COMBINING AUDIO AND VIDEO TEMPO ANALYSIS FOR DANCE DETECTION

by

RYAN MATTHEW FAIRCLOTH
B.S. University of Central Florida, 2005

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Electrical and Computer Engineering
in the School of Electrical Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2008

Major Professor: Mubarak Shah

ABSTRACT

The amount of multimedia in existence has become so extensive that the organization of this data cannot be performed manually. Systems designed to maintain such quantity need superior methods of understanding the information contained in the data. Aspects of Computer Vision deal with such problems for the understanding of image and video content. Additionally large ontologies such as LSCOM [1] are collections of feasible high-level concepts that are of interest to identify within multimedia content. While ontologies often include the activity of dance it has had virtually no coverage in Computer Vision literature in terms of actual detection. We will demonstrate the fact that training based approaches are challenged by dance because the activity is defined by an unlimited set of movements and therefore unreasonable amounts of training data would be required to recognize even a small portion of the immense possibilities for dance. In this thesis we present a non-training, tempo based approach to dance detection which yields very good results when compared to another method [2] with state-of-the-art performance for other common activities; the testing dataset contains videos acquired mostly through YouTube. The algorithm is based on one dimensional analysis in which we perform visual beat detection through the computation of optical flow. Next we obtain a set of tempo hypotheses and the final stage of our method tracks visual beats through a video sequence in order to determine the most likely tempo for the object motion. In this thesis we will not only demonstrate the utility for visual beats in visual tempo detection but we will demonstrate their existence in most of the common activities considered by state-of-the-art methods.

I am dedicating this work to my family and friends who have been a tremendous support for me throughout my life and especially during the last two years.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Mubarak Shah for his continued support over the last 2 years for his constant encouragement as I developed as a researcher. In addition I would like to thank my brother, Ronnie Faircloth, for his help in understanding the various aspects of music and tempo that I needed for my research. I would also like to thank my colleague and friend Phillip Berkowitz for being such a good sounding board for ideas and for offering his assistance whenever I needed it. Finally I would like thank my committee and the countless others who listened to my ideas and gave me feedback.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AVI	- Audio Visual Interleave
LSCOM	- Large Scale Concept Ontology for Multimedia
MATLAB	- Matrix Laboratory
PIV	- Particle Image Velocimetry

CHAPTER ONE: INTRODUCTION

Computer based information retrieval has become one of the most highly researched areas as the abundance of computer based multimedia is growing rapidly. Encompassed under this conglomeration is the need for an understanding of the various aspects of the data and features within the multimedia. Within the field of Computer Vision many organizations have formed large ontologies of high-level concepts, such as LSCOM [1], that have become the goal of researchers to uncover within image and video content. Over time many low-level features have been derived that often times are used collectively to undercover the various high-level concepts. This thesis is focused on uncovering and defining one such low-level feature, visual tempo, as it has never before been used in Computer Vision literature.

Before we get too far it is important that a distinction be made on exactly what we refer to as visual tempo. This is necessary as significant work has been done in the area of tempo as it relates to the progression of film as in [3]. Tempo in this regard refers to a quantitative value that captures various aspects of a movie to describe the pace of the movie for a particular shot. Our goal in this thesis is to understand tempo with respect to the motion of objects within a video clip analogous to tempo in music. While indeed the motion of objects is considered as one of the various aspects involved in film tempo, only the motion magnitude is generally considered as a holistic value within a shot incorporating both camera and object motion. Our definition of visual tempo is best understood through the activity of dance and how the visual motion relates to the music through what we have termed visual beats. In the next section we will present our exact definition for tempo and we will describe the relationship between audio and visual tempos as they will be used in our work.

What is Tempo?

Most definitions of tempo make reference to music and for good reason because tempo is a concept that is most easily understood through music. The majority of definitions within this context refer to tempo as the speed or rate at which the notes in a piece of music are played but a few more general definitions will refer to tempo as a rate of movement. In either definition, tempo is a concept that we as humans can easily sense and we often find ourselves caught in the tempo of music without even realizing it. Whether at a concert, dance recital, or listening to music in the car people always tend to tap, clap, sway, or move in some way to the music and it is all outward signs of our instinctive understanding of tempo and musical beats.

Throughout this thesis we will refer to tempo as understood through two sensory inputs, auditory and visual. From the perspective of an auditory sensor we will refer to tempo as such, an audio tempo, and it will be defined as the rate in which pulses or strong intensity input is delivered to the sensor. It is especially important for us in this thesis to provide a meaningful definition for tempo as observed by a visual sensor. We will refer to this as the visual tempo and will define it as the rate at which visual motion transitions, a so called “rate of movement.”

As mentioned above, humans can best relate to tempo through the example of clapping along to music. When clapping, the point at which the hands come together is a point of strong emphasis in the music called a beat. Tempo in fact is defined numerically in terms of beats per minute and musicians will often use a device called a metronome, which can be set to beep at various speeds of beats per minute, to help them “keep time” correctly, as it is usually referred to. When relating the audio and visual tempos of a video there is an important observation to understand; using the example of clapping if we were to only consider the points at which the hands make contact as visual beats we would only capture half of the actual tempo. Figure 1-1 provides an illustration of this using a synthetic piece of music. From this example it is important to notice that the actual tempo is with respect to beats as defined by the time signature, 4/4, given in the figure. This simple example shows a single measure with four quarter notes where each quarter note represents a single beat and therefore creating a total of four beats.

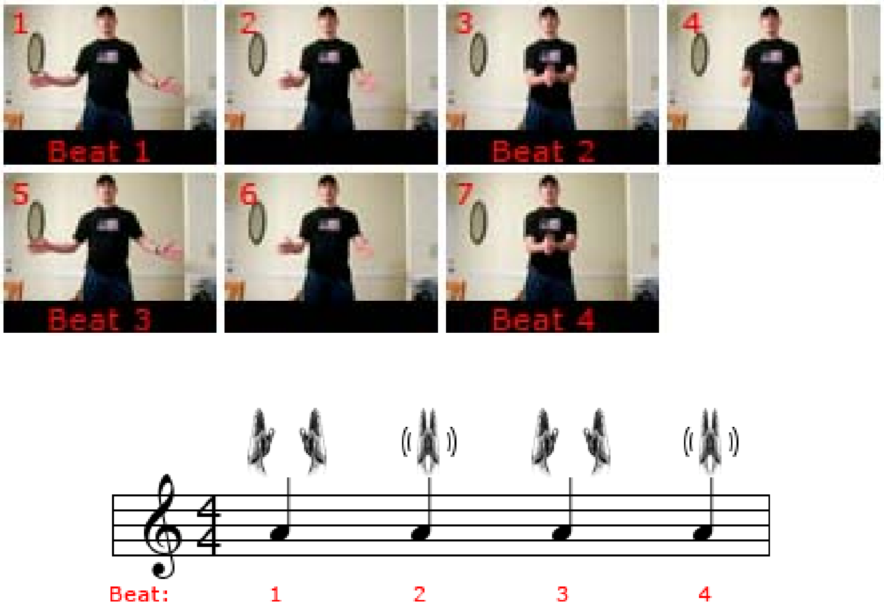


Figure 1-1: Illustration of beats using a synthetic example of a musical piece.

While in our example the actual clap takes place on notes two and four it is easily observed that the act of clapping has two transitions in motion and therefore all four visual beats would be expressed which are also illustrated in Figure 1-1. From this perspective it becomes apparent how the full tempo would be realized visually through the act of clapping and this same principal can be extrapolated to other forms of visual tempo expression as well such as tapping a foot, bobbing the head, or even dancing.

What is the Importance of Tempo?

Tempo can be recognized as a feature in many of the activities that we perform every day many of which we may not even realize such as brushing teeth or typing. In various workout and sporting activities there can also be an underlying tempo involved. In past research periodicity measures have been proposed which can find a tempo like quantity in many of the cases that we have just mentioned but we consider the concept of periodicity in this regard as merely a special case of tempo; we will discuss more on this in a later chapter. We wish to first discuss the major influences for this thesis on visual tempo analysis.

Audio Research

Automatic extraction of tempo from musical audio is a problem that has been the focus of music research for over 20 years. Major successes have been achieved within the last 10 years and in the last 3-4 years a community by the name Music Information Retrieval Evaluation eXchange (MIREX) has organized a yearly evaluation for state-of-the-art methods to accomplish a variety of tasks such as “Audio Tempo Extraction” and “Audio Beat Tracking.” This evaluation has led to the development and the widespread growth for highly advanced techniques in beat onset detection and tempo extraction in audio. Beat detection is not an easy task and detection techniques often return large numbers of candidate beats that need to be further analyzed in order to distinguish a tempo; both of these concepts will be discussed further from an audio standpoint in the next chapter on related work. In chapter three we will address both concepts from a visual perspective as we develop a successful visual tempo extraction algorithm.

Activity Recognition

Activity recognition is a largely unsolved problem in the field of Computer Vision. Even still many of the ontologies that were mentioned earlier include large sections of activities and significant work has been done in the exploration of a solution to the problem by various researchers [4], [5], [6], [2], [7]. In most cases a set of activities are considered and a general solution is sought for the set. Among the activities that are commonly used are walking, jogging, running, sitting, jumping, hand-waving, and other various exercise activities such as jumping jacks. Many of these actions have a well defined and well understood structure that can be exploited with standard pattern recognition approaches and/or the use of cyclic motion detection as in [8]. More on these techniques and others will be discussed in the related work.

Later in this thesis we will focus on the understanding of the activity of dance as an application for visual tempo. From our inquisitions of the relevant literature it is apparent that no significant work has been done for this activity. Perhaps the biggest reason for this void is due to the difficulty of the problem when considered from classical approaches. The main difficulty to recognizing dancing in videos for previous approaches is that there is no definition or model for what motions signify dancing. The most important aspect of the activity is in the relationship between the motions in the video and the content in the audio; the key element that links these two components is the tempo. We will show that with the knowledge of tempo in both the audio and visual content we can utilize our non-training based method to solve an unconstrained problem that is difficult for state-of-the-art activity recognition methods; before we discuss the details of our approach, however, we will briefly discuss some of the important features of our work.

The importance of this thesis in comparison to other research in activity recognition is in the unconstrained nature of the problem; later we will demonstrate this aspect with consumer video acquired through YouTube. In addition, only introductory work [9] has been introduced in relating visual tempo from video content with that of an audio tempo. We will show in this thesis that visual tempo analysis, as we define it for video content, can be achieved with highly accurate results and that it can further be used for work in the area of dance recognition.

In Chapter Two we will present previous work relevant to this thesis and in Chapter Three we will present our novel algorithm for computing visual tempo. Chapter Four provides our procedure for evaluating videos for dance detection including the tools that we have used. In Chapter Five we will present our experimental results and finally, in Chapter Six we will conclude our findings and discuss avenues for future work.

CHAPTER TWO: RELATED WORK

In this chapter we will discuss previous work that has been done across several disciplines of research. The methods and topics we cover are provided as a background for the better understanding of the current state-of-the-art for work pertaining to this thesis. It is our hope that this knowledge will allow for the most thorough realization of the contributions made by this thesis.

Tempo Analysis

It is of no surprise that considerable effort has been applied to the extraction of accurate tempo measurements for audio content [10]. As was discussed in the introduction chapter, we as humans can realize tempo through various mediums with ease and it is often the goal of research to seek ways to replicate the human solution to a problem. We mention this aspect because we believe that tempo is an important low-level feature that may aide in solutions to a wide range of problems. In the following subsections we will discuss how tempo analysis is performed and utilized in the fields of Music Research and in Computer Vision.

Audio Tempo Analysis and BeatRoot

BeatRoot is a software package that was developed by Simon Dixon and has most recently won the MIREX 2006 Audio Beat Tracking Competition. The details on how the software works can be found in [10], [11]. It is of importance to mention this work because the method used by BeatRoot for tempo analysis is intuitive and effective. In both of the papers cited Dixon provides a thorough history on the development of audio tempo analysis over the last 30 years and we therefore refer the reader to those sources for more information. We wish to outline here the method of tempo computation proposed by Dixon. The approach used in the BeatRoot software is to first discover the onset of potential beats in the audio. The actual methods to perform this operation are generally centered on the computation of the Short-Time Fourier Transform. Dixon discusses many of these techniques in [11] including the award winning method utilized by BeatRoot. The stage for beat onset detection typically generates a large number of false beats due mainly in part to harmonics present in the audio but there are other sources. For this reason it is necessary to employ an algorithm that is capable of deciding which set of beats provide the tempo for the audio. Therefore, the second stage evaluates what is known as the inter-onset interval or the time between beats. Dixon proposes a method called inter-onset interval clustering to evaluate all possible intervals between beats and to generate a list of the most likely interval widths called the tempo hypotheses. In the final stage of analysis all potential beats are tracked with respect to the tempo hypotheses from step two and the most likely series of beats is selected through a non-probabilistic method of scoring. There is a known issue with tempo tracking algorithms mentioned in [12] where the output tempo will be proportional to the correct tempo by a factor of one half or two in some cases. This is a result of the underlying musical properties for certain pieces and is important to note because the same issue arises with our visual tempo algorithm, however, results will show that the impact on our testing dataset is minimal. We will speculate on the effects of this issue in a later chapter.

Film Tempo

We briefly mentioned in the introduction chapter about previous work on visual tempo analysis in Film research and in this section we will discuss more on this with respect to tempo as a “rate of movement.” In [3] it is noted that film literature refers to an “expressive element” typically called the “tempo, pace, or subjective time.” Therefore we note that while the measure is not as we define it in our work it is validly defined as tempo. The main importance here is not to redefine tempo but rather to shed light on how tempo is currently being used from a Computer Vision standpoint.

According to [3] tempo is an important characteristic of film and often provides a foundation for other “higher-order semantic film elements.” The formulation for tempo from that work includes terms for both motion magnitude and shot length. It is unclear from the paper however as to what actually defines the motion magnitude but it is likely a characteristic of the camera motion. This agrees with what we have seen in many cases of film tempo as the different aspects of motion such as object motion and camera motion are captured in a single expression. We include here the equation used in [3] for computing tempo to provide a better understanding for comparison:

$$T(n) = \alpha \cdot W\big(s(n)\big) + \frac{\beta \cdot (m(n) - \mu_m)}{\sigma_m}$$

where n is the shot number, $s(n)$ is the shot length, $W(s)$ is a weighting function, and $m(n)$ is the motion magnitude of shot n.

