 dance performances with 20 different musical pieces. There are 31 different dance figures (i.e., $N = 31$) and a total of 1258 musical measure segments (i.e., $T = 1258$).

Table 6.1 shows the distribution of dance figures to different musical pieces. Each entry in the first column is a dance figure label l_j and each entry in the first row is a musical piece

[1] *Kasik* means *spoon* in English. The dance is named so because the dancers clap spoons while dancing.

S_k. Hence, multiple nonzero entries in a row shows that the same figure can be performed with different melodic patterns whereas nonzero entries in a column shows that different dance figures can be performed with the same melodic pattern. Therefore, Table 6.1 is a means of evidence that there is a many-to-many mapping between dance figures and musical measures.

Table 6.2, on the other hand, lists the exchangeable figure groups $\mathcal{G}_j$ for each dance figure l_j, obtained as explained in Section 6.2.4. For instance, dance figure l_1 can be performed in places where l_2 is performed, or vice versa, and the change of places between these two figures creates a different but still acceptable choreography according to our exchangeable figures notion. The notion of exchangeable figures will also be useful in evaluating the choreography synthesis output as we will explain later in this section.

We follow 5-fold cross-validation procedure for measure analysis task. We train musical measure models using four fifth of the musical audio data in the analysis part and use these musical measure models in the process of choreography estimation for the remaining one fifth of the musical audio data in the synthesis part. We repeat this procedure five times, each time using different parts of the musical audio data for training and testing. This way, we synthesize a new dance choreography for the entire musical audio data.

6.4.1 Objective Evaluation Results

We define the following four assessment levels to evaluate each dance figure label $\tilde{r}_t$ in the synthesized figure sequence $\tilde{\mathbf{r}}$ compared to the respective figure label r_t in the original dance choreography $\mathbf{r}$, assigned by the expert:

- $L0$ (*Exact-match*): $\tilde{r}_t$ is marked as $L0$ if $\tilde{r}_t$ matches r_t.
- $L1$ (*X-match*): $\tilde{r}_t$ is marked as $L1$ if $\tilde{r}_t$ does not match r_t, but it is in r_t's exchangeable figure group $\mathcal{G}_{r_t}$; i.e., $\tilde{r}_t \in \mathcal{G}_{r_t}$.
- $L2$ (*Song-match*): $\tilde{r}_t$ is marked as L_2 if $\tilde{r}_t$ neither matches r_t nor is in $\mathcal{G}_{r_t}$; but, $\tilde{r}_t$ and r_t are performed with the same musical piece; i.e., $\{\tilde{r}_t, r_t\} \in S_k$.
- $L3$ (*No-match*): $\tilde{r}_t$ is marked as L_3 if it is not marked as one of $L0$ through $L2$.

Table 6.1: Distribution of the dance figures to musical pieces

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}	S_{17}	S_{18}	S_{19}	S_{20}
l_1	29																			
l_2	34		5			13											11		32	43
l_3		7		20				7		8				6						
l_4							19													
l_5	12		27			7	12		27								32		4	14
l_6		7		4						4				11						
l_7	3																	7		
l_8				4										7		3				
l_9			3														12		8	5
l_{10}		9								20		2								
l_{11}			12										12							
l_{12}				4				2						4		5		4		
l_{13}													16							
l_{14}	3		11																	
l_{15}	2		12																	
l_{16}					17						24	14								
l_{17}														14						
l_{18}							8	18	20								12			47
l_{19}								4		14						4				
l_{20}				12										8		3				
l_{21}			16						30								18			
l_{22}											10									
l_{23}													62							
l_{24}															41					
l_{25}															41					
l_{26}																19				
l_{27}																		20		
l_{28}											10	5								
l_{29}																		10		
l_{30}	1								1	1			8							
l_{31}		29		22			21						55		1				53	

Table 6.2: List of figures and their exchangeable figure groups.

