Thesis for degree of Master of Engineering

Automatic Music Mixing Method using Machine Learning Tools

By

Kanyange Pamela

Department of IT convergence and application engineering,

The Graduate School

Pukyong National University

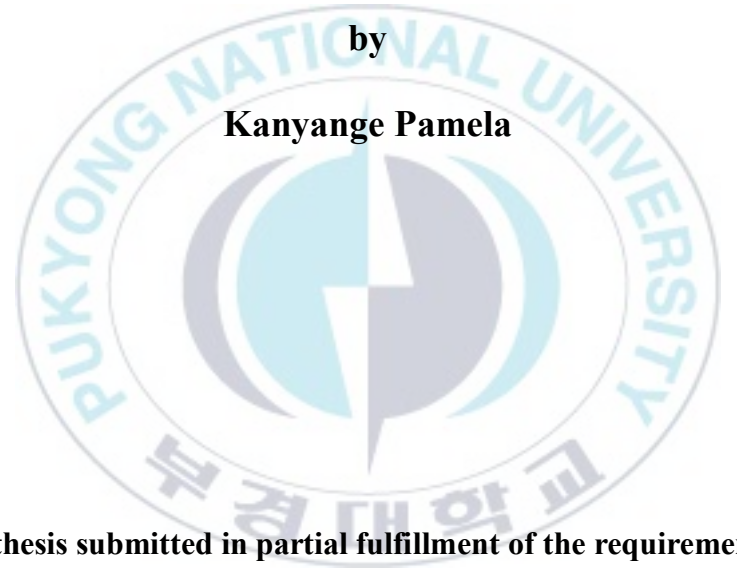
July 2017

Automatic Music Mixing Method using Machine Learning Tools
(기계학습 도구를 사용한 음악 믹싱 기법)

Advisor: Prof. Bong-Kee Sin

by

Kanyange Pamela



A thesis submitted in partial fulfillment of the requirements

for the degree of

Master of Engineering

In the Department of IT convergence and application engineering,

The Graduate School

Pukyong National University

July 2017

Automatic Music Mixing Method using Machine Learning Tools

A dissertation

By

Kanyange Pamela

Approved by:

_____

(Chairman) **Prof. 권기룡**

_____ _____

(Member) **Prof. 송하주** (Member) **Prof. 신봉기**

July 2017

Table of contents

List of figures

List of tables

List of abbreviations

SVM: Support Vector Machine

MFCC: Mel-frequency cepstral coefficients

HMM: Hidden Markov Model

DHMM: Discrete Hidden Markov Model

CHMM: Continuous Hidden Markov Model

FT: Fourier transform

DCT: discrete cosine transform

STFT: short-time Fourier transform

IFT: Inverse Fourier transform

BPM: Beats per minute

기계학습 도구를 사용한 음악 믹싱 기법

Pamela Kanyange

부경대학교 대학원　IT 융합응용공학과

요약

음악은 인간이 표현하는 감정의 언어이다. 우리는 집에서 파티를 할 때에도, 친구들과 기분 좋게 한 잔을 기울일 때에도, 고요한 밤 잔잔한 라디오 방송을 청취할 때에도 상황에 맞는 음악을 즐기길 원한다. 본 논문에서는 이러한 상황들에서 우리가 적절한 음악을 고를 수 있도록 도움을 주는 시스템을 제안한다.

본 연구는 수많은 음악 데이터에서 자동적으로 사용자의 기호와 음악들 간의 유사성에 따라 선택하고 자동적으로 곡을 믹싱 하는 방법을 제안한다. 비록 사용자가 해당 음악에 관한 아무런 정보가 없을 지라도, 제안 시스템은 음악의 특성에 따라 일련의곡을 매끄럽게 이어서 메들리를 만들어 준다.

음악들 간의 유사성을 판단하기 위하여 세 가지로 음악의 특성을 분석하였다. 첫째 음악의 장르, 두 번째는 음악의 조성, 마지막으로 음악의 빠르기이다. 이 세 가지 특성을 이용하여 음악 재생목록을 구성하고 믹싱한다.

본 논문에서는 음악 특징 추출과 믹싱 작업을 하기 위해 기계학습 기법인 지지 벡터 머신 (Support Vector Machine, SVM) 과 은닉 마르코프 모델 (Hidden Markov Model)을 적용하였다. 음악 장르 분류와 키 식별은 각각 90.6 %와 87.5 %의 정확도를 얻었다. 그리고 믹싱결과 90%의 호평을 얻었다.

Keywords: SVM, Markov chain, HMM, music genre classification, music key identification, tempo estimation, music mixing.

Chapter 1. Introduction

1.1.Background

Nowadays, people have more access to a variety of digital of music pieces which are stored locally or streamed down on the internet. With a large collection of music, users can face difficulties choosing what song to listen to next and in sequence. And regular DJ software's require the user to do the mixing by himself. But if the user lacks musical knowledge it is no easy task to make a pleasing mix.

In this research, a method of mixing songs automatically like Disc-Jockeys is proposed using machine learning tools. The system will consider the similarity of songs based on the user choice to stay in the same mood. To choose the next song, three features are considered: the music genre, the tempo of a song and the music key. For this task, SVM is used for genre identification, and Markov chain and Hidden Markov Model for key identification. In the final phase, two consecutive songs are mixed from a playlist by fading out one song and fading in the other with smooth transitions.

This dissertation is organized as follow: chapter 2 introduces the music genre classification using Support Vector Machine, chapter 3 describes the music key

identification, chapter 4 explains the tempo estimation, and chapter 5 develops the procedure of mixing songs. Finally, chapter 6 concludes the thesis.

Chapter 2. Related work

Many research efforts on music information retrieval have been already published but few tried on automatic mixing. In [11], the authors proposed a method of mixing with optimal tempo adjustment with a function to measure user discomfort based on the results of a subjective experiment. Furthermore, this paper proposes a unique tempo adjustment technique called “optimal tempo adjustment”, which is robust for any combination of tempi of songs to be mixed. The adjustment of tempo is done by simple signal expansion or contraction. However, it is obvious that tempo is just one of many elements in music that affect user perception.

In another research that developed MusicMixer [12], songs are mixed using audio similarity calculated via beat analysis and latent topic analysis of the chromatic signal in the audio by employing a machine learning method called latent Dirichlet allocation (LDA). The topic model is constructed by extracting the features of songs and applying LDA to the features.

There is another approach introduced by Basu [16]. In this, they look across the two-dimensional space of parameters: the timescale adjustment necessary to align A and B, and the time offset required to align A and B. They find these alignments and combine a wider class of songs with dance music (e.g., Mozart with techno).

They do this by jointly optimizing the energy alignment of both signals instead of attempting to detect individual beats/tempos.

Other studies were concentrated on music playlist generation [13, 14, and 15]. However in these approaches, the concern of mixing songs was not taken into consideration.

Chapter 3. Music mixing techniques

3.1.Music genre classification

Musical genres are categorical labels engendered by humans to characterize variants of music. They differ from tempo, song structure, harmony, melody, rhythms to lyrics. Music genre classification can be utilized in developing automatic playlists on our music player app or storing a large collection of sorted songs online. In order to automatically organize the songs into groups, we will use the spectrograms and the MFCC (Mel-frequency Cepstral Coefficients) to extract features and SVM (Support Vector Machines) to classify the music by genre using the extracted features.