One last source worth mentioning under film tempo is [9]. In this work the authors attempt to match motion tempo to musical material to form what they call “MTV-style” videos out of simple consumer video clips, or home videos. While this work provides perhaps the closest thing to our work in terms of a representation for visual tempo there are many details that are unclear. Their work on audio tempo analysis appears to match what we have seen elsewhere and used for our own work as well through BeatRoot. In order to capture visual tempo the work done in [9] proposes a method based on capturing both camera and object motion. To capture object motion their algorithm begins with face detection in order to locate subjects in the scene and these subjects are then tracked across all frames in order to determine the speed of the individual motion. It is because of the seemingly global aspect of their visual tempo that we include this as film tempo analysis. From the paper only the magnitude of motion appears to be considered as mention is made that the “activities of sitting and lying are matched with a lower music tempo while running is matched with a higher music tempo.” By our definition of visual tempo the activities of sitting and lying would not necessarily produce tempo values. Unfortunately in the paper no aggregate results are reported and only one results oriented figure is presented to demonstrate the result for a single song.

Activity Recognition

It was stated in the introduction that activity recognition is a highly active field in Computer Vision and the majority of the work is being done, often times quite successfully, under standard pattern recognition and model based techniques. In this section we will discuss several approaches that may be of interest for comparison. Some techniques provide measures of periodicity for various activities and often times provide sufficient results for limited datasets that are well constrained. Additionally we will discuss details on a state-of-the-art activity recognition method [2] that we used to compare results on a dataset for dance recognition.

Cyclic and Periodic Motion Techniques

Periodicity in image sequences is a feature that can be exploited in the simplest case through methods of cross correlation [7]. Cutler and Davis propose a method in which a correlation matrix is generated from an image sequence based on the cross correlation of every image with every other image in the sequence and then autocorrelation is performed on the resulting matrix. In Figure 2-1 we present the result of the cross correlation step on a jumping jacks sequence and from the result it should be obvious how periodicity can be discovered in the motions of this sequence.

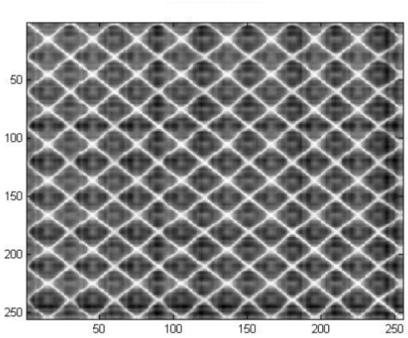


Figure 2-1: Cross correlation result for sequence of jumping jacks.

Additionally Cutler and Davis show that several activities can be identified by matching 2-D lattice structures to the result of autocorrelation. Results are presented that identify between a walking person, a running dog, and other, however, we do not believe that such a method would be practical or reliable beyond this simple example. In order to utilize this technique the moving object must be aligned throughout the sequence of images in many cases this can be done through background subtraction and registration. We found that this technique can even be applied to certain periodic dance routines as well but will only provide a measure of the periodicity of the dance and not the underlying tempo. In Figure 2-2 we provide results for this method on two sequences of the Macarena in order to justify the claim. In the figure, the spacing between successive diagonal white lines represents the number of frames in one period. In the illustration we have aligned the two matrices and connected the corresponding diagonal white lines with the parallel yellow lines shown and thus demonstrating that the periodicity of the dance remains the same across the two different performances. This fact should be intuitive as the dance moves are coordinated by the music and therefore should occur at the same rate of periodicity. Using this method we might be able to produce a simple dance classification algorithm but because the only information we have is periodicity we cannot detect dance in a video sequence as many activities are periodic.

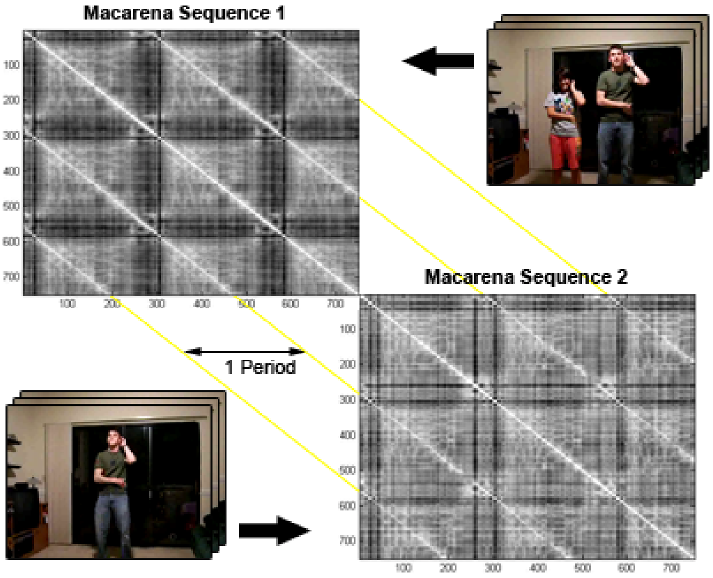


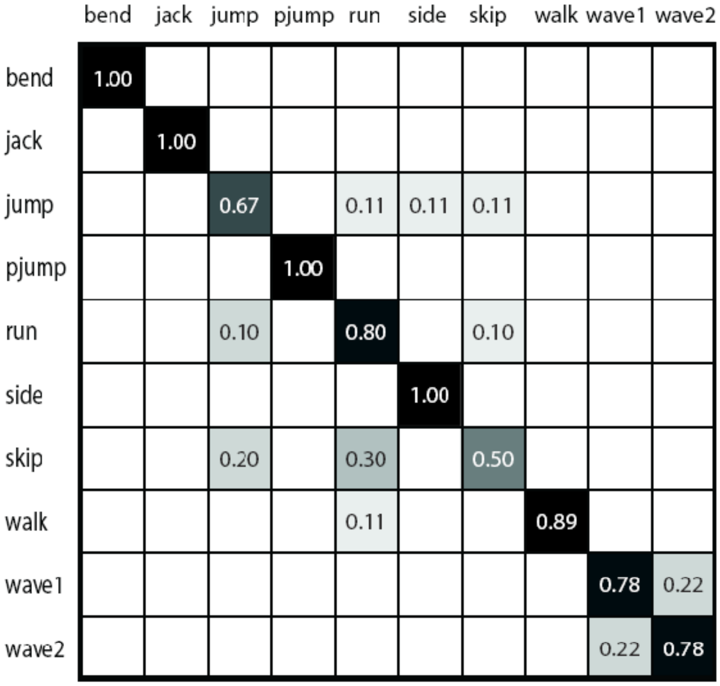
Figure 2-2: Illustrating periodicity for Macarena sequences.

In [5], Polana and Nelson define three common classes of motion, temporal textures, activities, and motion events. They propose that different motion recognition schemes are necessary for proper classification of these classes. Temporal textures according to Polana and Nelson are “motion patterns that exhibit statistical regularity but have indeterminate spatial and temporal extent.” An activity as they define it “consists of motion patterns that are temporally periodic and possess compact spatial structure.” The last class, motion events, “consists of isolated simple motions that do not exhibit any temporal or spatial repetition.” We propose that the activity of dance is a mixture of classes by these definitions as it typically contains a series of event like motions that occur contrarily with a determinate temporal extent to makeup an activity. Therefore we hold that while their method of activity recognition could as in [7] recognize the periodicity of various periodic dances it could not detect the activity of dance in a general sense.

Other Approaches

Many of the approaches currently being used for activity recognition involve 3-D spatiotemporal extension of previous 2-D spatial methods for object detection and classification. These methods extract various spatiotemporal features from training sequences and learn models with an SVM or another machine learning approach; finally applying the same routine to test data. Other more recent approaches such as [2] use 3-D interest point operators to locate words for a Bag-of-Words style approach. In [4] silhouettes of an activity are turned into volumes and features are extracted from the space-time volumes. Additionally, this method was used to experiment with ballet sequences where particular ballet moves were learned and detected. As we will discuss in the next section this type of approach is not practical for dance activity detection in a general sense.

In [2], Scovanner et al propose a 3-dimensional extension to the popular SIFT [13] feature and then employ a Bag-of-Words- approach to the features for activity recognition. The steps of the algorithm involve first computing the 3D SIFT features at random points throughout the space-time volume for each video in a dataset. Next, the feature descriptors are clustered using the K-Means algorithm and then a histogram is generated for each video based on the occurrences of each of the k-means in the volume. Typically for the testing set in these approaches the mean with the closest distance to each feature descriptor is considered a match and similarly a histogram is generated. In the final step the histograms are passed to an SVM classifier as feature vectors. Scovanner et al evaluate performance on a common activity recognition dataset [4], employing a leave-one-out strategy of testing and the results are provided in Figure 2-3. The main benefit of a leave-one-out testing strategy is to allow evaluation of the method on each data instance given the maximum amount of training data from the rest of the dataset. This same testing approach is used in the evaluation of the method proposed by Scovanner et al on the dance recognition dataset proposed by this thesis.



	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	1.00									
jack		1.00								
jump			0.67		0.11	0.11	0.11			
pjump				1.00						
run			0.10		0.80		0.10			
side						1.00				
skip			0.20		0.30		0.50			
walk					0.11			0.89		
wave1									0.78	0.22
wave2									0.22	0.78

**Figure 2-3:	3D SIFT method [2] results on the Irani action dataset [4].**

Dance Activity Recognition

Dance activity recognition has had only limited coverage within Computer Vision literature and what does exist has only presented introductory work on the topic. The approach taken by these methods is to perform event modeling for specific dance moves as in [4]. This approach is thus highly limited in the range of applications to dance activity recognition. We believe that these methods are ineffective for dance activity recognition in general because there are simply no motions which define dance as a whole. Certainly there are particular choreographed dances where specific moves can often be found by model or template matching but again the application range is extremely limited. As we will discuss later, our method utilizes the only aspect that is always present among various dances and dance styles; the underlying element is what we call the visual tempo.

Music Classification

Within music and signal processing literature a lot of work has been done in music versus non-music classification and many of the methods follow standard machine learning practices of feature extraction and then the application of a classifier such as SVM, AdaBoost, or a Bayesian classifier. Some of the various classes according to [14] into which an audio clip may belong are noise, speech, music, natural sounds, and artificial sounds. While for our work we are only interested in considering two classes, music and non-music, for which simple binary methods of classification will be sufficient most binary classification schemes, such as SVM based, require training on a negative class of features. It is therefore important that an appropriate scheme consider features of all possible input classes. There are many features that are commonly utilized in audio classification techniques including volume, zero-crossing rate (ZCR), Discrete Fourier Transform (DFT) coefficients, and Mel-Frequency Cepstral Coefficients (MFCCs). It is beyond the scope of this thesis to discuss these features and methods in detail but more can be found on these topics in various sources such as [14]. In this thesis we consider music versus everything else a solved problem and as will be mentioned later only consider video clips with music in the audio track.

Dance Music

Additional work on audio classification has been done in order to distinguish between various kinds of music. In [12] the authors deal specifically with dance music classification where the goal is to disambiguate between various styles of dance music. The interesting thing about this work is how they utilize BeatRoot in order to initially establish a set of candidate classes. In this work they were able to find tempo ranges into which different classes of dance music commonly occur. Once the candidate classes are found based on tempo analysis a final decision is made based on classifiers trained to disambiguate the classes that overlap in terms of tempo range.

Optical Flow

In this section we will briefly discuss what optical flow is and demonstrate how it is obtained for a video sequence. Optical flow is a commonly used concept in Computer Vision and there are many algorithms to choose from such as [15], [16], [17], [18], [19]. Given a pair of sequential input images optical flow describes the movement that took place between the two images through a field of motion vectors. Typically the vectors are computed over an equally spaced grid within the image dimensions. In Figure 2-4 we illustrate the process with an example input sequence of a person running. In the figure we show the input image sequence and the resulting flow field computed between every pair of images. Additionally we have provided a detailed look at one flow field portraying the motion vectors computed over an equally spaced grid of 4 pixels.

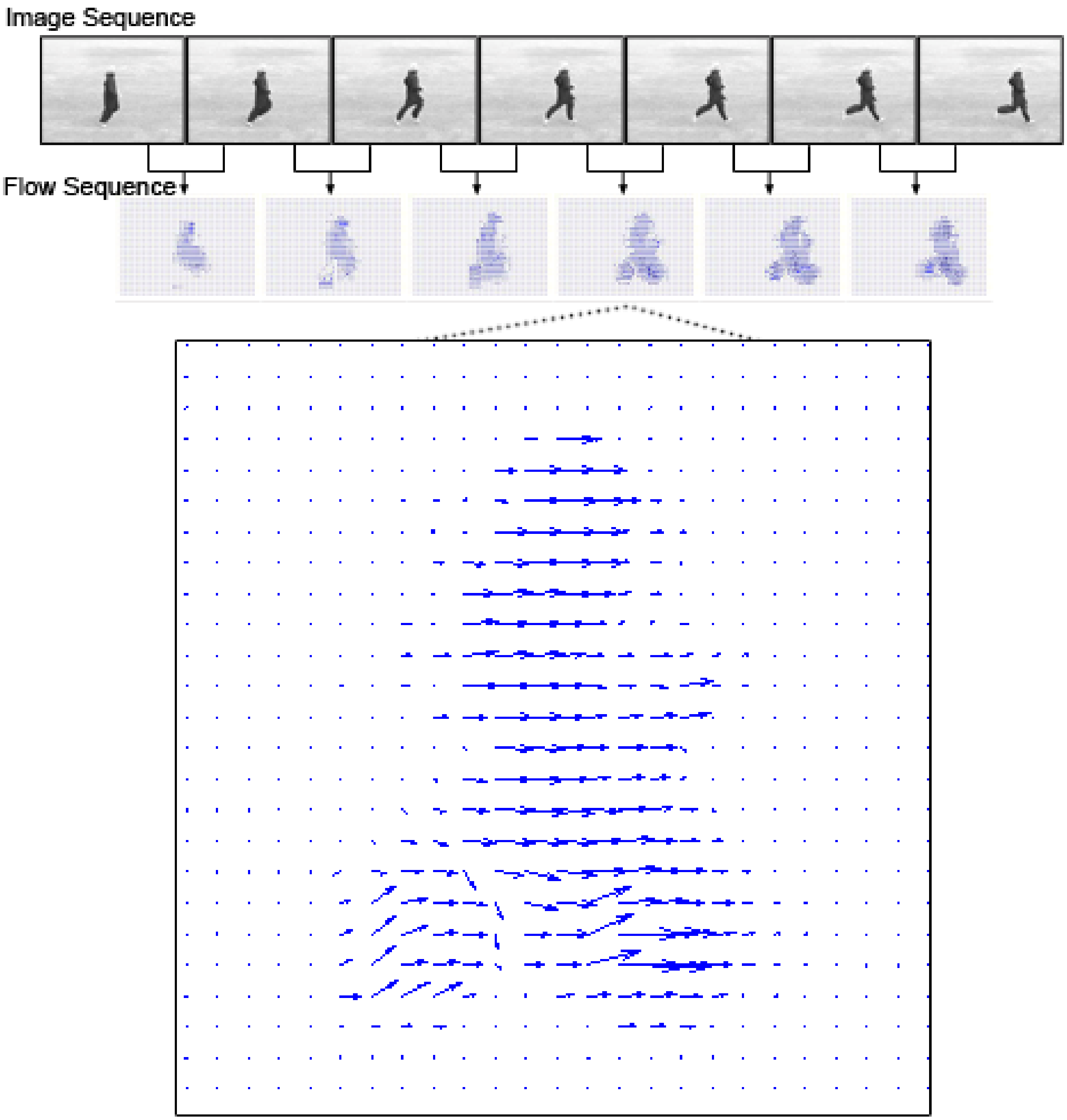


Figure 2-4: Illustrating optical flow for an input sequence.