l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}	l_{11}	l_{12}	l_{13}	l_{14}	l_{15}	l_{16}	l_{17}	l_{18}	l_{19}	l_{20}	l_{21}	l_{22}	l_{23}	l_{24}	l_{25}	l_{26}	l_{27}	l_{28}	l_{29}	l_{30}	l_{31}
l_2	l_1	l_6	l_5	l_1	l_3	l_1	l_3	l_2	l_3	l_2	l_3	l_{11}	l_1	l_1	l_{10}	l_6	l_2	l_6	l_3	l_2	l_{16}	l_{11}	l_{25}	l_{24}		l_7	l_{10}			
l_5	l_5	l_8	l_{18}	l_2	l_8	l_2	l_6	l_5	l_6	l_5	l_6	l_{23}	l_2	l_2	l_{22}	l_8	l_3	l_{12}	l_6	l_5	l_{28}	l_{13}				l_{12}	l_{16}			
l_7	l_7	l_{10}	l_{31}	l_4	l_{10}	l_{14}	l_{12}	l_{11}	l_{16}	l_9	l_8		l_5	l_5	l_{28}	l_{20}	l_4		l_8	l_9							l_{22}			
l_{14}	l_9	l_{12}		l_9	l_{12}	l_{27}	l_{17}	l_{14}	l_{28}	l_{13}	l_{19}		l_7	l_9			l_5		l_{12}	l_{11}										
l_{15}	l_{11}	l_{18}		l_{11}	l_{17}		l_{20}	l_{15}		l_{14}	l_{20}		l_9	l_{11}			l_{21}		l_{17}	l_{14}										
	l_{14}	l_{20}		l_{14}	l_{19}			l_{21}		l_{15}	l_{27}		l_{11}	l_{14}						l_{15}										
	l_{15}			l_{15}	l_{20}					l_{21}			l_{15}	l_{21}						l_{18}										
	l_{18}			l_{18}						l_{23}			l_{21}																	
	l_{21}			l_{21}																										

Figure 6.5 displays all assessment levels in a single confusion matrix. We also associate a penalty score ranging from 0 to 3 with the levels $L0$ through $L3$, respectively. Then, we calculate an overall penalty score for measuring the "goodness" (coherence) of the resulting dance choreography. According to this scheme, low penalty scores indicate good choreography synthesis results. Recall that we estimate alternative choreographies in five different ways, as explained in Section 6.3.1. Figure 6.6 compares the number of figures that fall into each assessment level for all synthesis scenarios. The penalty scores for the *only-acoustic* and *only-transition* choreographies are 1036 and 2602, respectively. The penalty score for the *likely path* choreography is 1144, which is slightly higher than the penalty score of the *only-acoustic* choreography whereas it decreases to 705 and 796 for the *best path* and *exchangeable path* choreographies, respectively. Among all synthesis scenarios, the *best path* synthesis scenario results in a choreography with the smallest penalty score as expected. Introducing variations into the output dance choreography, the *exchangeable path* synthesis scenario performs slightly worse than the *best path* synthesis scenario, however, its penalty score is still much smaller than the penalty scores of the reference *only-acoustic* and *only-transition* choreographies. We see that *best path* and *exchangeable path* scenarios are successful at decreasing the number of dance figures that fall into $L3$ in reference choreographies *only-acoustic* and *only-transition*.