We will describe two ways of extracting features, spectrograms and MFCC. Then discuss on Support Vector Machines, the model we will use to classify the music. And talk about our experiments and evaluation of the model.

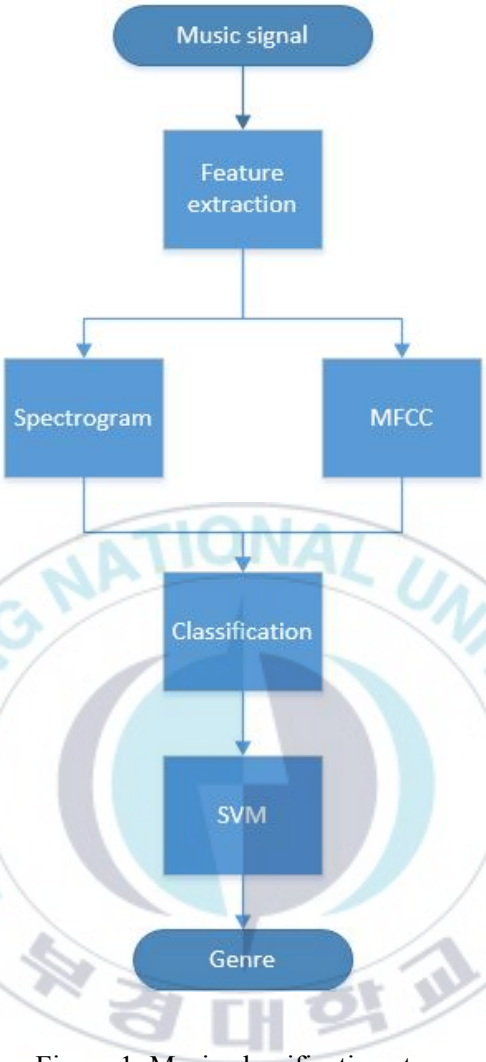


Figure 1. Music classification steps

3.1.1. Feature extraction

To make music data comparable and algorithmically employable, the first step in most of all music processing tasks is to extract useful features that capture important key aspects while removing unnecessary details. Feature extraction involves reducing the number of resources required to describe a large set of data by transforming the data in the high-dimensional space to a lower-dimensional to facilitate the learning and generalization steps.

Spectrograms

A spectrogram is a visual form of the spectrum of frequencies in a sound as they differ with time at various frequencies in a waveform. In other words, it shows the amplitude of the frequency parts of the signal that changes over time. It is represented with time and frequency. The dark zones of a spectrogram represent the peaks of a signal (Figure 2).

The spectrogram using short-time Fourier transform (STFT) is calculated by squaring magnitude of the STFT of an input signal s(t):

$$spectrogram(t, \omega) = |STFT(t, \omega)|^2$$

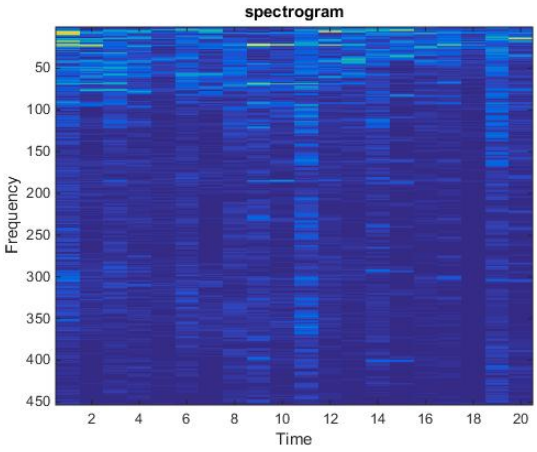


Figure 2: Spectrogram representation

MFCC

Widely used in automatic speech recognition system, Mel-frequency cepstral coefficients (MFCCs) are short-term spectral-based features. A cepstrum is an outcome of taking the Inverse Fourier transform (IFT) of the log magnitude of the measured spectrum of a signal.

MFCCs are generally created as follows:

1. Split the signal into short frames
2. Calculate the Fourier transform (FT) of a signal.
3. Map the powers of the spectrum obtained above onto the Mel.
4. Make the logs magnitude at each of the Mel-frequencies.
5. Take the discrete cosine transform (DCT) of each Mel log powers.
6. The MFCCs are the amplitudes of the obtained spectrum.

3.1.2.Classification using SVM

SVMs were basically designed for binary classifications. However, many practical problems, have more than two classes. At this end, they extended the standard SVM to a multi-class SVMs which is seen as an extension of the binary SVM classification. The concept is to break down the problem into many binary classifications and combine them to get the final classification. Two mostly used approaches are one-against-all and one-against-one methods. The one-against-all method separates each class from all others and constructs a combined classifier while the one-against-one method separates all classes' pairwise and constructs a combined classifier using voting system. In our approach, we chose to use the one-against-one method.

Also known as "pairwise", One-against-one approach consists of creating one SVM classifier for each possible pair of classes. The data point will belong to the class with the most votes. It separates all classes' pairwise and constructs a combined classifier using voting system.

3.2. Music Key identification

Music key is an essential feature in music analysis governing the entire music. However, its identification far from easy. Music key is an influential feature that can be useful for music classification. In music theory, we have 24 different keys in total that include of 12 major and 12 minor keys. Each key consist of a progression of seven different pitches. To understand more the concept of keys, let's take an example of the key of "G major", the song loop around the seven notes of the G major scale: G, A, B, C, D, E, and F♯. This means that the essential notes composing this music are all retrieved from that set of notes.

A challenging problem in music key identification is that some keys are very similar to each other and very difficult to discriminate, if not impossible. Figure 3 shows the circle of 5ths giving the relationship among the 12 tones of the chromatic scales to their corresponding keys major and minor. And the majority of the misclassifications are caused by the difficulty to distinguish the keys in the same circle. Hand-labeling the key of many songs is an extremely time-consuming and tedious task.

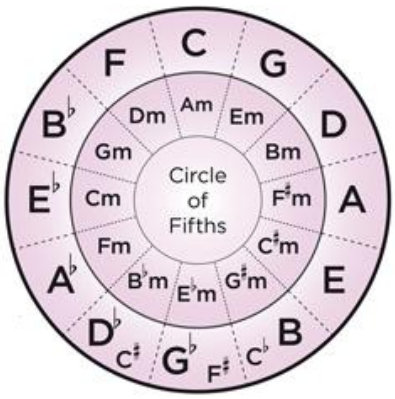


Figure 3: The circle of 5ths

To this end, we propose a method of finding the music key by extracting chroma features and classify it via Hidden Markov Model. The study has been conducted on different genres of music by different authors chosen randomly on the internet, unlike most related works which use only one genre of music. A key limitation of this research is that it's very difficult to obtain a training set which has been labeled reliably by people. We aim to show through our analysis that this model returns result is consistent with experiments evidence.The organization of the proposed approach is illustrated in Figure 4.