In this chapter we have presented background and related work to this thesis. We hope that this information has provided the reader with an understanding of how our work fits in the area of activity recognition and has a better understanding of the challenges presented by the problem we aim to solve. In the next chapter we will provide our solution to visual tempo extraction and later we will discuss how we utilize it for dance detection in video sequences.

CHAPTER THREE: COMPUTING VISUAL TEMPO

The inspiration for our work in visual tempo analysis stems from the problem of dance recognition in video content. While considering various possible solutions to dance activity recognition we asked ourselves how we as humans know that there is dancing taking place in a particular situation. Upon inspection of several videos we concluded that we can make this observation very easily because we can recognize a so called visual tempo in the movements. However, visual tempo analysis has been an area of little research from the perspective of a strict metrical measurement as in music tempo. As we mentioned in the introduction chapter this is different from the work that has been done on film tempo. In this chapter we will discuss an effective way to compute visual tempo given an input video clip. A flowchart for the algorithm is given in Figure 3-1 and each component will be described in detail throughout the remainder of this chapter. From the flowchart it can be observed that we have broken the algorithm into three major components: visual beat detection, inter-onset interval clustering for tempo hypotheses and visual beat tracking.

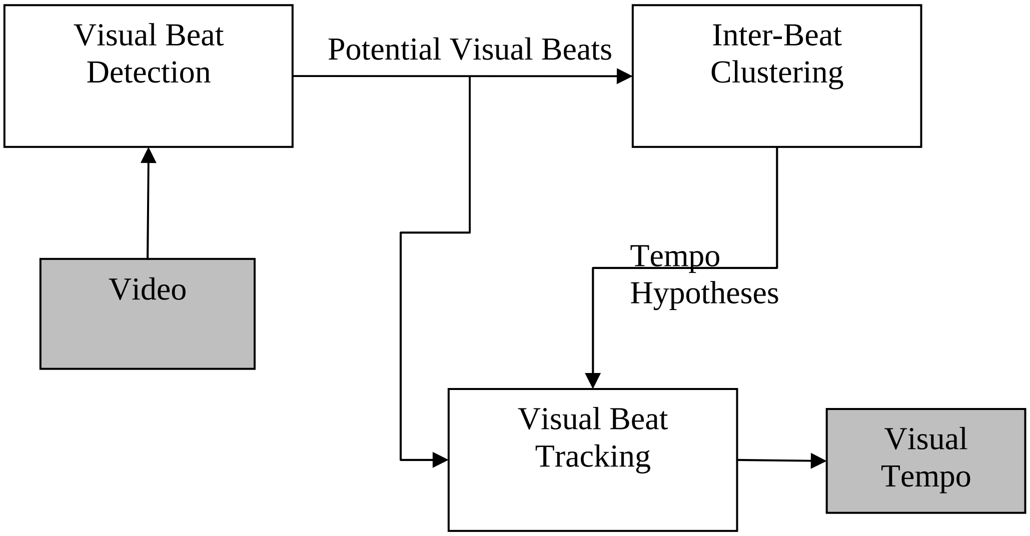


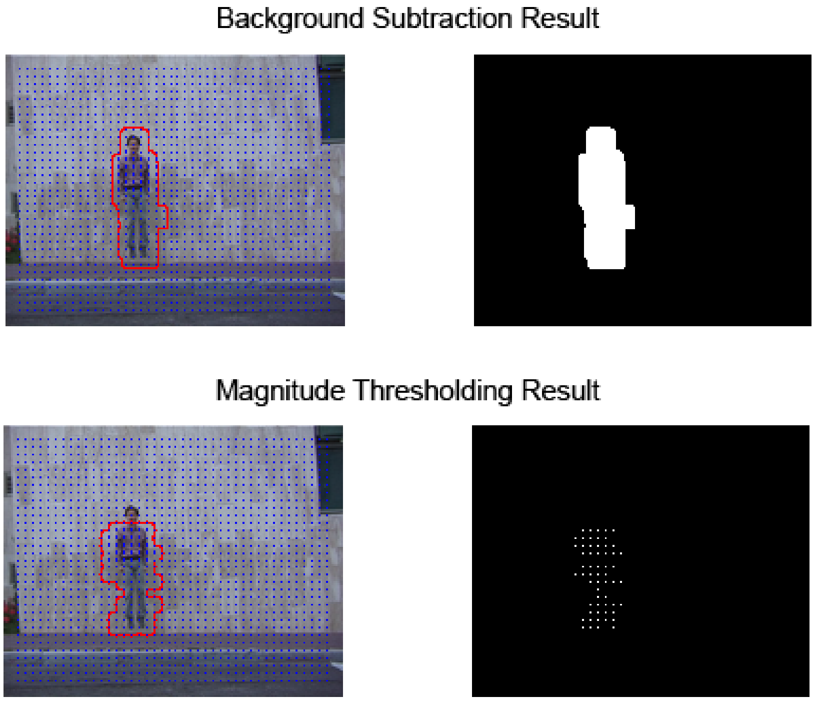
Figure 3-1: Visual tempo extraction process flowchart

Visual Beat Detection

For the purposes of this thesis we define a visual beat as any significant transition in the motion of an object in whole or part contained in an input video clip. The first stage of our approach is to detect such visual beats. In audio research various methods have been proposed over the years for performing beat onset detection in audio and many of them are outlined in [11]. As we will discuss in this chapter our approach to visual beat detection involves the analysis of a one dimensional signal obtained from the optical flow of an input video.

Computation and Isolation for the Motion of Interest

The initial stage of our approach is to compute the optical flow between successive frames of an input video and for our experiments we utilize Particle Image Velocimetry components available for MATLAB from The URAPIV Project [15]. Typically the results of optical flow can be quite noisy for various reasons and often there is extraneous motion in the areas of background clutter due to slight camera jitter, sensor noise or mismatched pixels or regions. More details will be given in the next section but it is important for our algorithm in the case of large background regions that the motion for the objects of interest be isolated. In our initial experiments using a standard activity recognition dataset [4] we were able to use binary masks which were acquired through background subtraction and provided with the dataset. This type of approach is ideal to isolate the flow for the objects of interest but it is not always feasible. We also found that when the motion of the objects of interest dominates the flow due to camera motion or other effects that a simple threshold on the flow magnitude can be effective to isolate the dominate flow of the objects. Figure 3-2 shows an example comparing the results of both isolation methods. In the next section we will discuss how to analyze the flow obtained from this stage.



Background Subtraction Result

Magnitude Thresholding Result

Figure 3-2: Showing the effectiveness of both forms of motion isolation.

Analyzing Flow and Discovering Potential Beats

As was discussed earlier, in previous “tempo like” research we have found in all cases that the feature of importance from the standpoint of object motion is the motion magnitude. Our research and experimentation tells us that for computing a tempo based on an object’s motion over time that the phase of the motion contains more information. The goal of this section is to describe our feature of interest and to demonstrate how to compute it. It will be the goal of later chapters to provide experimental results that demonstrate the usefulness from a low-level feature standpoint.

Once the flow for the objects of interest has been computed and isolated the next step is to simply average the flow for these regions independently for the horizontal and vertical components of the motion vectors. We then combine these two components into a single phase value using the following standard equation:

$$\theta = \tan^{-1}\left(\frac{v}{u}\right)$$

where u and v represent the horizontal and vertical components of the motion vector respectively. Next we apply a small adjustment to the computed θ when using a 0-360° based scale. This adjustment was introduced based on an observation of the flow generated by a person translating horizontally in the video frame and is illustrated in Figure 3-3. This adjustment is defined in the following equation and will be explained in a moment.

$$\theta_a = \begin{cases} 180 - \theta, & 90 \leq \theta < 180 \\ 540 - \theta, & 180 \leq \theta < 270 \\ \theta, & elsewhere \end{cases}$$

This process is repeated for all frames of optical flow from the video clip and the corresponding θ_a values are collected into a one dimensional signal given as $\theta(n)$ where n denotes the frame number. The example in Figure 3-3 shows the one dimensional signal described thus far for the case of a single person running across the field of view first from right to left and then from left to right. We can observe in this plot that the period of time where the runner is moving from left to right we have clearly defined peaks in $\theta(n)$ but this is not the case as the runner traverses across the frame from right to left. Unfortunately we cannot provide video clips in document form but hopefully we can mentally picture what is taking place as the runner moves across the field of view in addition to the obvious horizontal translation. We realize that the runner is also translating with a slight vertical component as well. As the runner translates horizontally along the 0° vector her vertical component causes oscillation at the extreme ends of the 0-360° scale which cause the sharp peaks in the plot. Moving horizontally along the 180° vector the vertical component causes only a slight oscillation above and below 180°. The modification to θ that has been proposed simply folds the space along the y-axis mapping the regions of quadrants II and III into quadrants I and IV respectively allowing both directions of translation to exhibit equal responses. This will effectively exaggerate the peaks when motion is moving along the 180° vector. It may seem reasonable to think this adjustment may adversely affect other phase transitions or that a similar adjustment could be made for other quadrants but in practice we find that this is sufficient and works well.

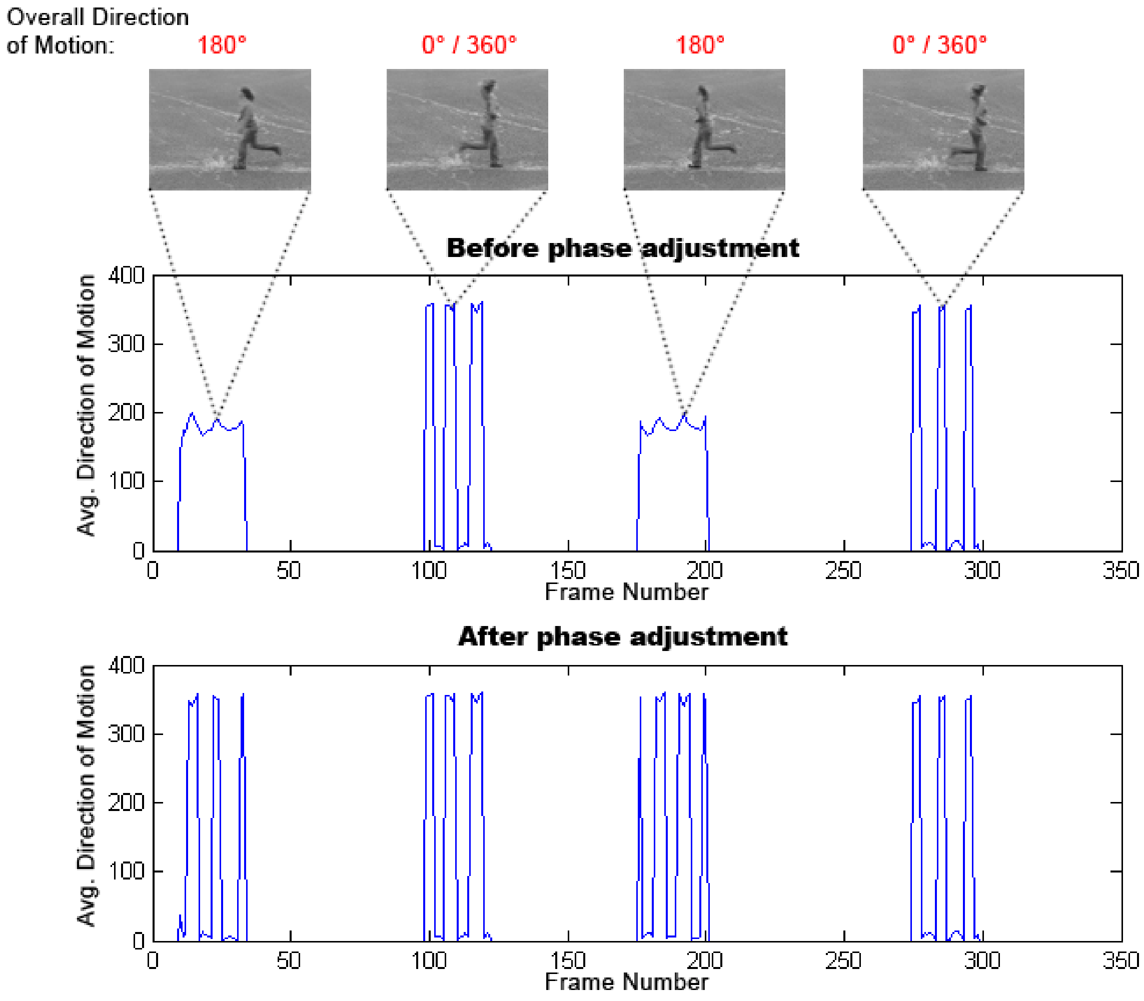


Figure 3-3: Illustrating the importance for the phase adjustment.

Next we typically divide the signal by 360º to normalize it into the interval [0,1) and follow with the application of a rectangular averaging window usually in length between 5 and 10. Generally this will result in better peaks and will additionally diminish small anomalies that may show up in the signal for various reasons. Caution does need to be taken and several values should be tried to find what works best as peaks that are too close may be merged together during averaging.