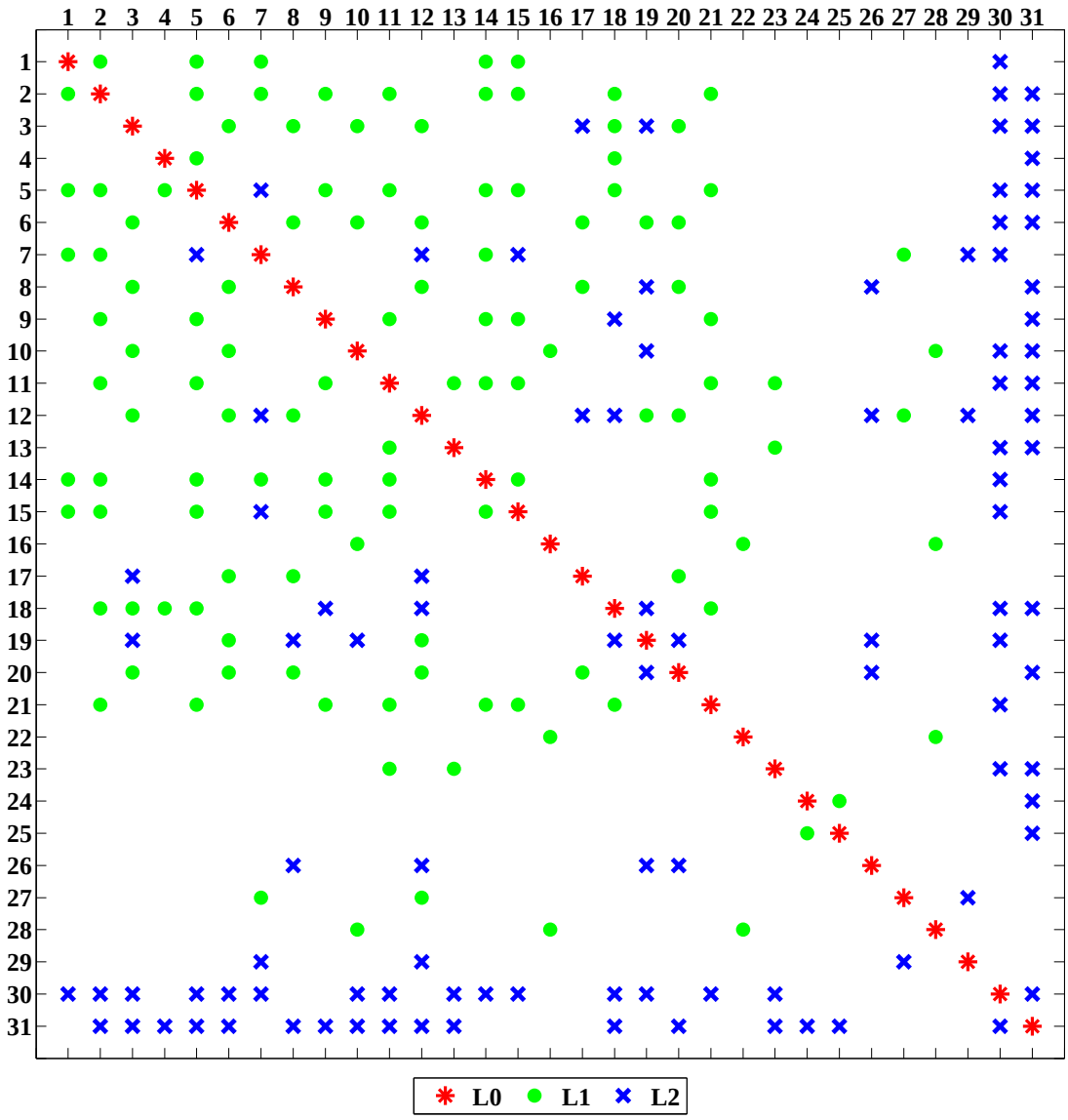


Figure 6.5: All assessment levels are put into a single confusion matrix. The empty entries of this matrix correspond to assessment level $L3$.

Looking at Figure 6.6 from another point of view, we see that among all the assessment levels, $L0$ through $L2$ are indicators of the diversity of alternative dance figure choreographies rather than being an indicator of error. $L3$, however, indicates an error in the dance choreography synthesis process. In this context, we see that only ∼49% of the *only-transition* choreography and only ∼85% of the *only-acoustic* choreography fall into the first three as-