Similar research has been done with different techniques, we took a look on some of them. a considerable number of them are based on Krumhansl-Schmuckler key-finding model, in which the distribution of pitch classes in a piece is compared with an ideal distribution or "key profile" for each key

proposed by Krumhansl and Schmuckler in 1990 [5] [6] [8]. Arun Shenoy, Roshni Mohapatra and Ye Wang introduced another technique: a combination of Chroma based frequency analysis and music knowledge of rhythm structure and chord change patterns followed by rule-based inference [7].Our research focuses on a chroma feature analysis technique using a Hidden Markov Model to identify the key.

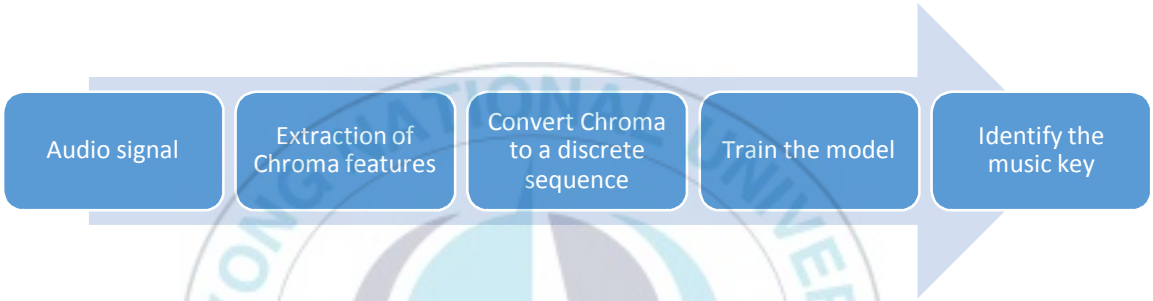


Figure 4: The architecture of the music key identification system

3.2.1. Chroma features

A chroma is an attribute of pitches and, a pitch class is a set of all pitches sharing the same chroma. In the music discipline, the term chroma relates to the twelve different pitch classes used in Western music notation: C, C♯, D, D♯, E, F, F♯, G, G♯, A, A♯, B. In other words, all notes that have the same chroma value belong to the same pitch class. For instance, the pitch class that corresponds to the chroma C consists of the set {. . . , C0, C1, C2, C3 . . .}. Chroma features represent the intensity associated with each of the 12 semitones within one octave, but all octaves are combined together. Let us consider a piano keyboard, the chroma C refers to all the C notes irrespective of the octave, high C or low C.

The major particularity of chroma features is that they capture harmonic and melodic characteristics of music while being robust to diversity in timbre and instrumentation. Every pitch that we perceive corresponds to a particular frequency f of a sinusoid in a sound signal. That's why chroma features became a major tool for processing and analyzing music data. Chroma features are derived from the energy found within a given frequency range in short-time spectral representations of audio signals extracted on a frame-by-frame basis. They segment the audio signal into narrow time intervals and take the FT of each segment. Consider a signal $f(t)$ to be analyzed at time t we compute the STFT for each window W:

$$STFT_f^u(t', u) = \int [f(t)\, W(t - t')] e^{-j2\pi ut} dt$$

The main idea of chroma features is to combine all spectral information that relates to a given pitch class into a single coefficient. In other words, it's the sum of the spectral energy overall.

$$C_f(b) = \sum_{z=0}^{z-1} |X_{lf}(b + z\beta)|$$

where X_{lf} is the log-frequency spectrum z the octave index ∈ [0, Z − 1] ,Z the number of octaves, b the pitch class (chroma) index ∈ [0, β − 1] where β is the number of bins per octave.

The chromagram is a 12-dimensional matrix which consists of a weight of the strength of the 12 different pitch classes in each of the spectrogram windows, across all octaves.

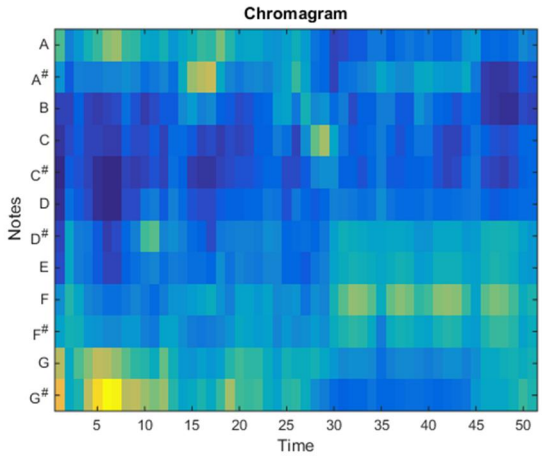


Figure 5: Chromagram representation

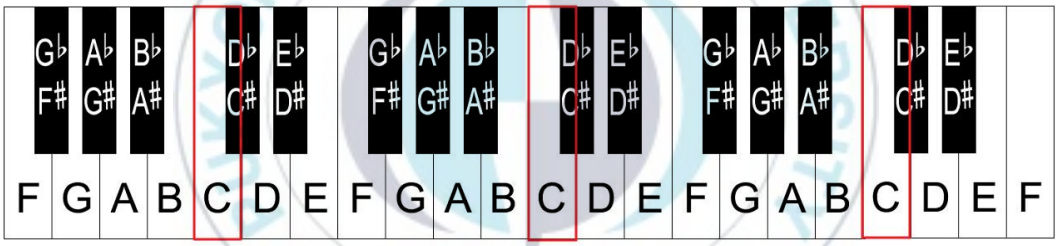


Figure 6: Chroma C on a piano

Let us consider a piano keyboard, where the keys correspond to the equal-tempered scale, the Chroma attribute will refer to the twelve pitch spelling attributes (A, A#, B, C, C#, D, D#, E, F, F#, G, G#). For example, the Chroma C refers to all the C notes irrespective of the octave (high C or low C). See Figure 5. Similarly, the Chroma C# refers to all C# notes.

3.2.2. Markov Chain

A Markov process is a stochastic process (random process) in which the probability distribution of the current state is conditionally independent of the path of past states, a characteristic called the Markov property. Markov chain is a discrete-time stochastic process with the Markov property.

To train a Markov chain, we estimate the transition probabilities.Let's consider a discrete-time $N=\{0,1,2,...\}$,States S $\{S_0,S_1,S_2,...,S_n\}$ and observed sequences q $\{q_0,q_1,q_2,...,q_n\}$,the Markov Assumption will be :

$$P(q_t = S_i | q_{t-1} = S_j, q_{t-2} = S_{k,..}) = P(q_t = S_i | q_{t-1} = S_j) = a_{ji}$$

The future behavior of the system depends only on the current state *i* and not on any of the previous states.

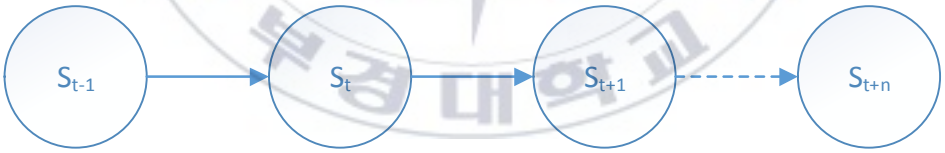


Figure 7: Markov model architecture with state S and sequence length t

3.2.3. Hidden Markov Model

Hidden Markov Model is a tool for modeling time series data. It represents the probability distribution given a sequence of observations. It also provides a way of computing the joint probability of a set of hidden and observed discrete random variables.