Now that we have prepared the one dimensional signal for $\theta(n)$ we need to perform peak analysis to find potential visual beats. To do this we find both the local maximums in the signal and the local minimums. Then for each maximum we apply the peakiness test shown in Figure 3-4 from [20] and given by the following equation:

$$Peakiness = \left(1 - \frac{V_a + V_b}{2P}\right) \times \left(1 - \frac{N}{W \times P}\right)$$

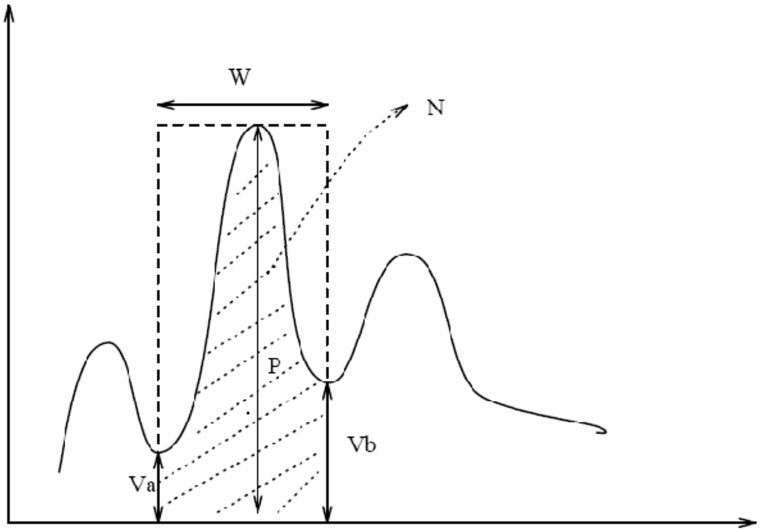


Figure 3-4: Illustration for the peakiness test from [20].

This allows us to quickly eliminate spurious peaks and to consider only peaks with higher peakiness values as potential visual beats. Additional steps can also be taken to better perform the elimination process by applying a gradually increasing set of peakiness thresholds and updating the peakiness for the remaining peaks after each iteration by locating the new best minimum on either side of the peaks. Figure 3-5 provides pseudo code for our algorithm of visual beat detection that has been discussed in this chapter.

Before we conclude this section it should be noted that the final peakiness value for each peak is maintained for use later in the visual beat tracking algorithm as a quantitative value for the salience of the potential beats. The next section will discuss the method we use for finding the tempo hypotheses from the potential beats and we will then provide a detailed explanation for the beat tracking algorithm.

Definitions

u_k denotes horizontal component of optical flow frame k
v_k denotes vertical component of optical flow frame k
θ_k denotes phase for average flow at frame k
w denotes a rectangular averaging window
$*$ denotes convolution

Algorithm

```
FOREACH optical flow frame
    $k = frame\ number$
    $\theta_k = \tan^{-1} \frac{mean(v_k)}{mean(u_k)}$

    $\theta_k = \begin{cases} 180 - \theta_k, & 90 \le \theta_k < 180 \\ 540 - \theta_k, & 180 \le \theta_k < 270 \\ \theta_k, & elsewhere \end{cases}$
END FOREACH

$\theta = \theta * w$

FOREACH local maximum in $\theta$
    $k = position\ of\ maximum\ in\ \theta$
    $pkness = Peakiness(\theta, k)$
    IF $pkness > pkThreshold$
        $Beat.onset = k$
        $Beat.salience = pkness$
        Add to $Beats$ array
    END IF
END FOREACH
```

Figure 3-5: Visual beat detection algorithm.

Inter-onset Interval Clustering for Tempo Hypotheses and Visual Beat Tracking

Unfortunately the process of visual beat detection cannot be the end of our method for several reasons but it should be easily understood that the visual beat detection algorithm is not perfect. Often times there can be missed beats due to low peakiness or extra beats due to incorrect flow or camera jitter. We also wish to quantify an actual tempo for the motion of objects in an input video sequence. For these reasons we must first gather all of the knowledge that we have from the potential beats and then we can perform visual beat tracking to find the most consistent set of beats that will comprise the underlying tempo. Our algorithm additionally presents a probabilistic score for the likelihood of the computed tempo. In Dixon's algorithm for beat tracking a score is provided for selection of the final tempo but there is no upper-bound or lower-bound placed on the value. We feel that a probabilistic score not only provides more consistent information to the end user but additionally provides the user with a comparable measure of confidence for the selected tempo. More on the usefulness of the probabilistic score will be discussed in the conclusions and future work chapter of this thesis.

Inter-onset Interval Clustering

In this section we provide a description for the algorithm that collects a set of tempo hypotheses that will later be provided to the beat tracking algorithm. At the most rudimentary level this stage of the algorithm simply considers the difference between every possible combination of visual beat onsets as a potential tempo. The actual approach combines similar interval widths into clusters by means of a cluster width threshold that we have fixed at 6 for our experiments. As an output, one interval is provided for each of the resulting clusters that are simply the mean intervals. Further analysis can be done to score the final clusters allowing for the elimination of unlikely intervals or intervals which have only little visual beat evidence to support them. We in fact implemented this feature and experimented with it but ultimately do not use it as we feel that the more advanced beat tracking algorithm will also recognize these weak candidates.

The output from this algorithm is thus a collection of tempo hypotheses which are the input to the next stage along with the visual beat onsets themselves and their corresponding salience values.

Visual Beat Tracking

With a noisy input signal this algorithm becomes the most crucial step in arriving at the correct tempo. The algorithm described in the remainder of this section will utilize all of the evidence collected from previous sections in order to provide the most consistent set of visual beats and the underlying tempo as well as the probabilistic confidence that was discussed earlier. The main focus of this algorithm is to track potential chains of visual beats at the hypothesized interval of distance from each other while allowing for periods of missing visual beats, or gaps, and additionally allowing the visual beat intervals to vary a specified percentage of the effectively online interval width both plus and minus. Under this construct it is therefore possible for the tempo to drift slower and faster throughout the clip but should appear fairly consistent within small windowed regions. In the remainder of this section we will step through the visual beat tracking algorithm as we have used it; the original audio beat tracking algorithm proposed by Dixon can be found in [10].

Our visual beat tracking algorithm has three inputs the first is the collection of tempo hypotheses from the previous section, the second will be called the events and the final parameter will be the number of frames in the input video. Each event represents a single visual beat onset and contains both the frame number detected as the onset and the salience value provided through the peakiness test. The key players in the tracking algorithm will be called agents. Each agent keeps track of several bits of information as it traverses the list of events.

The information of importance to an agent is its current beat interval, the prediction for its next event, a history of all past events, the number of penalties it has, its number of successful matches and finally its score. The initialization starts by populating a list of agents, one for each tempo hypothesis for each event onset less than some startup period and initializing the agents beat interval and history correspondingly. Therefore if 5 events occur before the end of the startup period and there were 5 tempo hypotheses, the number of initial agents will be 25. In our version of the algorithm we specify the startup time as 75% percent of the number of frames in the input clip; this is an extremely large startup time but we work with relatively short videos where there may be no tempo based motion for several seconds. This then allows for a more brute force approach to ensure that all possibilities are considered. Individual agents are initialized with a score equal to the salience value of the agent's initial event normalized by the sum of all salience values. Each agent starts with zero penalties, only a single match and a prediction equal to the onset of the initial event plus the beat interval.

The main loop iterates through all events updating all agents whose next predicted onset matches the current event's onset with some tolerance. The tolerance is specified as a percentage above and below the beat interval for the agent; we typically use a value between 20-25% in either direction. This allows the tempo to drift over time as was mentioned earlier in this section. If an agent is arrived at whose predicted onset does not match the current event with a tolerance window then the agent has one added to its penalties and the history is updated with a dummy event; the dummy event having zero salience and an onset time equal to the onset of the last event in history plus the beat interval for the agent. During iterations if an agent prediction successfully matches an event as described above then the agent's history is updated with the current event, one is added to the matches, the score is updated with the normalized salience of the event multiplied by one minus the absolute value of the relative error between the event's onset and the agent's prediction normalized by the width of the tolerance window.

A much more narrow tolerance window is also used by the algorithm to allow for redundancy in the case where an event is accepted by the large tolerance window and it is incorrect. In other words an additional agent is added to the list and initialized identically to the current agent before accepting the current event whenever an agent's prediction matches within the larger tolerance window but outside of the narrower tolerance window. Figure 3-6 illustrates with an example the concept behind the two tolerance windows.

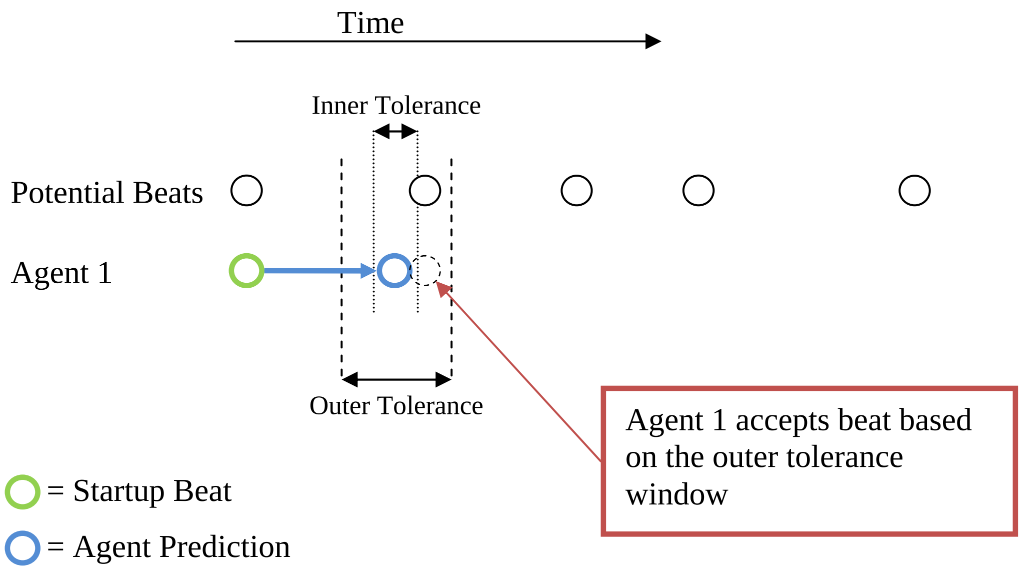


Figure 3-6: Illustrating the acceptance of new events with two tolerance windows

Finally, the algorithm wraps up by updating the agent scores and then returning the beat interval and corresponding beats for the highest scoring agent. The final update to agent scores is to incorporate information that was collected in the main loop. For each agent the score is updated with the following equation:

$$score = score \times \left(1 - nPenalties \times \frac{intervalWidth}{nFrames}\right) \times \left(\frac{nMatches}{nEvents}\right)$$

which allows for the simultaneous penalization and reward of the agent while factoring in the agent’s interval width. In the end the score is tailored to favor faster tempos as opposed to very slow tempos of up to several times the video frame rate. Again, we provide our exact algorithm in the form of pseudo code in Figure 3-7.

Definitions

Inputs:

$Beats$, array from visual beat detection
$Tempos$, array of tempo hypotheses
$nFrames$, number of frames in video clip

Outputs:

$vTempo$, visual tempo
$vScore$, probabilistic score for visual tempo
$vBeats$, array of frame numbers corresponding to the tracked visual beats

Initialization:

$startupPeriod = ceil(0.75 \times nFrames)$
$outerTolerancePre = 0.25$
$outerTolerancePost = 0.25$
$innerTolerance = 8$
$totalSalience = sum(Beats.salience)$
$nBeats = length(Beats)$
$Agents$ structure array with fields:
$(beatInterval, prediction, history, score, penalties, matches)$

Algorithm

```
FOREACH $tempo$ in $Tempos$
    FOREACH $beat$ in $Beats$
        IF $beat.onset < startupPeriod$
            $agent.beatInterval = tempo$
            $agent.prediciton = beat.onset + tempo$
            $agent.history[end + 1] = beat$
            $agent.score = \frac{beat.salience}{totalSalience}$
            $agent.penalties = 0$
            $agent.matches = 1$
            $Agents[end + 1] = agent$
        END IF
    END FOREACH
END FOREACH

FOREACH $beat$ in $Beats$
    $newAgents = \{\quad\}$
    FOREACH $agent$ in $Agents$
        $preTolerance = ceil(outerTolerancePre \times agent.beatInterval)$
        $postTolerance = ceil(outerTolerancePost \times agent.beatInterval)$
        $timeout = agent.beatInterval + postTolerance$
        WHILE $beat.onset - agent.history[end].onset > timeout$
            $nBeat.onset = agent.history[end].onset + agent.beatInterval$
            $nBeat.salience = 0$
            $agent.history[end+1] = nBeat$
            $agent.penalties = agent.penalties + 1$
            $agent.prediction = agent.prediction + agent.beatInterval$
        END WHILE

        $tolWidth = preTolerance + postTolerance$

        WHILE $agent.prediction + postTolerance < beat.onset$
            $agent.prediction = agent.prediction + agent.beatInterval$
        END WHILE

        IF $agent.prediction - preTolerance \leq beat.onset\ AND$
            $beat.onset \leq agent.prediction + postTolerance$
            IF $|agent.prediction - beat.onset| > innerTolerance$
                $newAgents[end+1] = agent$
            END IF

            $error = beat.onset - agent.prediction$
            $relError = \frac{error}{tolWidth}$
            $agent.matches = agent.matches + 1$
            $agent.beatInterval = agent.beatInterval + relError$
            $agent.prediction = beat.onset + agent.beatInterval$
            $agent.history[end+1] = beat$
            $scoreInc = \left(\frac{beat.salience}{totalSalience}\right) \times (1.0 - |relError|)$
            $agent.score = agent.score + scoreInc$
        END IF
    END FOREACH
    Add $newAgents$ to $Agents$ array
END FOREACH

FOREACH $agent$ in $Agents$
    $agent.score = agent.score \times \left(1.0 - \frac{(agent.penalties \times agent.beatInterval)}{nFrames}\right)$
        $\times \frac{agent.matches}{nBeats}$
END FOREACH

Return the agent with the highest score
```

Figure 3-7: Algorithm for Visual Beat Tracking.

In concluding this chapter we have presented a novel approach to discovering visual beats in video sequences and further explored our approach to extrapolating these beats into a quantitative visual tempo for object motion. We will discuss in the following chapter how this visual tempo can be utilized along with state-of-art audio tempo analysis to perform very good dance detection.

CHAPTER FOUR: APPLYING TEMPO ANALYSIS TO DANCE ACTIVITY RECOGNITION

One main goal of this thesis is to provide concrete evidence for the justification behind our defined visual tempo. First we wish to adequately put forth a framework for dance activity recognition and second we believe that our formulation and quantification of visual tempo is not only unique when compared to previous work but that it also captures aspects of object motion that like film tempo can be used as a low level feature or clue to uncovering higher level concepts. This will be the basis of this chapter in which we will demonstrate how our visual tempo can be utilized to perform dance activity recognition effectively and accurately through a non-training based approach.