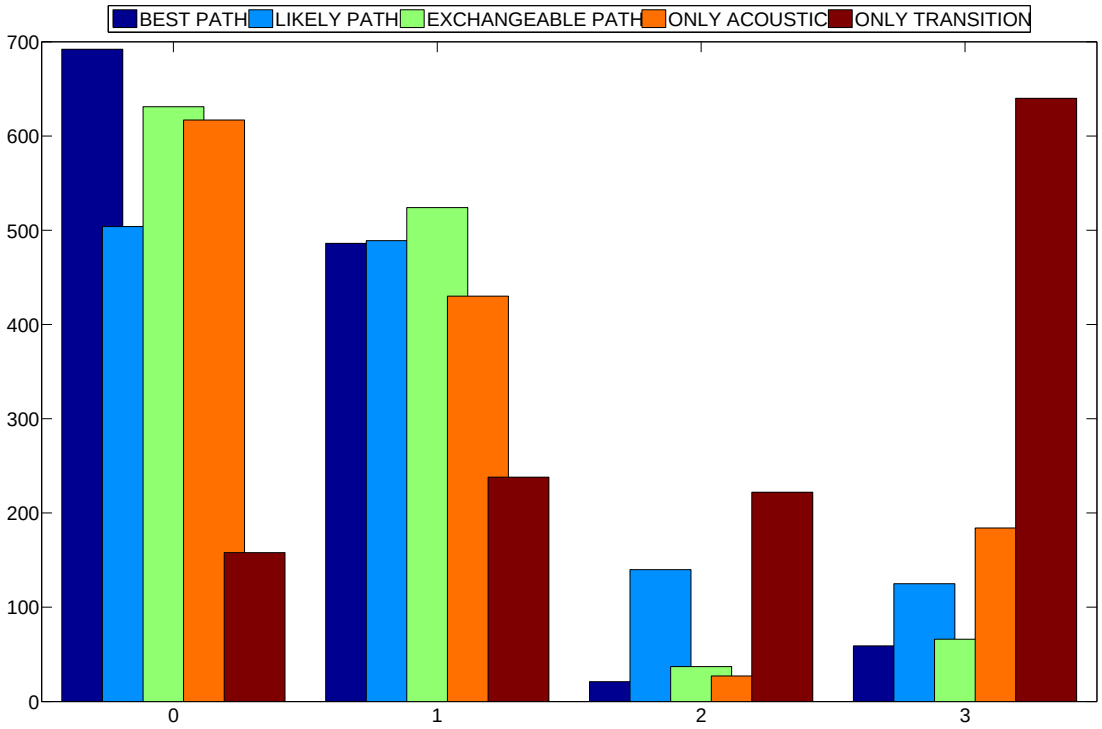


Figure 6.6: The number of figures that fall into each assessment level for the proposed five different synthesis scenarios.

sessment levels. On the other hand, using the mapping obtained by our framework increases this ratio to ∼95% for both the *best path* and the *exchangeable path* synthesis scenarios. The percentage drops to ∼90% for the *likely path* synthesis scenario, yet it is still a high percentage of the entire dance sequence. This decrease is due to the fact that the second scenario considers only the acoustic scores of the measure models rather than also using the exchangeable figures model as a constraint on the choice of figures to replace one another.

6.4.2 Subjective Evaluation Results

Subjective A/B comparisons are performed using the music-driven dance performance animations to measure opinions on the coherence of the synthesized dance choreographies with the accompanying music segment. The subjects are asked to indicate their preference of the music-driven synthesized dance animation segments for an A/B test pair on a scale of (-2; -1; 0; 1; 2), where the scale corresponds to *strongly prefer A*, *prefer A*, *no preference*, *prefer*

Table 6.3: Distribution of A/B test pairs to the original and the synthesized choreographies

	Original	Best Path	Likely Path	Exchangeable Path	Only Transition
Original	1	3	3	3	3
Best Path		1	3	3	3
Likely Path			1	3	3
Exchangeable Path				1	3
Only Transition					1

B, and *strongly prefer B*. We manually extracted 35 short segments from the audiovisual database, where each segment is approximately 15 seconds. The distribution of these 35 dance performance animation segments to the original and the synthesized choreographies is given in Table 6.3.

The subjective tests are performed over 17 subjects. The average preference scores for all comparison sets are presented in Table 6.4. The subjective A/B comparisons suggest that it is hard to differentiate between the proposed synthesis scenarios, i.e., the subjects have not strongly preferred one scenario over the others. For instance, subjects have preferred the *likely path* choreography over the *original* choreography, and the *original* choreography over the *exchangeable path* choreography. However, subjects show no preference between *likely path* and *exchangeable path* choreographies. The reason for this situation may be the selected set of short test video segments. It is also possible to state that all the proposed synthesis scenarios yield somewhat coherent choreographies that are appealing to the subjects.

The animation demo videos of the synthesized dance choreographies are available online at *http://mvgl.ku.edu.tr/bodymotionanalysis/pami10/*. These demo videos are selected to demonstrate comparatively the outputs of different choreography synthesis scenarios over several excerpts taken from the database.