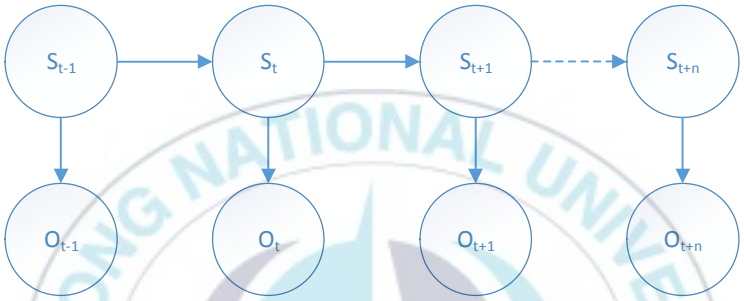


Figure 8: HMM architecture with a sequence of length t, observation O and hidden state S.

Discrete Hidden Markov Model

HMM is a popular stochastic modeling tool comprising three sets of parameters as denoted in the triple $\lambda = (A, B, \pi)$, where A is the transition matrix of A_{ij} = P (transition from state i to state j), B the emission matrix of probabilities as B_{ij} = P (emission of symbol j from node i) and as the state prior $\pi = (\pi_i)_i$. In the training phase, the model parameters are estimated to maximize the likelihood of the model given observed sequences. Baum–Welch algorithm also known as Expectation Maximization is used to find the maximum likelihood parameters [9]. Given a

sequence, we compute the posterior estimates of various hidden variables in the E-step using the forward-backward algorithm. Based on these quantities, we optimize the model parameters in the M-step. The topology of an HMM also matters for improving performance. In this research, we employed the ergodic topology for all the 24 key HMMs, where any transitions are possible.

In the recognition phase, we employ the Bayes classifier with equal class priors. Thus the classifier is simply likelihood-based, given a test sequence ϒ, we compute the log likelihood of the modelλ_k, as follows:

$$log\ P(Y|\lambda_k) = log \sum_X P(Y, X|\ \lambda_k)\ ,\ k=1,\ldots,\ 24.$$

where X represents an arbitrary Markov chain $X=X_1, X_2,.., X_T$. This can be efficiently computed using the forward algorithm on the Viterbi algorithm. Then, the model that returns the highest log-likelihood is selected as the one representing the key of the song.

$$key = arg\max_k \log P(Y|\lambda_k)$$

Continuous Hidden Markov Models

Continuous HMM is an extension of the basic HMM. It makes the HMM work with continuous observations instead of only allowing discrete sequence. In CHMM, the parameter B, the state probability of the model cannot be represented as a simple

matrix of point probabilities but rather as a complete probability density function with continuous observation. The problem with CHMM is that with a small number of mixtures, we observe a poor performance and a large number of mixtures increases computation and parameters to be estimated. Another disadvantage, CHMM needs a big amount of training set to train the model correctly

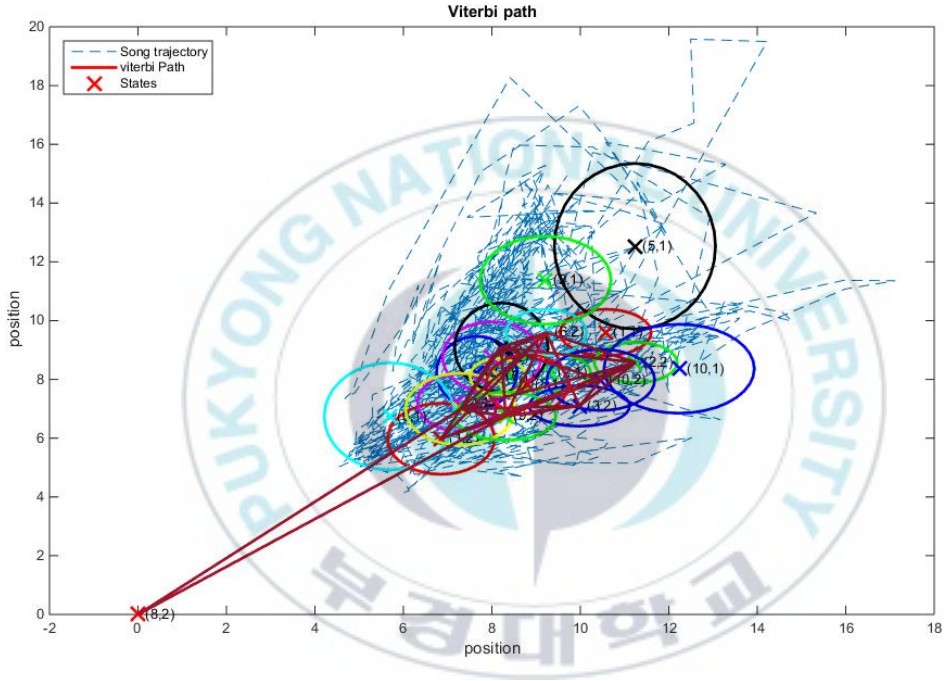


Figure 9: The broken polyline represents a song trajectory in feature space. The ellipses denote HMM state Gaussians. The central solid lines is an annotation made by a key HMM using the Viterbi algorithm. Although messy, it shows that the state Gaussians appropriately models the input signals

Table 1 gives a summary of the result. Despite the simplicity of DHMM compared to the CHMM, the former exhibited the best result. And even the simple Markov chains fared better than the CHMMs which are believed to need more data to work better. Based on the set of test results reviewed this far, we can conclude that discrete HMM is a more powerful approach to this problem. But we also mention that continuous HMM should have a potential for improvement given sufficient training set.

3.3.Tempo estimation

Tempo estimation is a fundamental task in music information retrieval (MIR). It is commonly used in music similarity, recommendation and Dj's related software. In musical terminology, the tempo is a characteristic of a song that represents the speed of a song. It shows how fast or slow the song goes. Tempo is measured by beats per minutes (BPM). For instance, a song with 120bpm means that the song has 120 beats every one minute. Identifying the beats in music audio consist of finding where a large change of sound energy occurs. A sound will be recognized as a beat if only his energy is far higher to the previous energy.

3.3.1. Implementation

In order to estimate the tempo of a song, our steps follow what is indicated in [10] conducted by Daniel P.W. Ellis. The task is divided into two parts: to convert an audio signal to an onset strength envelope and to perform an envelope autocorrelation.