For the purposes of this work the activity of dance in video will be considered as significant motion which transitions at a rate equivalent to that of the beats in the audio track. This means that we will consider a video to contain the activity of dance if there is first music present in the audio track and second transitions in the motion of the visual content occur, as described in our chapter on Computing Visual Tempo, at the same rate as the beat in the audio as given by BeatRoot. Many music detection algorithms have been proposed as in [14] with high accuracy and therefore in order to best demonstrate our technique we will consider the task of music detection trivial by considering only video clips which are known to contain dominant musical content. We also place no restrictions on the camera motion in the visual content however it should be realized that the method proposed by this work will perform optimally with conditions of static camera as camera motion tends to dampen the underlying motion of moving objects.

Our algorithm attempts to match audio and visual tempos in video content through the use of independent tempo analysis. In this chapter we will describe our algorithm providing detailed information for the various components through a series of ordered sections.

Video Segmentation

As the first stage in our proposed framework we perform segmentation of input videos into smaller clips of generally around 300 frames. There are two main reasons for the inclusion of this rather trivial step. The first reason is rather simple, smaller video clips are easier to work with but secondly we wish to allow for the use of general video clips which may contain the activity of dance in only a short portion this can often be seen in home videos like many of those found on sources such as YouTube. By analyzing smaller clips we are also allowing for different segments to vary in their audio and visual tempos. This is important for longer clips of dance recitals for instance that often times contain more than one song. The list could go on and on with possibilities but in fact there is one potential drawback to using smaller video segments. Both the audio and visual tempos are acquired through a similar method of beat tracking which could benefit from the availability of more time to discover the underlying tempo. We believe that this is outweighed by the benefits and experimental results show that the segment lengths we use are plenty effective for this purpose.

Finding the right tools to prepare input data was not an easy task as we wanted to evaluate the usefulness of our approach with all types of videos from professional quality such as music videos to amateur videos from YouTube. Once we had the right combination of tools the task was no problem so for completeness we mention the packages that we utilized for data preparation in Table 4-1.

Table 4-1: Table presenting the tools used for data preparation

Video Conversion	FFmpeg – an open source utility available for most operating systems http://ffmpeg.mplayerhq.hu/
Video Splitting	VirtualDub – a utility providing many useful tools for working with AVI format videos http://www.virtualdub.org/
Audio Extraction	FFmpeg

Audio and Visual Analysis

The most important aspect of our work for dance detection is in the ability to compare audio and visual tempos. Throughout the next few sections we will discuss the use and interactions of the various components of the framework and we will describe our method of tempo comparison.

Music Detection

The knowledge of whether or not music occurs in the audio of a video clip is important when trying to perform dance detection as we have defined it. In the related work chapter we gave an overview on some of the various techniques used by current methods. In the literature it is obvious that considerable work has been done on distinguishing between various classes of audio. In fact as we discussed in our related work rather successful work has been done in order to distinguish between various styles of dance music. For these reasons we make the assumption that music versus non-music classification is trivial and all video clips that we analyze will be manually selected because they contain dominate musical audio. As we mentioned this section is included simply because the concept is a valid piece to the overall framework and should not be overlooked.

Audio Tempo Detection

For this stage of the framework we utilize the Java based software package mentioned throughout this thesis called BeatRoot. In the related work we discussed a little bit on how the program works and how it relates to our visual tempo algorithm. Since BeatRoot is considered a beat tracking algorithm the output is simply the timestamp for the audio beats and we therefore take advantage of the fact that our input segments are short. We can simply estimate an overall audio tempo by averaging the inter-beat intervals. A tempo value from a strictly musical perspective would be measured in units of beats per minute, however in our experiments we wanted to know at what frame each audio beat mapped to in the video. Therefore we convert the list of beat times into frame numbers and simply average the inter-frame intervals for a value in units of frames per beat.

Visual Content Analysis

In this stage we perform the visual tempo analysis described in the previous chapter. The visual tempo and score can be used to determine if there is strong evidence of a visual tempo but individually this step is not enough to detect dancing. Unfortunately dancing is not the only activity with tempo based motion as we mentioned in previous chapters. Therefore neither the audio nor the visual tempos can be used independently for dance detection and in the next section we will discuss how we generally combine the tempo information.

Combining Audio and Visual Tempos

In order to consider a video to contain dance we must match the audio and visual tempos. From our visual tempo analysis we provide a probabilistic score from the visual beat tracking stage that allows our algorithm to better decide whether a tempo based motion exists or not. Unfortunately from the audio analysis we do not have such a score but this could be one avenue for future work. Nevertheless, we must compare tempos and in doing so we generally allow the tempos to match within a threshold of each other. The typical threshold used is 10% of the audio tempo. Therefore if the visual tempo matches within 10% of the audio tempo we would declare the video to contain dance. As we mentioned in the related work chapter, there is a known issue with current beat tracking methods in which the tempo reported will be proportional to the actual tempo but could be twice or half the tempo for instance. We find this to be the case in many of the dance videos that the algorithm gets wrong but unfortunately allowing such variations increase the false positive rate significantly. We therefore do not accommodate for such conditions when matching tempos and such cases would be consider non-dance videos. In the next chapter we will report aggregate results through the use of a precision versus recall curve which shows the results for a range of operating thresholds and more will be discussed on this at that time.

CHAPTER FIVE: EXPERIMENTS & RESULTS

In this chapter we will demonstrate the usefulness of our visual beat detection and visual beat tracking algorithms utilizing two standard activity recognition databases highly cited in Computer Vision literature. The various activities in these datasets are the common activities of running, walking, jogging, jumping and skipping among others. Although many of the activities in these datasets are periodic none of them are performed with a strict tempo as is typically the case with dancing however these datasets will adequately demonstrate the effectiveness of the visual beat detection algorithm and will further show the flexibility of the visual beat tracking algorithm. A few activities from these datasets were unusable because no dominant motion could be isolated due to a small number of pixels on the moving parts and camera jitter. Unfortunately one of these activities is handclapping so we have substituted our own handclapping sequence as we wish to validate our opening example.

Datasets

We have collected data from several sources in order to validate our approach to visual beat detection. The first video sequence we will demonstrate is one that we created. This sequence was filmed using a standard digital handheld camera and has never been used in any other work. The next dataset was introduced in [4] and is commonly referred to as the Irani action dataset. The dataset includes the binary masks for the person in each sequence obtained through background subtraction. We chose to use this dataset because it has been highly cited in action recognition literature over the last 2 years and is therefore familiar to Computer Vision researchers. One final dataset that we use is a collection of various dance videos from YouTube and a handful that we have recorded ourselves.

Experiments

Clapping Sequence

In Figure 5-1 we present the results on a sequence of clapping. From the figure we can see that most of the visual beats were successfully found and those that were missed or rejected were interpolated by the tracking algorithm. When considering sequences such as this one where the movements do not follow a strict tempo, as is the case with dance, it is unfair to ask for one from the algorithm however this sequence was generated with fairly consistent movements; therefore we will provide a comparison. The clapping sequence consisted of 20 transitions in 300 frames yielding a rate of 15 frames per beat and the estimated tempo from our algorithm was 15.1 frames per beat.

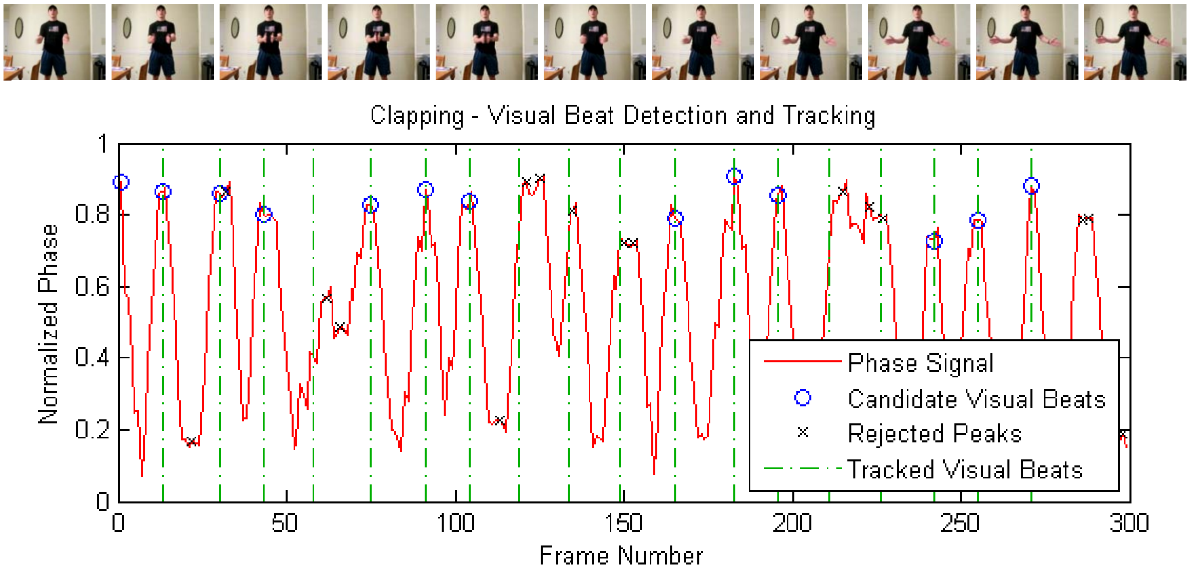


Figure 5-1: **Result on clapping sequence.**

Irani Dataset

The following are results on selected sequences from the Irani dataset [4]. The result for a sequence on jumping jacks is shown in Figure 5-2. The result for a sequence on running is shown in Figure 5-3. The result for a sequence on walking is shown in Figure 5-4. The result for a sequence on long jumping is shown in Figure 5-5. The result for a sequence on waving is shown in Figure 5-6.

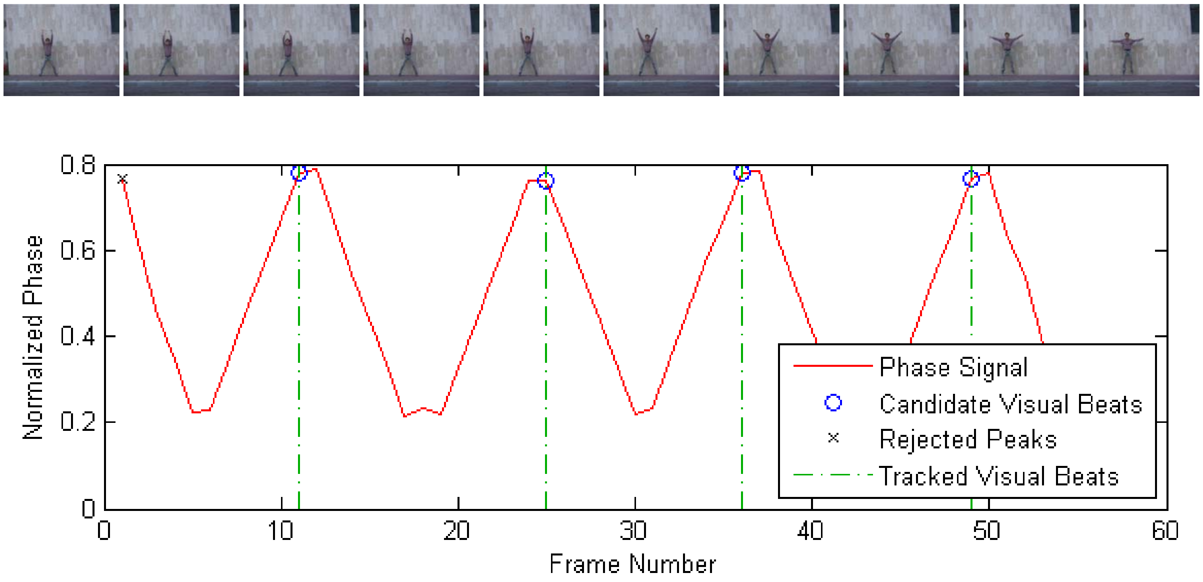


Figure 5-2: Result on jumping jacks sequence.