Table 6.4: The subjective A/B pair comparison test results

	Original	Best Path	Likely Path	Exchangeable Path	Only Transition
Original	-0.3	-0.1	0.7	-0.6	
Best Path		0.5	0.1	0.2	
Likely Path			0.4	0.0	
Exchangeable Path				0.1	
Only Transition					

6.5 Summary

In this chapter, we describe a novel framework for automatic creation of alternative music-driven dance choreography synthesis and animation. For this purpose, we construct a many-to-many mapping from musical measures to dance figures based on correlations between dance figures and musical measures as well as correlations between successive dance figures, in terms of figure-to-figure transition probabilities. We, then, use this mapping to synthesize a music-driven sequence of dance figure labels via a constraint based dynamic programming procedure. With the help of exchangeable figures notion, the proposed framework is able to yield a variety of different dance figure sequences. These output sequences of dance figures can be considered as alternative dance choreographies that is in synchrony with the driving audio signal. To evaluate the synthesis results, we also devised an objective assessment scheme that measures the "goodness" of a synthesized dance choreography with respect to the reference choreography. To sum up, the proposed dance choreography analysis-synthesis framework has the following contributions: (i) modeling of many-to-many mappings from music to dance; (ii) automatic synthesis of alternative plausible dance choreographies via exchangeable figures model; (iii) realistic dance animations that respect intra-figure variations; (iv) a system that values both the correlations between music and dance as well as the correlations between consecutive dance figures; (v) an objective evaluation scheme for an artistic content.

Chapter 7

CONCLUSIONS

In this thesis, we proposed a novel framework for modeling, analysis, annotation and synthesis of multimodal dance performances. Specifically, we focused on finding mappings, which are in general many-to-many, between musical audio patterns and dance figure patterns for music-driven dance motion animation. A number of applications of these frameworks are also mentioned briefly at the end of this chapter. Our main contributions in this work are: (i) modeling of many-to-many mappings from music to dance; (ii) automatic synthesis of alternative plausible dance choreographies; (iii) realistic dance animations that respect intra-figure variations; (iv) a system that values both the correlations between music and dance as well as the correlations between consecutive dance figures; and (v) an objective evaluation scheme for an artistic content.

The evolution of the overall framework points out to a very important fact: amount and content of the audiovisual database is critical in developing a multimodal analysis-synthesis framework, and determines the quality and performance of the overall system. We have experienced this fact in several different parts of this thesis. We can discuss, one more time, the steps we have taken in Chapters 4, 5 and 6 to see the reason behind this argument. We initially had a limited database in Chapter 4 that allowed us to do only a primitive analysis task. We enlarged our audiovisual database a little bit, but in a constrained manner (where each dance performance included several repetitions of one dance figure). This way, in Chapter 5, we were able to perform both analysis and synthesis tasks which meant taking all the steps from the input to the desired output of the proposed multimodal framework. However, since the database was too constrained, it was not possible to generalize the results to more complex and realistic situations. For that reason, we tried to focus on a more complex audiovisual database in Chapter 6 and obtained satisfactory results with a complete multimodal analysis-synthesis framework in the end. Even the last database we used was not enough to model higher-order n-gram statistics instead of bigram probabilities.

The proposed framework currently requires expert input and tedious labeling to start with the audiovisual feature extraction and modeling tasks. This labor intensive preprocessing can be eliminated by introducing automatic measure/dance figure segmentation capability into the framework. However, such automatic segmentation techniques are not yet currently available in the literature and seems to remain as open research areas in the near future. Using HMMs to model the dance motion trajectories for representing the variations among different realizations of a dance figure was enough for this particular study because we mainly concentrated on synthesis of personalized dance performances. However, one can consider other methods such as "style machines" that will also represent the stylistic variations associated with a dance figure. Even though we tested our framework only on *kasik* folk dance database, we strongly believe that the proposed framework can be successfully applied to other types of dance performances. We also believe that the proposed framework can be easily modified to apply for other multimodal applications such as speech-driven facial expression synthesis, etc.