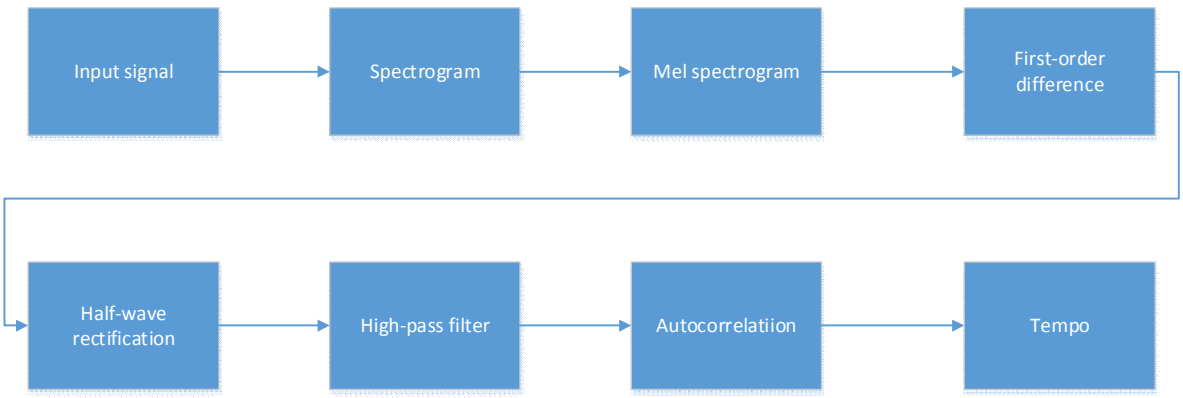


Figure 10: Tempo estimation procedure

We first calculate the short-term Fourier transform (STFT) magnitude (spectrogram) of the input audio. Then we construct a Mel-scaled power spectrogram where the frequency bands are equally separated of each one. The Mel spectrogram is later expressed in dB and we compute the first order difference along time in each band. In other words, we calculate the difference between the energy in a Mel band at time t and the energy of the same at time t-1. A positive value of the first order difference points out a rise of energy in one band. We then effectuate a half-wave rectification where every negative value is set to zero, then the remaining values are summed across all the Mel bands. The obtained signal is passed through a high-pass filter to make sharp peaks.

Given the onset envelope, we perform an autocorrelation which is the correlation of the onset strength envelope with a delayed reproduction of itself. This helps to find regular patterns or periodicity in it.

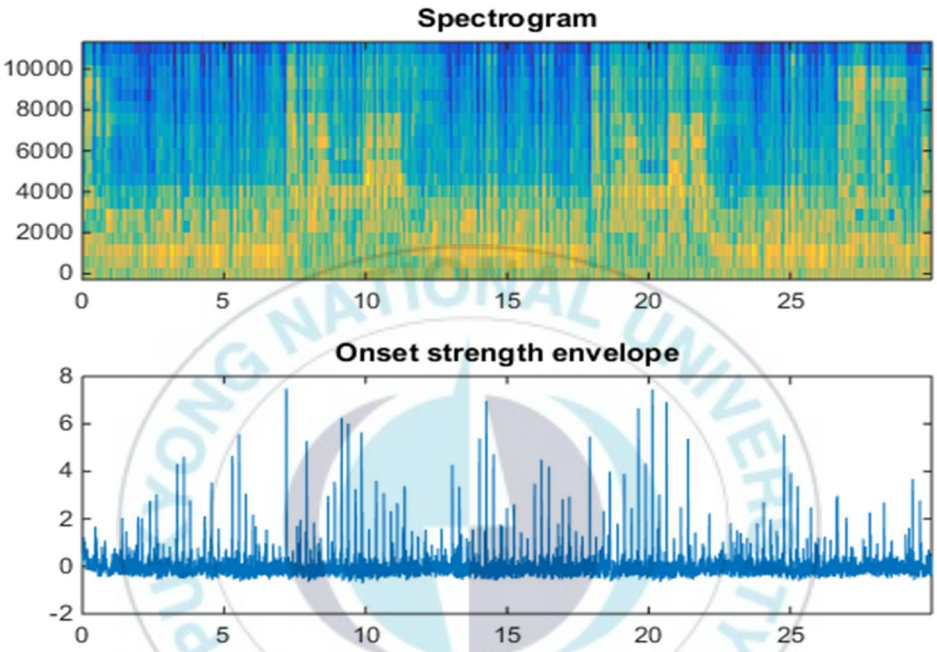


Figure 11: An example of the spectrogram and the onset strength envelope of a song

The outcome of the autocorrelation is multiple peaks. To choose the best peak, we apply a weight to the sequence of correlations to down weight periodicity peaks far from the bias towards 120BPM like human tempo viewpoint. Specifically, our tempo period strength is given by:

$$TPS(T) = W(T)\sum_{t} O(t)O(t-T)$$

where W(τ) is a Gaussian weighting function on a log-time axis:

$$W(T) = exp\left\{-\frac{1}{2}\left(\frac{\log_2 T/T_0}{\sigma_T}\right)^2\right\}$$

The tempo is simple the T with the biggest $TPS(T)$ value [10].

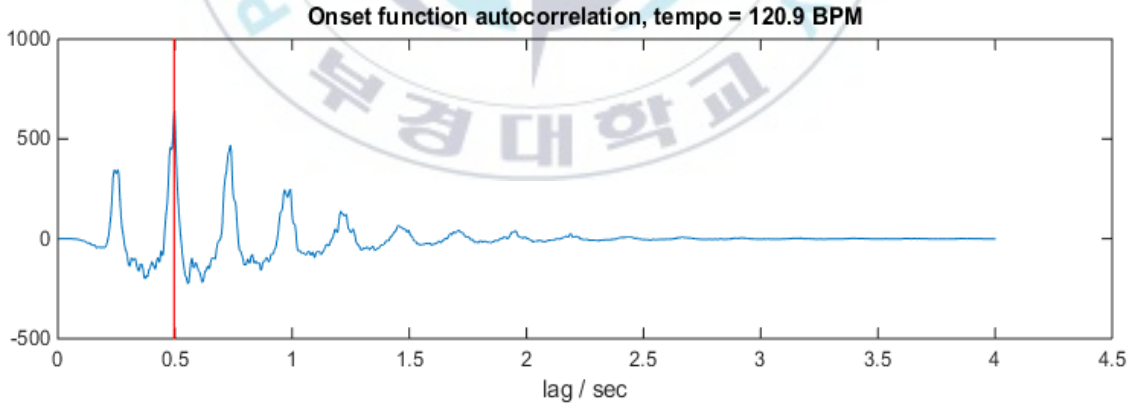


Figure 12: Autocorrelation of the onset envelope

3.4.Mixing process

In this chapter, we describe the final part of our work: the mixing process. After extracting all the features talked in the previous chapters, the music genre, the music key and the tempo of the songs, we compute the music similarity between songs and make a list of songs to play. With the list of songs obtained, we arrange them based on their tempos. Sometimes songs have a silence part in the starting point of it, so we remove it.

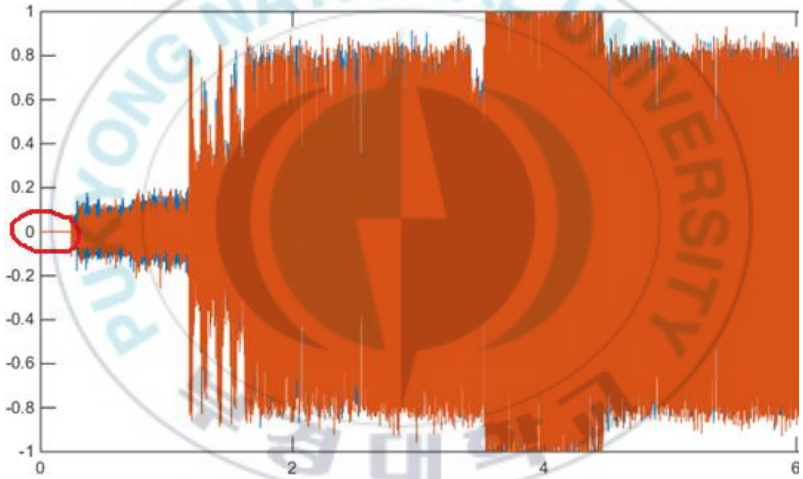


Figure 13: The red circle point to the silence part of the audio to remove

We mix songs by overlapping 15 seconds of the outro of the current song playing with 15 seconds of the intro of the next song. To create a smooth transition between the two songs, we reduce the volume gradually the current song and increase the volume of the following song. We also match the beats by increasing or reducing the tempo of the upcoming song to match it with the currently playing.