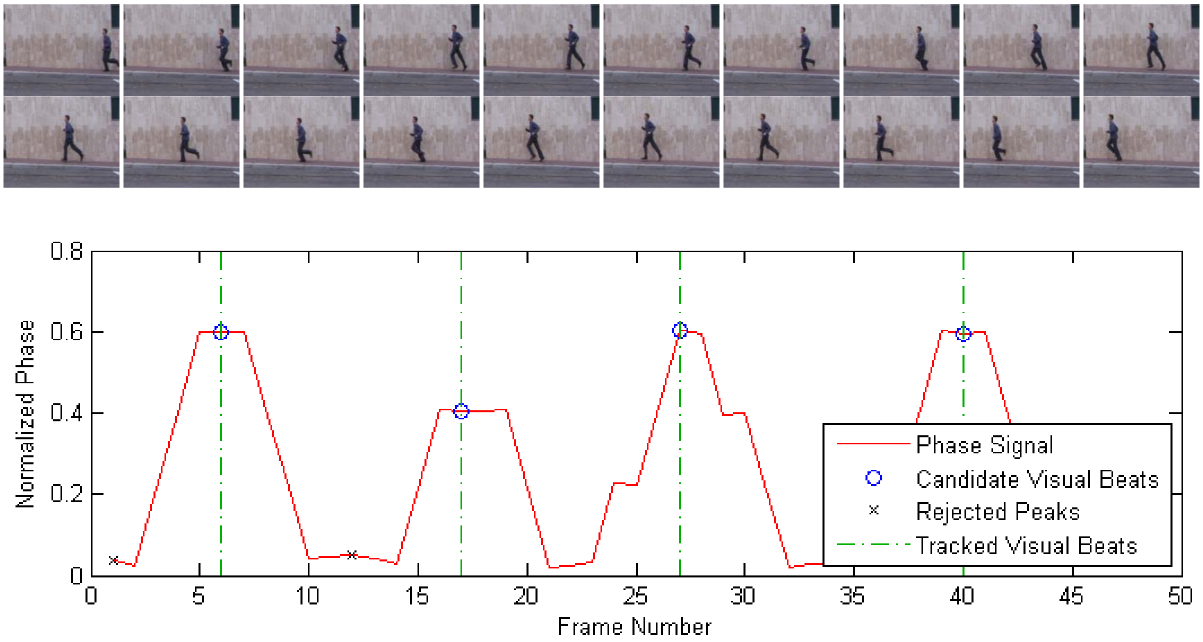


Figure 5-3: **Result on running sequence.**

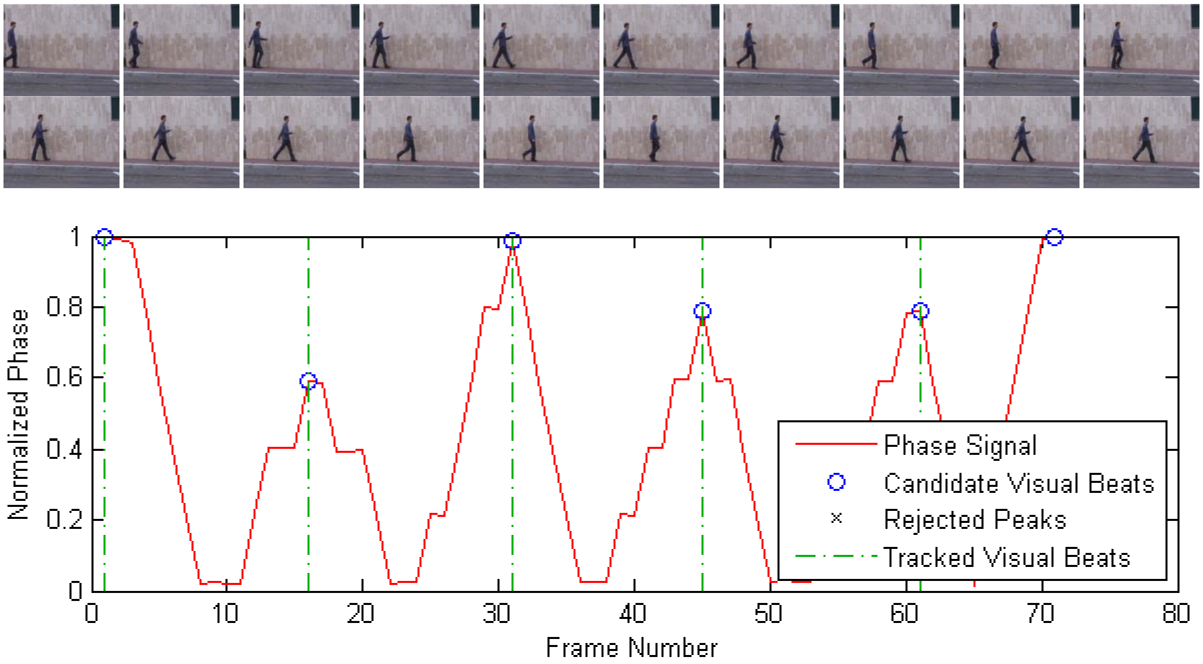


Figure 5-4: **Result on walking sequence.**

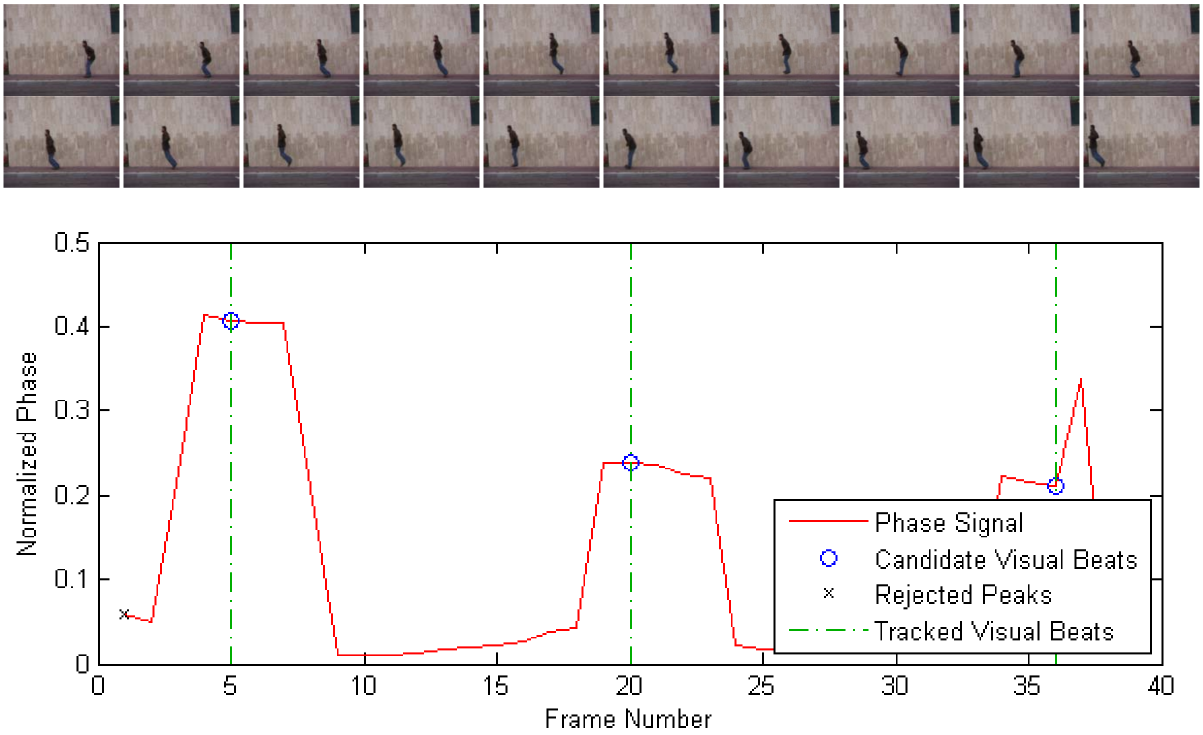


Figure 5-5: Result on long jumping sequence.

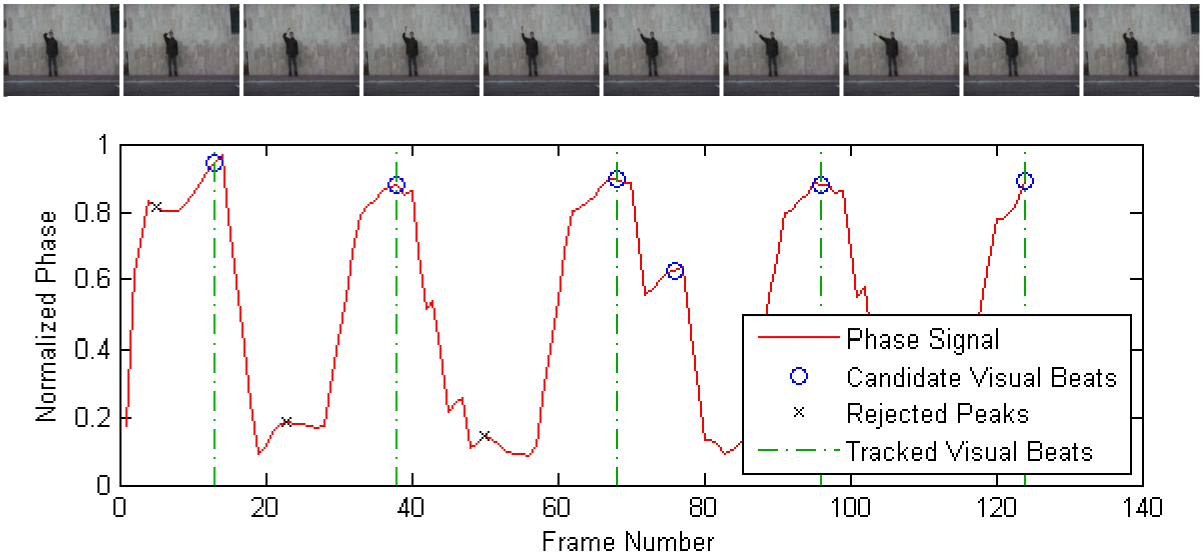


Figure 5-6: Result on waving sequence.

Dance Activity Data

In this section we will present results for our algorithm on our own data for the activity of dance. This data allows for the evaluation of both the visual beat detection algorithm and the visual beat tracking algorithm. The audio tempos compared to are computed as the average inter-onset interval provided by BeatRoot for the corresponding audio sequence.

The results in Figure 5-7 are generated from a 300 frame video sequence of the Macarena. The audio tempo we obtained for this song is 17.5 video frames per beat and the output from our algorithm is 17.8 clearly demonstrating the correlation between our defined visual tempo and the music tempo.

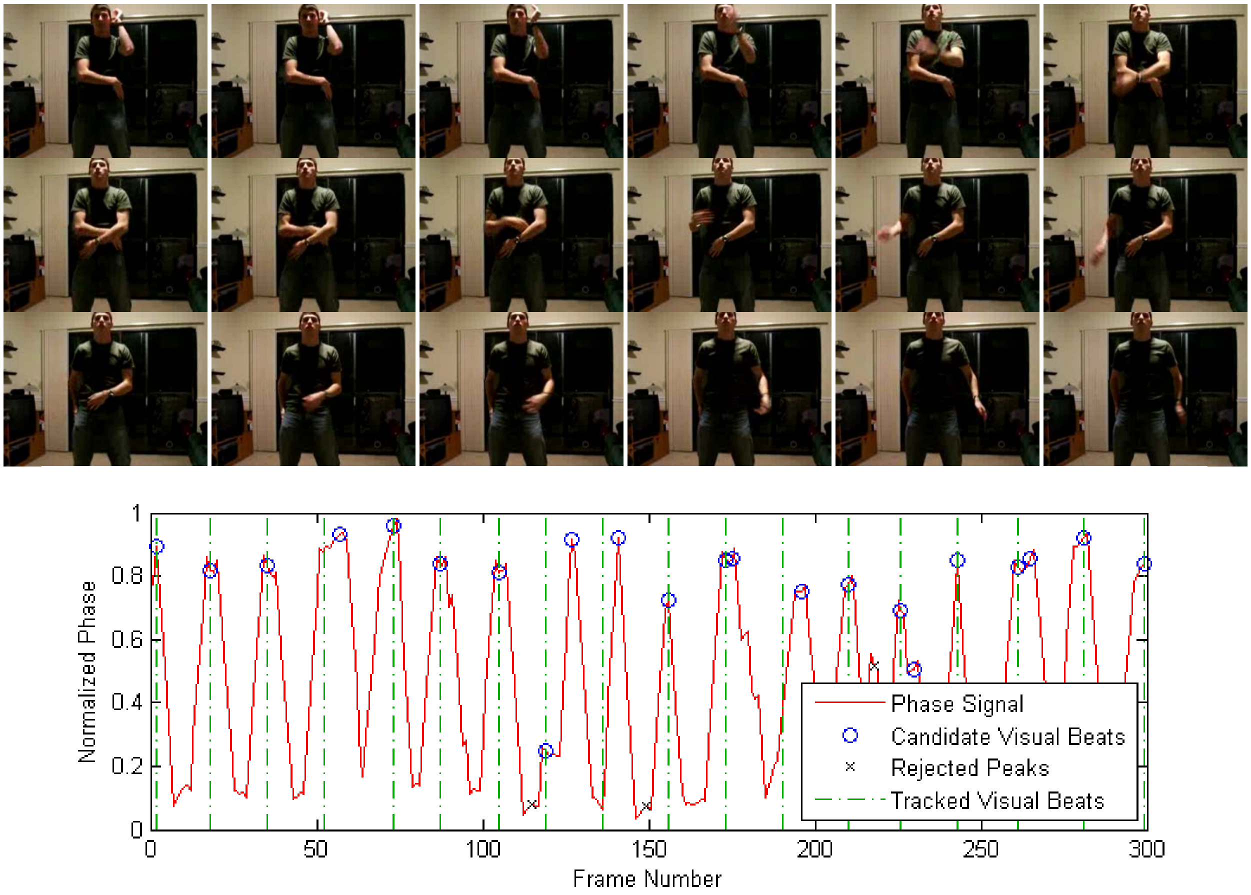


Figure 5-7: Result of visual beat detection and tracking for the Macarena.

Our next sequence is 300 frames from the Electric Slide with an audio tempo of 16.65 video frames per beat. The result from our algorithm on this sequence is 16.68 video frames per beat and the visual beat detection and tracking results are shown in Figure 5-8.

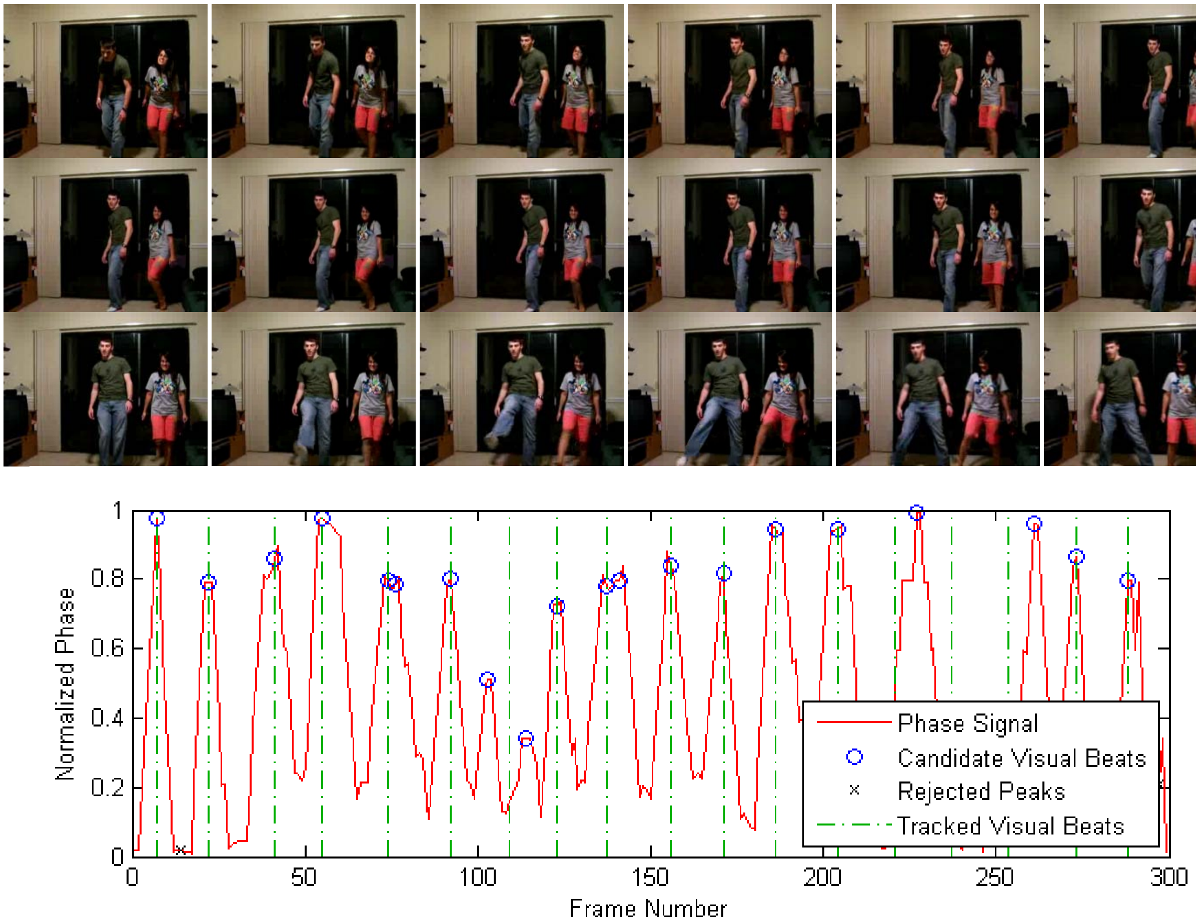


Figure 5-8: Result of visual beat detection and tracking for the Electric Slide.