In conclusion, the experimental results and demonstrations show that the proposed framework is successful at creating plausible alternative dance choreographies and can be used in several other application areas some of which is mentioned in the sequel. **Dance Evaluation:** Synthesis of 3D dancing avatars for visual evaluation of synthesized choreographies using the reference models learned from a professional performer. **Dance Tutor:** A tool that automatically evaluates recorded dance performances of dance students using a library of pre-built dance models and dance performance analyses. **Cultural Heritage:** Folk dances are unfortunately becoming extinct as population ages in some nations. Learning the models of folk dances will help preservation of such cultural values by passing them from generations to generations. **Entertainment:** Automatic synthesis of dance performances from audio only for on-line games such as 'Second Life' and '3D Life' and screen savers or visualization effects for media applications on mobile devices such as iPhone, iPod, and laptops.

The future research in the context of dance performance analysis-synthesis can be extended to handle more realistic choreography designs that also considers spatial formations, plastic aspect of the dance motions and progression in space by making the necessary additions and modifications to the final framework explained in Chapter 6.

BIBLIOGRAPHY

[1] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O'Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. L. Nicolelis, "Learning to control a brain-machine interface for reaching and grasping by primates," *PLoS Biol*, vol. 1, no. 2, pp. e42, 10 2003.

[2] T. Chen, "Audiovisual speech processing," *IEEE Signal Processing Magazine*, vol. 18, no. 1, pp. 9–21, 2001.

[3] C. Bregler, M. Covell, and M. Slaney, "Video rewrite: driving visual speech with audio," in *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1997, pp. 353–360, ACM Press/Addison-Wesley Publishing Co.

[4] M. Brand, "Voice puppetry," in *SIGGRAPH'99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1999, pp. 21–28, ACM Press/Addison-Wesley Publishing Co.

[5] G. Caridakis, G. Castellano, L. Kessous, A. Raouzaiou, L. Malatesta, S. Asteriadis, and K. Karpouzis, *Multimodal emotion recognition from expressive faces, body gestures and speech*, Springer, 2007.

[6] K. Ohnishi, T. Shimono, and K. Natori, "Haptics for medical applications," *Artificial Life and Robotics*, vol. 13, pp. 383–389, 2009.

[7] L. Kim and S. H. Park, *A Haptic Sculpting Technique Based on Volumetric Representation*, vol. 3179/2004, Springer Berlin / Heidelberg, 2004.

[8] A. Ross and A. K. Jain, "Multimodal biometrics: An overview," in *Proc. of 12th European Signal Processing Conference, EUSIPCO*, September 2004, pp. 1221–1224.

[9] A. Camurri, B. Mazzarino, M. Ricchetti, R. Timmers, and Volpe G, *Multimodal Analysis of Expressive Gesture in Music and Dance Performances*, vol. 2915/2004, Heidelberg: Springer Berlin, 2004.

[10] M. E. Sargin, E. Erzin, Y. Yemez, A. M. Tekalp, A. T. Erdem, C. Erdem, and M. Ozkan, "Prosody-driven head-gesture animation," *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007*, vol. 2, pp. 677–680, 2007.

[11] William C. Reynolds, "Foundations for the analysis of the structure and form of folk dance: A syllabus," *Yearbook of the International Folk Music Council*, vol. 6, pp. 115–135, 1974.

[12] Christoph Bregler, Stephen M. Omohundro, Michele Covell, Malcolm Slaney, Subutai Ahmad, David A. Forsyth, and Jerome A. Feldman, "Probabilistic models of verbal and body gestures," in *Computer Vision in Man-Machine Interfaces*. 1998, pp. 267–290, Cambridge University Press.

[13] Okan Arikan and D. A. Forsyth, "Interactive motion generation from examples," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 483–490, 2002.

[14] Lucas Kovar, Michael Gleicher, and Frédéric Pighin, "Motion graphs," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 473–482, 2002.

[15] Matthew Brand and Aaron Hertzmann, "Style machines," in *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2000, pp. 183–192, ACM Press/Addison-Wesley Publishing Co.

[16] Jianyuan Min, Huajun Liu, and Jinxiang Chai, "Synthesis and editing of personalized stylistic human motion," in *I3D '10: Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, New York, NY, USA, 2010, pp. 39–46, ACM.

[17] S. Gao and C.-H. Lee, "An adaptive learning approach to music tempo and beat analysis," *Acoustics, Speech, and Signal Processing. Proc. IEEE Int. Conf. on*, vol. 4, pp. 237–240, 2004.

[18] Daniel P. W. Ellis, "Beat tracking by dynamic programming," *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.