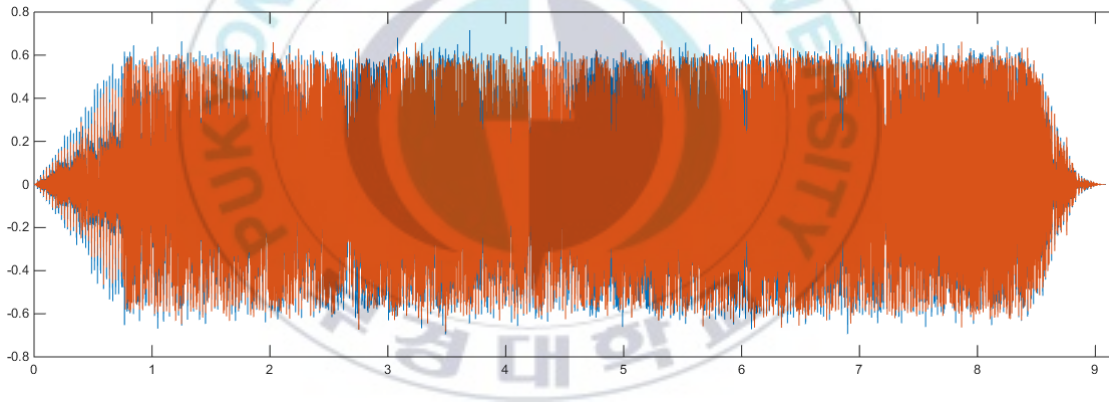


Figure 14: Song increased gradually its volume at the start and decreased at the end.

Chapter 4. Experiments

4.1.Data description

Table 1 : Data description

Experiment	Music genre classification	Music key identification
Model	SVM	Markov chain,HMM
Data quantity	100/genre	10/key
Source	GTZAN Genre Collection	random songs
File formats	wav,mp3	wav,mp3

4.2.Music genre classification

To test our music genre classification method, we used GTZAN Genre Collection [3]. Then, we chose four types of genre: blues, classical, reggae and disco. We collected 100 samples for each genre and conducted ten tests using spectrograms and other ten tests MFCC to extract the features and classified the data using SVM one-against-one approach with the quadratic kernel function. The best performance came with spectrogram with an accuracy rate of 90.6% to 78.75% for MFCC. All the work has been done using MATLAB software. Figure 15 shows the test results of the music genre classification using spectrogram and MFCC. Based on this, we can affirm that spectrogram is more reliable than MFCC for music genre classification.

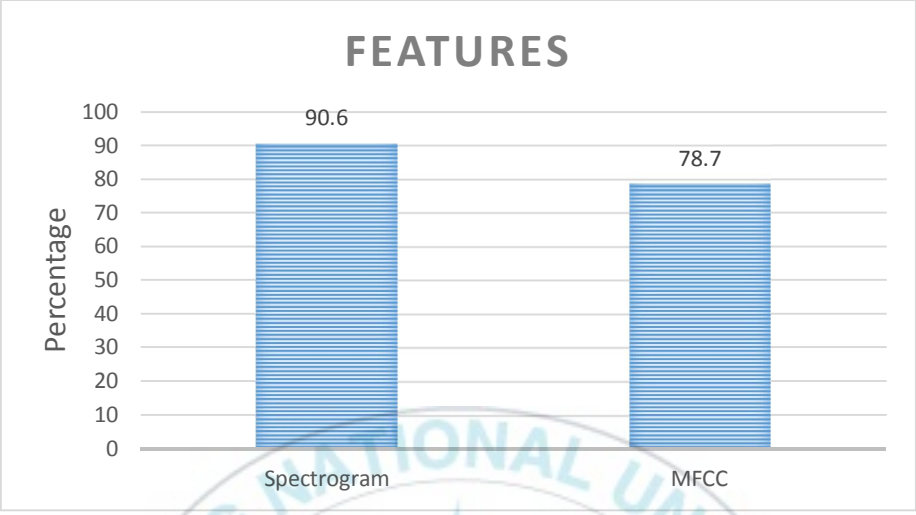


Figure 15: Accuracy results of the experiments

4.3.Music key identification with Markov chain

In our work, we used 10 songs for each key to train our model which gives 240 songs in total. Then, we extracted the Chroma features for each song and we took the position of the maximum value of each column of the chromagram and made it the sequence to train the Markov chain. With an input test sequence, we calculated the corresponding probabilities in each transition matrix and compare them to find the biggest probability which will correspond to the key.

To evaluate the Markov chain, we used 5 songs per each key which mean 120 songs. The result was 81.6% of accuracy. All the work has been done using MATLAB software.

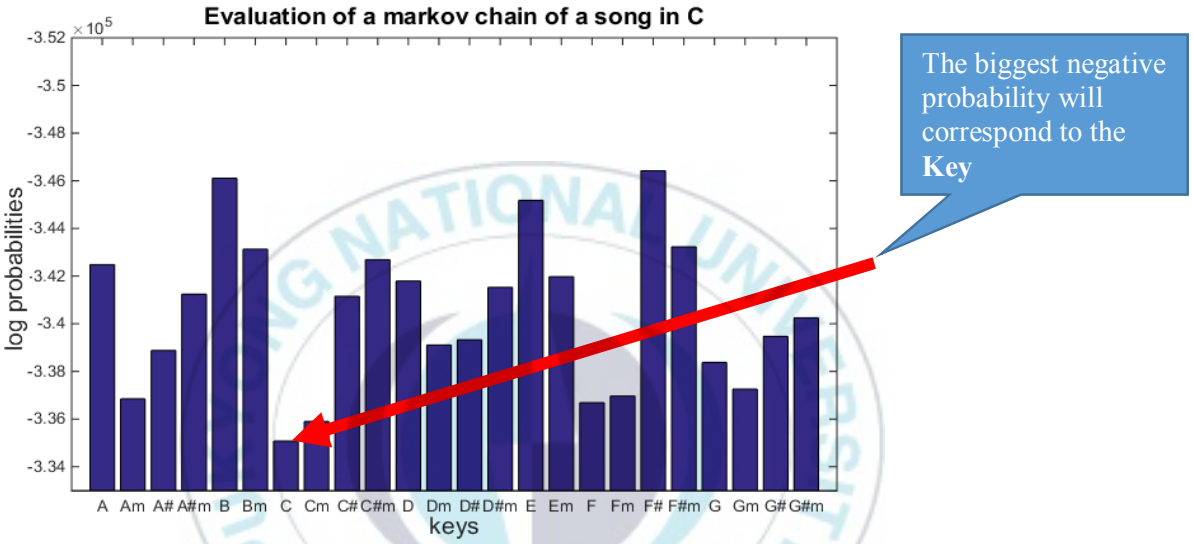


Figure 16: Evaluation of log probabilities

The below figure shows the test results of the Markov chain applied to music key detection using Chroma features.