The final sequence from our dataset that we will provide is a 300 frame clip from the YMCA. This clip has an audio tempo of 14.1 video frames per beat and our algorithm tracked the visual tempo at 13.4 video frames per beat. The result of visual beat detection and tracking for this clip of the YMCA is provided in Figure 5-9.

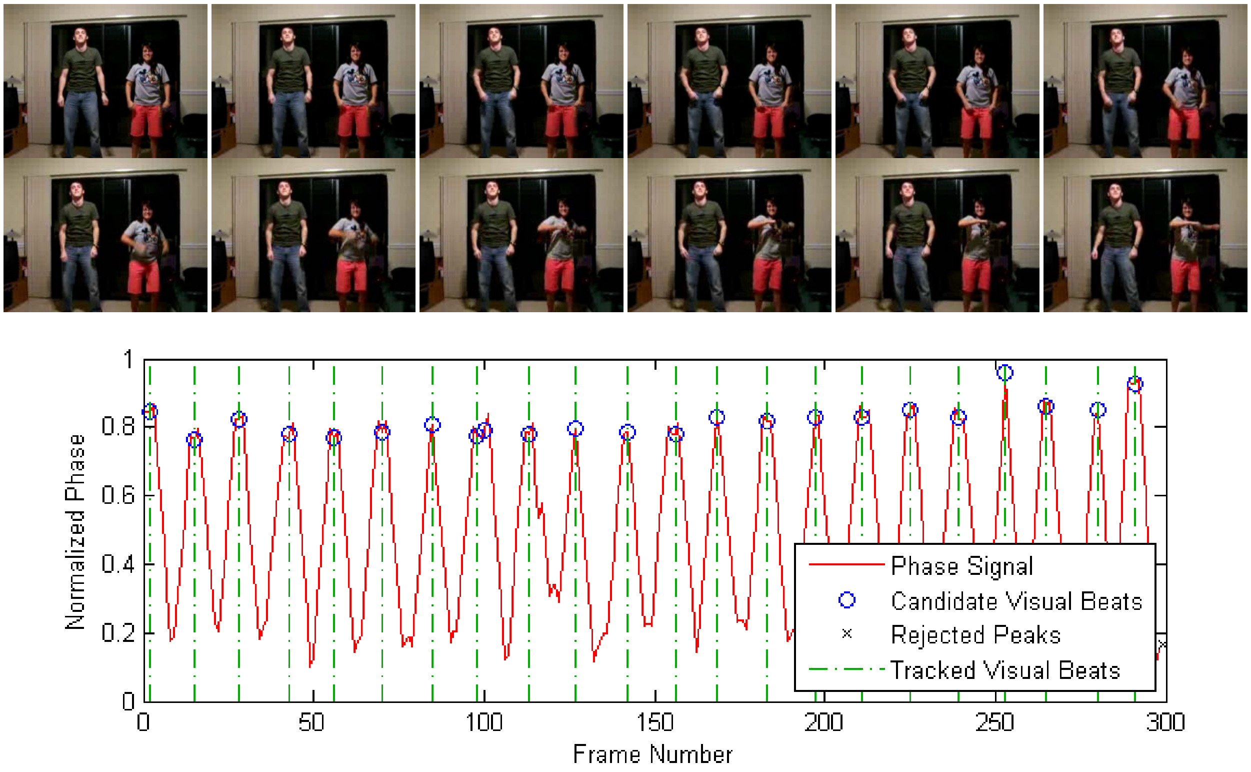


Figure 5-9: **Result of visual beat detection and tracking for the YMCA.**

YouTube Dance Data

Until now we have presented results on videos that were captured in relatively controlled circumstances by ourselves and other activity recognition researchers. In order to rigorously test our algorithm we decided to collect videos of dancing from YouTube and we will now present the results on some of the videos we collected to demonstrate robustness. It is worth noting that these videos typically depict groups of people dancing where as previous videos we presented only contained one or two. Also included in these videos are various styles of music and dance and more on this will be discussed as we present each result.

Figure 5-10 depicts the result of visual beat detection and tracking for a YouTube video (dQEexOtKd8A) of a South Indian dance team. The audio tempo for the sequence of 300 frames is 14.9 video frames per beat and the output of our algorithm is 15.3 video frames per beat.

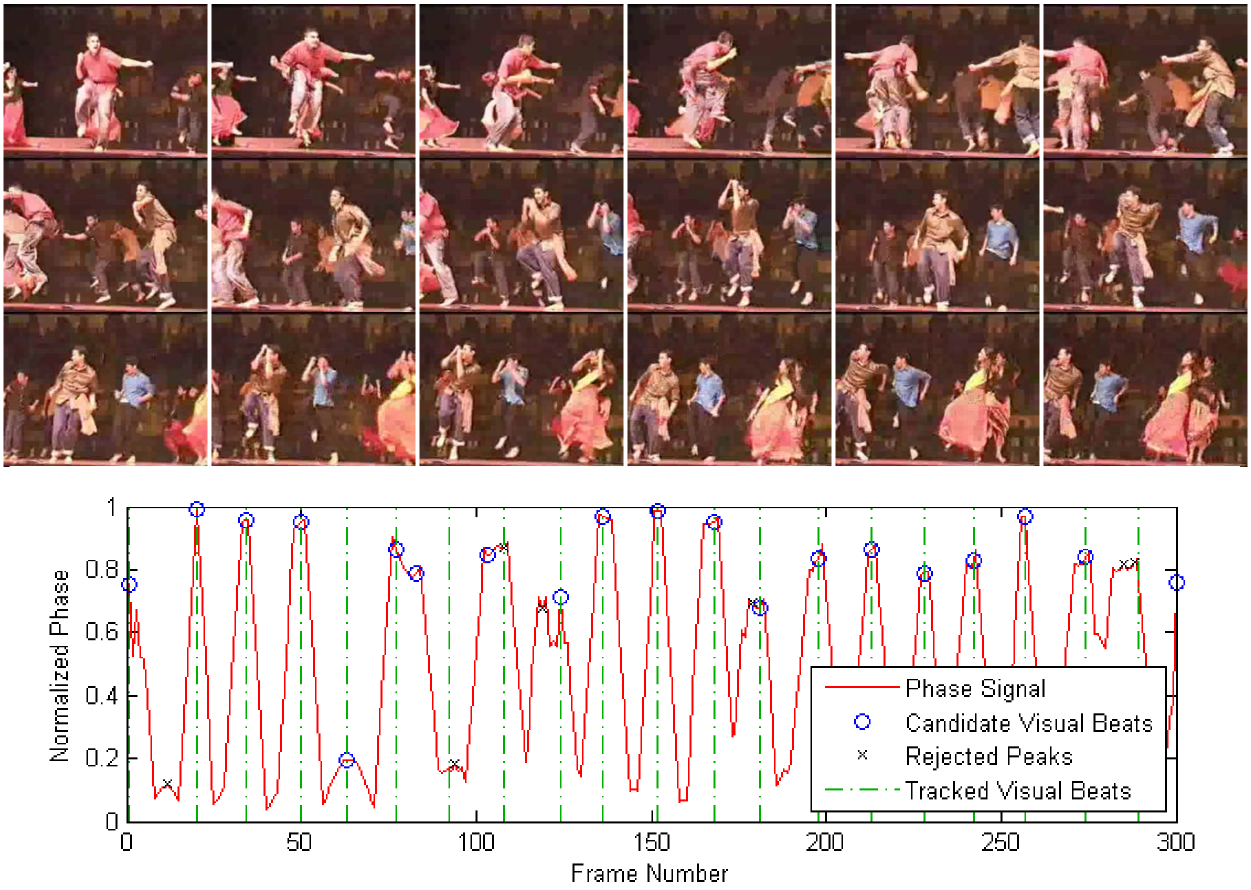


Figure 5-10: Result on visual beat detection and tracking of YouTube video dQEexOtKd8A.

This next sequence is also a YouTube video (6UV3kRV46Zs) but this video contains a group of dancers demonstrating the Chicken Dance. The result for visual beat detection and tracking is given in Figure 5-11. The audio tempo detected in this sequence is 14.7 video frames per beat while the visual tempo detected by our algorithm is 15.5 video frames per beat.

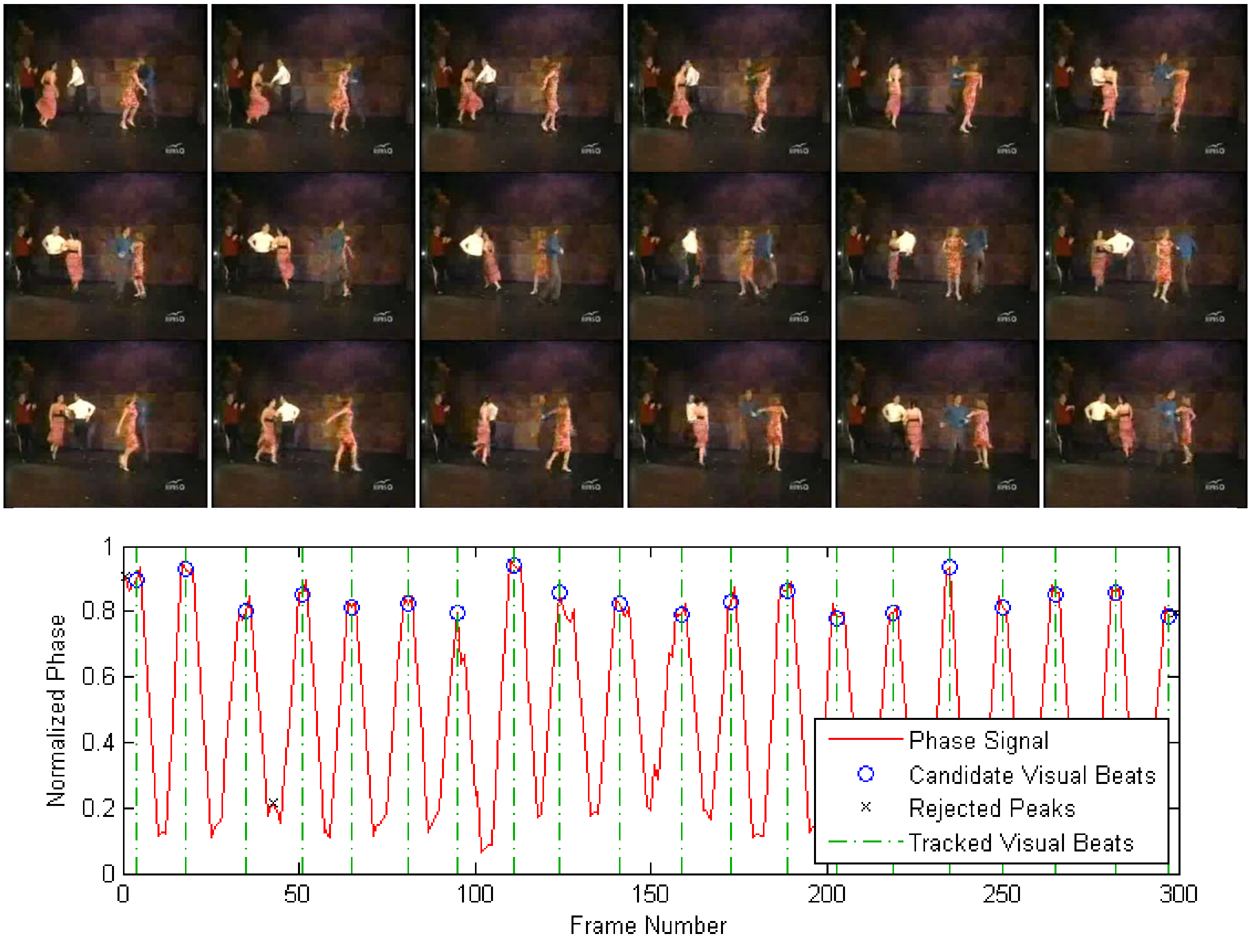


Figure 5-11: Result on visual beat detection and tracking of YouTube video 6UV3kRV46Zs.

The last YouTube dance video (kmK2hV4YVEQ) that we will present here depicts a group of dancers performing a break dance style routine. This sequence additionally includes significant camera motion, changes of view and camera, and various lighting effects. The visual beat detection and tracking results are presented in Figure 5-12. The audio tempo from this sequence is 15.9 video frames per beat and the visual tempo identified by our algorithm is 16.0 video frames per beat. As can be seen in the figure the phase signal is noisier than in previous examples yet the beat tracking algorithm is able to successfully locate a strong pattern of visual beats.