[19] M. F. McKinney, D. Moelants, M. E. P. Davies, and A. Klapuri, "Evaluation of audio beat tracking and music tempo extraction algorithms," *Journal of New Music Research*, vol. 36, no. 1, pp. 1–16, 2007.

[20] A.P. Klapuri, A.J. Eronen, and J.T. Astola, "Analysis of the meter of acoustic musical signals," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 1, pp. 342 – 355, jan. 2006.

[21] M. Gainza, "Automatic musical meter detection," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, 19-24 2009, pp. 329 –332.

[22] Takuya Fujishima, "Realtime chord recognition of musical sound: A system using common lisp music," in *Proceedings of the International Computer Music Conference*, 1999, pp. 464–467.

[23] Kyogu Lee and Malcolm Slaney, "Automatic chord recognition from audio using a supervised hmm trained with audio-from-symbolic data," in *AMCMM '06: Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, New York, NY, USA, 2006, pp. 11–20, ACM.

[24] D.P.W. Ellis and G.E. Poliner, "Identifying 'cover songs' with chroma features and dynamic programming beat tracking," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, 15-20 2007, vol. 4, pp. IV–1429 –IV–1432.

[25] S. Kim, P.G. Georgiou, and S. Narayanan, "A robust harmony structure modeling scheme for classical music opus identification," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, 19-24 2009, pp. 1961 –1964.

[26] A. Hutchinson, *Labanotation: The System of Analyzing and Recording Movement*, Theatre Arts Books, 1977.

[27] Y. Li, T. Wang, and H.-Y. Shum, "Motion texture: a two-level statistical model for character motion synthesis," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 465–472, 2002.

[28] A.P. Ruiz and B. Vachon, "Three learning systems in the reconnaissance of basic movements in contemporary dance," *World Automation Congress, 2002. Proceedings of the 5th Biannual*, vol. 13, pp. 189–194, 2002.

[29] Marc Cardle, Loic Barthe, Stephen Brooks, and Peter Robinson, "Music-driven motion editing:local motion transformations guided by music analysis," *Eurographics UK Conference, Annual*, vol. 0, pp. 38–44, 2002.

[30] Hyun chul Lee and In kwon Lee, "Automatic synchronization of background music and motion in computer animation," in *Computer Graphics Forum*, 2005, vol. 24, pp. 353–361.

[31] Tae-hoon Kim, Sang Il Park, and Sung Yong Shin, "Rhythmic-motion synthesis based on motion-beat analysis," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 392–401, 2003.

[32] Gazihan Alankus, A. Alphan Bayazit, and O. Burchan Bayazit, "Automated motion synthesis for dancing characters," *Comput. Animat. Virtual Worlds*, vol. 16, no. 3-4, pp. 259–271, 2005.

[33] Takaaki Shiratori, Atsushi Nakazawa, and Katsushi Ikeuchi, "Dancing-to-music character animation," *COMPUTER GRAPHICS FORUM*, vol. 25, no. 3, pp. 449–458, 2006.

[34] Danielle Sauer and Yee-Hong Yang, "Music-driven character animation," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 5, no. 4, pp. 1–16, 2009.

[35] Jae Woo Kim, Hesham Fouad, John L. Sibert, and James K. Hahn, "Perceptually motivated automatic dance motion generation for music," *Comput. Animat. Virtual Worlds*, vol. 20, no. 2‐3, pp. 375–384, 2009.

[36] and Huang T.S. Naphade, M.R., "Discovering recurrent events in video using unsupervised methods," in *Proc. of the Int. Conf. on Image Processing 2002 (ICIP 2002)*, 2002, number II, pp. 13–16.

[37] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," *Computer Vision and Image Understanding: CVIU*, vol. 73, no. 3, pp. 428–440, 1999.

[38] D. M. Gavrila, "The visual analysis of human movement: A survey," *Computer Vision and Image Understanding: CVIU*, vol. 73, no. 1, pp. 82–98, 1999.

[39] S. Yonemoto, A. Matsumoto, D. Arita, and R.-I. Taniguchi, "A real-time motion capture system with multiple camera fusion," in *Proc. IEEE Int. Conf. on Image Analysis and Processing: ICIAP*, 1999, pp. 600–605.