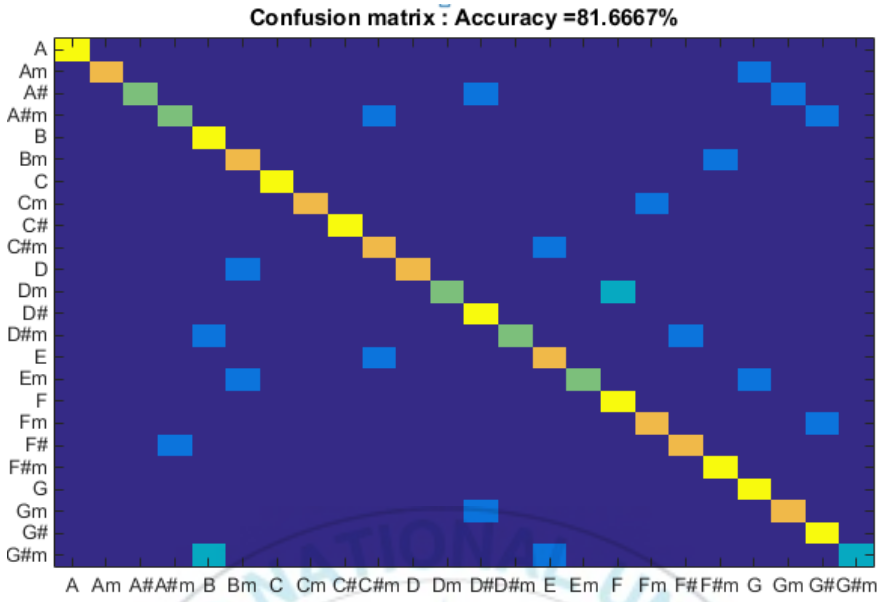


Figure 17: Accuracy results of the experiments

4.4. Music key identification with HMM

The proposed method has been implemented in MATLAB using the HMM toolbox by Kevin Murphy [10]. To train the key HMMs models, we collected training samples of 10 songs for each key and additional 5 songs for each key to test the model performance. For both training and testing, we first extract the chroma features from each song and created a sequence of indices to the maximum element in each column of the 12-dimensional chromagram. A sample result is shown in figure 5.

We have tested HMMs with a varying number of states (5, 10, 15, 20, 25, 30, 50 states), then conducted a series of tests on the models to find the optimal number of states. According to the test result as shown in Figure 19, with 25 states was found to be the best choice.

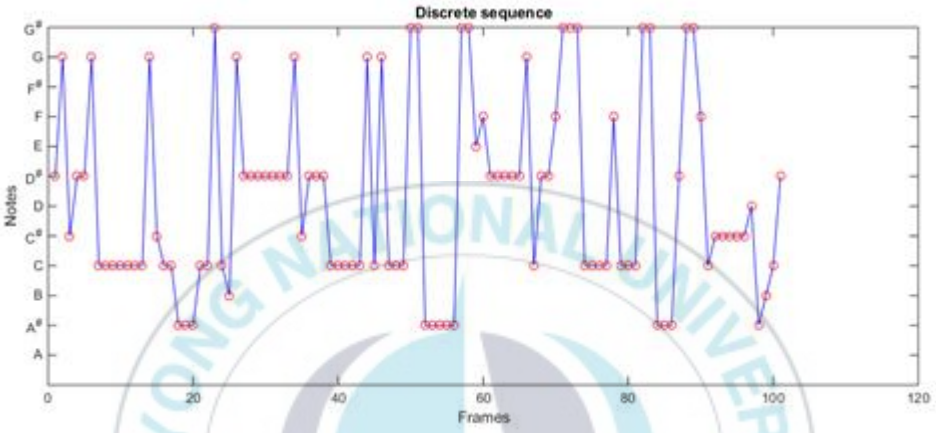


Figure 18: Discrete sequence of pitch codes derived from a current chromagram

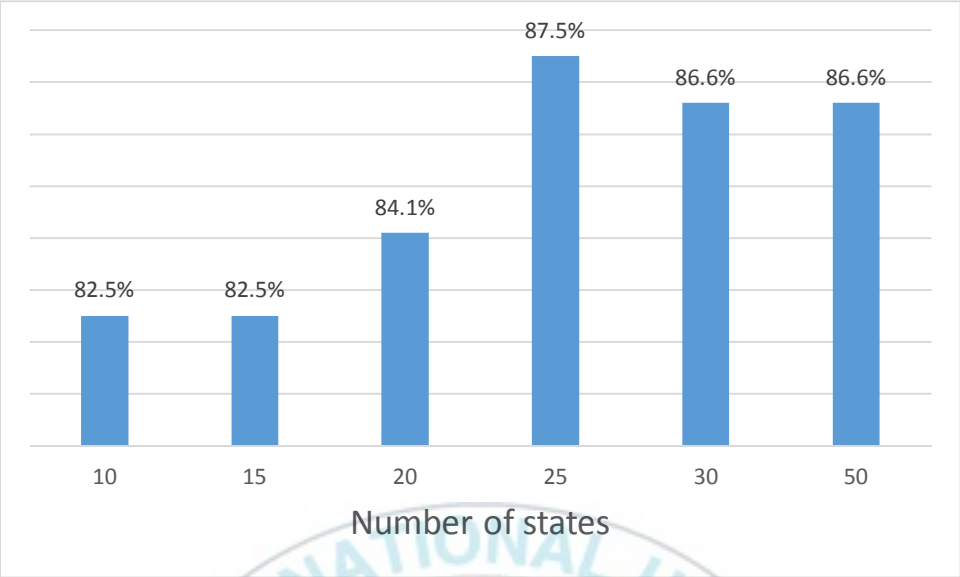


Figure 19: HMM performance with different number of states

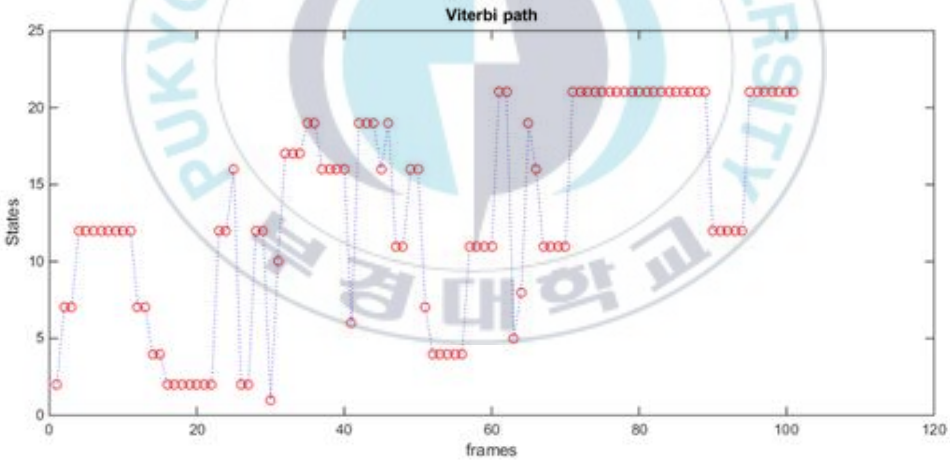


Figure 20: The Viterbi path of an HMM with 25 states given a song as represented as the sequence in Figure 18. This shows us the most probable path found using the Viterbi algorithm.