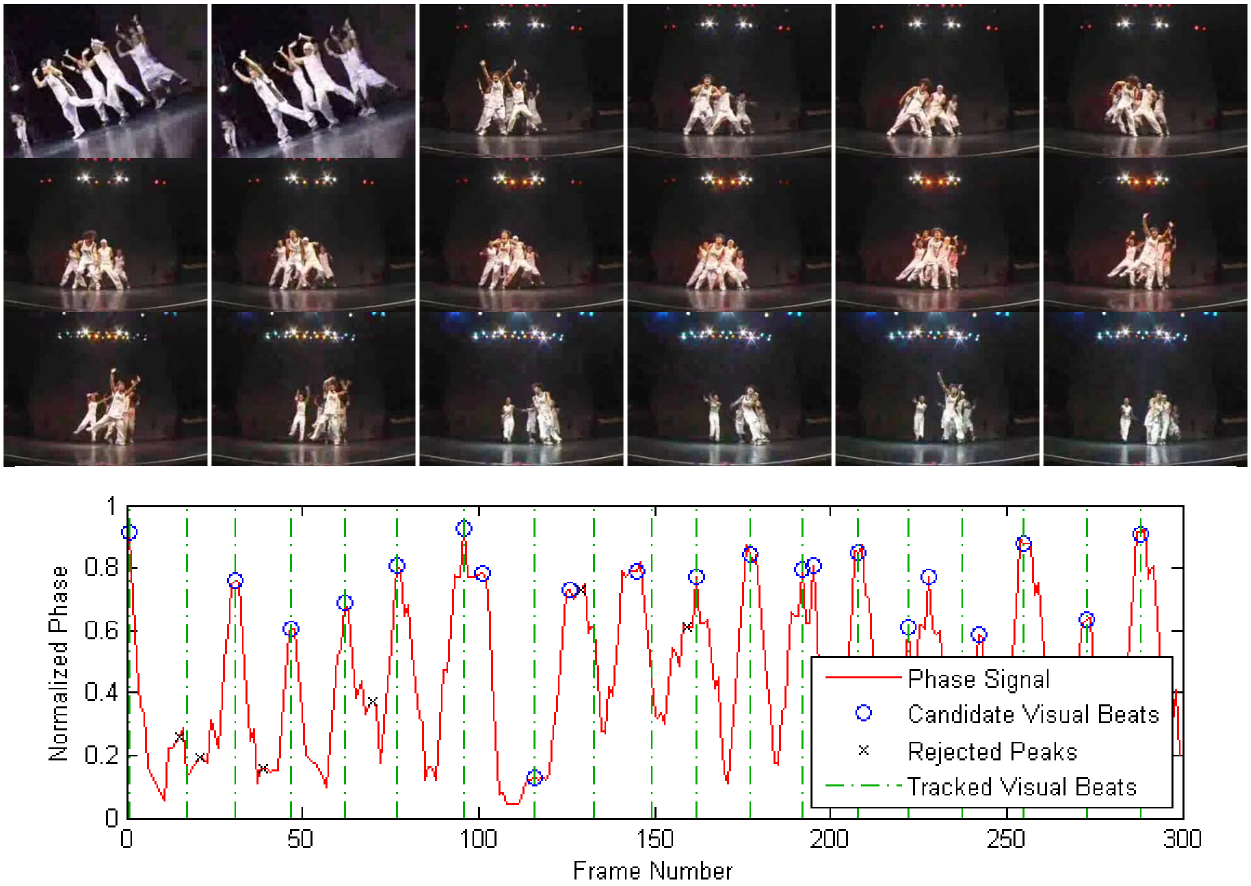


Figure 5-12: Result on visual beat detection and tracking of YouTube video kmK2hV4YVEQ.

Aggregate Results

In order to demonstrate the effectiveness of our proposed algorithm we collected a dataset consisting of mostly YouTube videos but there are also several videos that we recorded using a handheld digital camera. The set consists of 65 dance videos and 60 non-dance videos. Each video clip is 300 frames and both the positive and negative sets contain music in the audio. We evaluated our non-training based tempo method on this dataset and a resulting confusion matrix is given in Figure 5-13.

		Detected Class	
		Dance	**Non-Dance**
Actual Class	**Dance**	43	22
	Non-Dance	5	55

Figure 5-13: Confusion matrix for the proposed Tempo method on the dance dataset.

Figure 5-14 presents results comparing our method to the method proposed by Scovanner et al in [2]. Since the method of Scovanner et al uses random points we provide results on 4 separate runs of the method and additionally the features used during these runs are the gradient magnitude for a cube around each random point. It can be observed in the figure that our proposed method can correctly identify around 65% of the positives while maintaining a very high precision of almost 95%. The precision versus recall curve for our method is generated by getting the ratio of the absolute difference between the audio and visual tempos to the audio tempo alone for each video; this is summarized by the following equation.

$$score = \frac{|audioTempo - visualTempo|}{audioTempo}$$

We then threshold the resulting score at a fixed interval and compute both precision and recall assuming every video with a score higher than the threshold is classified as dance. The same technique is applied to the SVM method except the scores used to threshold are the SVM confidence values.

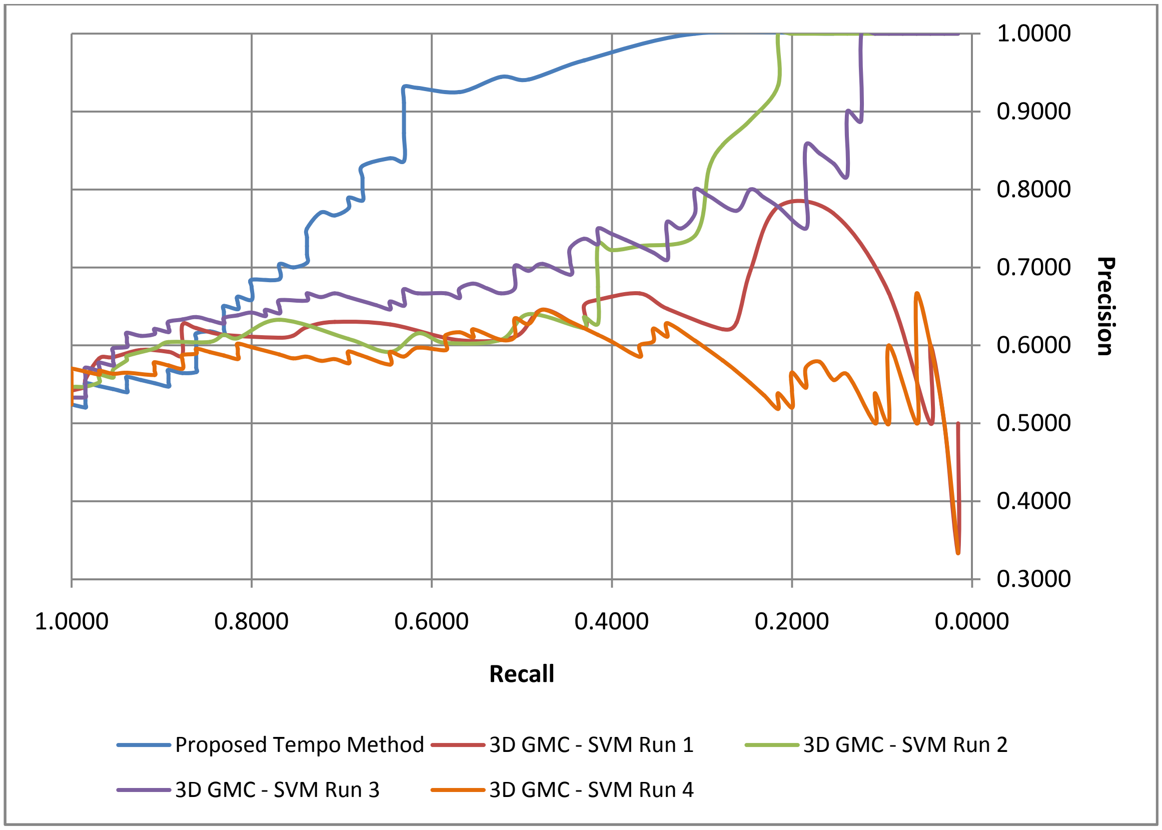


Figure 5-14: Dance detection results comparison

We feel that these results adequately illustrate our claim that training based methods cannot effectively detect dance as an unconstrained problem. Even utilizing a leave-one-out testing strategy we have shown that the training is insufficient to classify the dataset. Our non-training based approach allows robustness to aspects that training based methods typically struggle with such as learning of background clutter or the learning of a particular type of data characterized by acquisition noise or other common effects such as variations in lighting. Additionally training based approaches cannot often generalize to changes in view without additional training but our algorithm is not limited by changes in view.

CHAPTER SIX: CONCLUSION

Summary

Throughout this thesis we have considered previous work including activity recognition, audio tempo analysis, and music classification. In chapter one we defined important terms related to this thesis; we stated our exact definition of visual tempo as we use it and provided visual evidence for a better understanding. In addition, we formalized our goals by first stating our problem and then outlining our proposed solution. Chapter two focused on discussing the current state-of-the-art in various areas of research as they pertain to this thesis. We also mentioned previous work that is most similar to our work and presented the strengths and weakness of those approaches. Chapter three formalized our method for visual beat detection and demonstrated how to compute visual tempo by our definition and as it relates to music. In chapter four we presented a framework utilizing our proposed visual tempo to perform dance activity recognition. Finally in chapter five we demonstrated the existence of visual beats in a variety of activities using a standard dataset and further proved the effectiveness of our visual tempo measure through impressive results on our proposed dance recognition dataset composed of mainly unconstrained consumer videos from YouTube.

Conclusion

In concluding this thesis we wish to reiterate several of the important aspects of our work and the contributions we have made. Through our work we have reintroduced tempo to Computer Vision as a quantifiable feature of visual motion in the same regard as tempo to music research and music in general. In so doing we have additionally introduced the visual beat as a new term and feature for use in visual motion research. We provided a novel algorithm for visual beat detection and developed effective enhancements to a state-of-the-art algorithm for beat tracking in music research. The proposed algorithm is a non-training based approach which is robust to many of the common issues faced by training based methods such as view changes and variations in data including light effects and sensor type. We have finally assembled a challenging dataset for the evaluation of dance recognition methods consisting of mainly YouTube videos. These videos span several dance styles including dances by different cultural groups and additionally varying numbers of performers and types of performances from professional stage dances to amateurs performing at home. We believe that this large variety within the dataset further proves the effectiveness of our algorithm and demonstrates a robustness that is uncommon to other activity recognition methods.

Future Work

In this thesis we have provided only one solution to visual beat detection but as was mentioned earlier there are many more sophisticated methods used to accomplish beat detection in music research. More work should be done to determine if these methods are effective for visual motion. Additional work should be conducted in either case to find a method more capable of finding the specific onset of visual beats. Our work effectively finds potential beats but due to averaging it becomes unclear of the exact onset for each visual beat. Methods capable of onset detection of beats in visual motion may prove more powerful when performing dance activity recognition. This may also prove useful for future applications of visual beats as there is still more work that can be done in discovering these uses. Our probabilistic output from visual beat tracking could support additional work in the application of learning algorithms to visual tempo analysis and additionally more sophisticated scoring schemes could be sought.

LIST OF REFERENCES

[1] **Naphade, Milind, et al.** *Large-Scale Concept Ontology for Multimedia.* 3, July-September 2006, IEEE MultiMedia, Vol. 13, pp. 86-91.

[2] **Scovanner, Paul, Ali, Saad and Shah, Mubarak.** *A 3-Dimensional SIFT Descriptor and its Application to Action Recognition.* Augsburg, Bavaria, Germany : s.n., 2007. ACM MM. 978-1-59593-701-8.

[3] **Adams, Brett, Dorai, Chitra and Venkatesh, Svetha.** Formulating Film Tempo. [ed.] Chitra Dorai, Svetha Venkatesh and Mubarak Shah. *Media Computing: Computational Media Aesthetics.* s.l. : Kluwer Academic Publishers, 2002, 4, pp. 57-84.

[4] **Blank, Moshe, et al.** *Actions as Space-Time Shapes.* Beijing, China : IEEE Computer Society, 2005. IEEE International Conference on Computer Vision. Vol. 2, pp. 1395-1402. 1550-5499.

[5] **Polana, Ramprasad and Nelson, Randal.** Temporal Texture and Activity Recognition. [ed.] Mubarak Shah and Ramesh Jain. *Motion-Based Recognition.* s.l. : Kluwer Academic Publishers, 1997, pp. 87-124.

[6] **Schuldt, Christian, Laptev, Ivan and Caputo, Barbara.** *Recognizing Human Actions: A Local SVM Approach.* s.l. : IEEE Computer Society, 2004. International Conference on Pattern Recognition. Vol. 3, pp. 32-36. 1051-4651.

[7] **Cutler, Ross and Davis, Larry.** *Robust Real-Time Periodic Motion Detection, Analysis, and Applications.* 8, August 2000, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, pp. 781-796.

[8] **Polana, Ramprasad and Nelson, Randal.** *Detection and Recognition of Periodic, Nonrigid Motion.* 3, s.l. : Springer Netherlands, June 1997, International Journal of Computer Vision, Vol. 23, pp. 261-282. 0920-5691 (Print), 1573-1405 (Online).

[9] **Lee, Shih-Hung, Yeh, CHia H. and Kuo, C.-C. Jay.** *MTV-style home video generation via tempo analysis.* [ed.] Chang Wen Chen. 2004. SPIE. Vol. 5600, pp. 238-249.

[10] **Dixon, Simon.** *Automatic extraction of tempo and beat from expressive performances.* 1, 2001, Journal of New Music Research, Vol. 30, pp. 39-58.

[11] **Dixon, Simon.** *Onset Detection Revisited.* Montreal, Canada : s.n., 2006. International Conference on Digital Audio Effects.

[12] **Gouyon, Fabien and Dixon, Simon.** *Dance music classification: A tempo-based approach.* Barcelona, Spain : s.n., 2004. International Conference on Music Information Retrieval.

[13] **Lowe, David G.** *Distinctive image features from scale-invariant keypoints.* 2, s.l. : Kluwer Academic Publishers, November 2004, International Journal of Computer Vision, Vol. 60, pp. 91-110. 0920-5691.

[14] **Gerhard, David.** *Audio Signal Classification.* School of Computing Science, Simon Fraser University. Burnaby, BC, Canada : s.n., 2000. Ph.D. Depth Paper.

[15] URAPIV - where Matlab meets Particle Image Velocimetry. [Online] http://urapiv.wordpress.com/.

[16] **Camus, Theodore Armand.** *Real-Time Quantized Optical Flow.* Como, Italy : IEEE Computer Society Press, 1997. IEEE Workshop on Computer Architectures for Machine Perception. pp. 71-86.

[17] **Horn, Berthold K.P. and Schunck, Brian G.** *Determining Optical Flow.* Cambridge, MA, USA : Massachusetts Institute of Technology, 1980.

[18] **Lucas, B. D. and Kanade, T.** *An Iterative Image Registration Technique with an Application to Stereo Vision.* Vancouver : s.n., 1981. Proceedings of the 7th International Joint Conference on Artificial Intelligence. pp. 674-679.

[19] **Proesmans, Marc, et al.** Determination of Optical Flow and its Discontinuities using Non-Linear Diffusion. *ECCV '94: Proceedings of the Third European Conference - Volume II on Computer Vision.* London, UK : Springer-Verlag, 1994, pp. 295-304.

[20] **Mubarak, Shah.** *Fundamentals of Computer Vision.* Orlando : s.n., 1997.