[40] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, 1997.

[41] Y. Ricquebourg and P. Bouthemy, "Real-time tracking of moving persons by exploiting spatio-temporal image slices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 797–808, 2000.

[42] Y. Ehara, H. Fujimoto, S. Miyazaki, S. Tanaka, and S. Yamamoto, "Comparison of the performance of 3d camera systems," *Gait and Posture*, vol. 3, no. 3, pp. 166–169, Sep. 1995.

[43] Y. Ehara, H. Fujimoto, S. Miyazaki, M. Mochimaru, S. Tanaka, and S. Yamamoto, "Comparison of the performance of 3d camera systems II," *Gait and Posture*, vol. 5, no. 3, pp. 251–255, Jun. 1997.

[44] D. Comaniciu and V. Ramesh, "Mean shift and optimal prediction for efficient object tracking," in *Proc. IEEE Int. Conf. on Image Processing*, 2000, vol. 3, pp. 70–73.

[45] Autodesk, "Autodesk motionbuilder," Website, `www.autodesk.com/motionbuilder`.

[46] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357 – 366, aug 1980.

[47] R. Shepard, "Circularity in judgements of relative pitch," *Journal of the Acoustic Society of America*, vol. 36, no. 12, 1964.

[48] Alonso M., B. David, and Richard G., "Tempo and beat estimation of music signals," in *Proceedings of ISMIR 2004, Barcelona, Spain*, 2004.

[49] M. F. McKinney, D. Moelants, M. E. P. Davies, and A. Klapuri, "Evaluation of audio beat tracking and music tempo extraction algorithms," *Journal of New Music Research*, vol. 36, no. 1, pp. 1–16, 2007.

[50] F. Ofli, Y. Demir, E. Erzin, Y. Yemez, and A. M. Tekalp, "Multicamera audio-visual analysis of dance figures," *IEEE International Conference on Multimedia and Expo, 2007. ICME 2007*, pp. 1703–1706, 2-5 July 2007.

[51] F. Ofli, Y. Demir, E. Erzin, Y. Yemez, and A. M. Tekalp, "Multicamera audio-visual analysis of dance figures using segmented body model," .

[52] F. Ofli, C. Canton-Ferrer, J. Tilmanne, Y. Demir, E. Bozkurt, Y. Yemez, E. Erzin, and A.M. Tekalp, "Audio-driven human body motion analysis and synthesis," *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 2233–2236, 31 2008-April 4 2008.

[53] F. Ofli, Y. Demir, E. Erzin, Y. Yemez, A. M. Tekalp, K Balci, I. Kiziloglu, L. Akarun, C. Canton-Ferrer, Tilmanne J., E. Bozkurt, and A.T. Erdem, "An audio-driven dancing avatar," *Journal on Multimodal User Interfaces*, vol. 2, no. 2, pp. 93–103, 01 Sep. 2008.

[54] U. Bagci and E. Erzin, "Automatic classification of musical genres using inter-genre similarity," *IEEE Signal Processing Letters*, vol. 14, no. 8, pp. 521–524, August 2007.

[55] S. Young, "The htk hidden markov model toolkit: Design and philosophy," in *Technical Report TR.153, Speech Group, Department of Engineering, Cambridge University (UK)*, 1993.

[56] F. Ofli, E. Erzin, Y. Yemez, and A. M. Tekalp, "Multi-modal analysis of dance performances for music-driven choreography synthesis," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 14-19 2010, pp. 2466 –2469.

[57] M. E. P. Davies and M. D. Plumbley, "Context-dependent beat tracking of musical audio," *Audio, Speech, and Language Processing, IEEE Trans. on*, vol. 15, no. 3, pp. 1009–1020, 2007.

[58] S. J. Young, N. H. Russel, and J. H. S. Thornton, "Token passing: a simple conceptual model for connected speech recognition systems," *Technical Report: CUED/F-INFENG/TR.38 Cambridge University Engineering Department*, July 1989.

[59] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 26, no. 1, pp. 43 – 49, feb 1978.

[60] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems 14*. 2001, pp. 849–856, MIT Press.

[61] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*. 1967, pp. 281–297, University of California Press.

[62] Peter Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, no. 1, pp. 53–65, 1987.