The result of this analysis shows that the proposed technique is reliable with an accuracy of 87.5% and also the method is not limited to any style of composition as we used different audio style to train and to test.

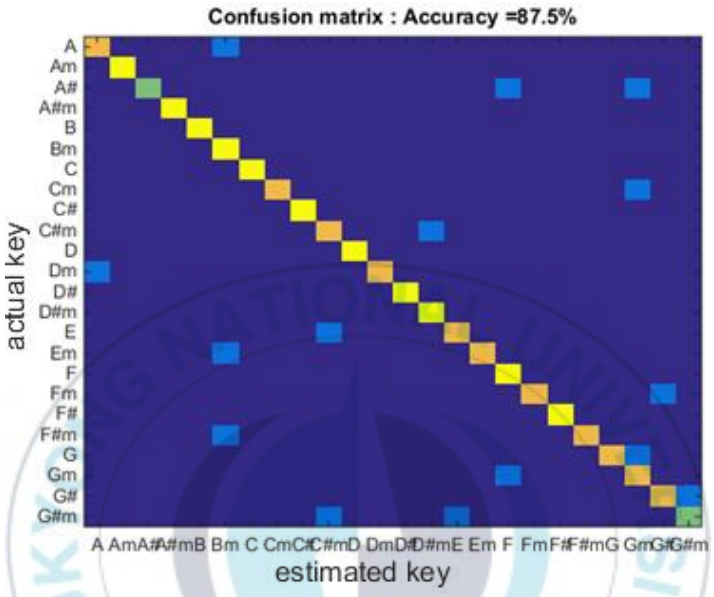


Figure 21: Confusion matrix giving the accuracy of the estimated keys for songs with known actual keys with 87.5% of songs were correctly classified.

To verify this method, we performed a test and we managed to detect the key accurately for 105 out of the 120 songs.

With the same dataset, we compared the classification performance of three different models Markov Chain, discrete and continuous HMM.

Table 2: Classification accuracy from different models

Model	Accuracy

Markov Chain	81.6%
DHMM	87.5%
CHMM	77.5%

4.5.Evaluation of the overall system

A subjective experiment were carried out to demonstrate the effectiveness of our proposed method.

The evaluation has been performed on five students with no skills of mixing songs. They listened to the music mix result then rated it from 1 to 5: poor, fair, good, very good, or excellent. Two features were considered to measure the performance of our approach: how similar the songs are and how smooth the transition is.

Table 3: Results of our experiment.

Student	Music similarity	Smooth transition
1	5	5
2	5	5
3	5	4
4	4	4
5	4	4
Average	92%	88%
Overall	90%	

Chapter 5. Conclusion

In this dissertation, we have presented a method for creating a mix automatically of multiple songs. We joined some machine learning algorithms to make a robust and simple system for assisting users to make a good mix effortlessly.

We provided a comparative study of two features extraction methods for music genre classification: spectrogram and MFCC (Mel-frequency Cepstral Coefficients). We have also discussed on Support Vector Machines, the model we used for classification. Our technique has managed to achieve 90.6% of accuracy with spectrogram for tests conducted on 80 songs divided into four genres.

We also have presented an approach for detecting music keys from a song by using HMM and Chroma features. For the experiments, the models were evaluated on test sets of 120 songs giving an accuracy of 87.5% for an HMM of 25 states.

We clarified how we estimate the tempo of a song by using the method proposed by Daniel P.W. Ellis.

Finally, we combined all the music analysis techniques and explained our mixing procedure to make a good mix.

The experimental results from this study show that the obtained results are satisfactory. We believe part of the test errors could be reduced by increasing the training sets for the music genre classification and music key identification.

Even though our system offers an easy way for creating a mix based on music similarity with a smooth transition, it cannot replace professional DJs. Human perception is always needed on the field. For example, a DJ can know which songs or singers are popular in the moment, the audience's mood so he can make a more suitable mix based on it.

For our future work, we will continue to explore the mix creation and improve our software by adding more features.

References

[1] Meinard Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*

[2] Bartsch, M. A. and Wakefield, G. H. *"To catch a chorus: Using chroma-based representations for audio thumbnailing," in Proc. Int. Workshop on Applications of Signal Processing to Audio and Acoustics, Mohonk, NY, pp. 15–19 (2001)*

[3] *GTZAN Genre Collection: http://marsyasweb.appspot.com/download/data_sets/*

[4] Lawrence R. Rabiner, *"A tutorial on hidden Markov models and selected applications in speech recognition," in Proceedings of the IEEE, February 1989, vol. 77, no. 2.*

[5] E. D. Scheirer, *"Tempo and beat analysis of acoustic musical signals," J. Acoust. Soc. Amer., vol. 103, no. 1, pp. 588–601, 1998.*

[6] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, *"An experimental comparison of audio tempo induction algorithms," IEEE Trans. Audio, Speech, Lang. Process., vol. 14, no. 5, pp. 1832–1844, Sep. 2006.*

[7] J. Zapata and E. Gómez, *"Comparative evaluation and combination of audio tempo estimation approaches," in Proc. AES Conf. Semantic Audio, Jul. 2011. [8] S. Hainsworth, "Beat tracking and musical metre analysis," in Signal Processing Methods for Music Transcription. New York, NY, USA: Springer, 2006, pp. 101–129.*

[9] S. Dixon, *"Automatic extraction of tempo and beat from expressive performances," J. New Music Res., vol. 30, no. 1, pp. 39–58, 2001. [10] S. Dixon, "Onset detection revisited," DAFx, vol. 120, pp. 133–137, 2006*

[10] Daniel P.W. Ellis, *"Beat Tracking by Dynamic Programming", LabROSA,* Columbia University, New York July 16, 2007

[11] Ishizaki, H., Hoashi, K., Takishima, Y.*: "Full-automatic DJ mixing system with optimal tempo adjustment based on measurement function of user discomfort"*. In: Proceedings of ISMIR, pp. 135–140 (2009)

[12] Hirai T., Doi H., Morishima S. (2016) "*MusicMixer: Automatic DJ System Considering Beat and Latent Topic Similarity*". In: Tian Q., Sebe N., Qi GJ., Huet B., Hong R., Liu X. (eds) MultiMedia Modeling. Lecture Notes in Computer Science, vol 9516. Springer, Cham.

[13] Platt, J. Burges, C., Swenson, S., Weare, C., Zheng, A.: "*Learning a gaussian process prior for automatically generating music playlists"*. In: Proceedings of NIPS, pp. 1425–1432 (2001)

[14] Aucouturier, J.J., Pachet, F.: *"Scaling up music playlist generation"*. In: Proceedings of ICME, pp. 105–108 (2002)

[15] Pampalk, E., Pohle, T., Widmer, G.: *"Dynamic playlist generation based on skipping behavior"*. In: Proceedings of ISMIR, pp. 634–637 (2005)

[16] Sumit Basu. 2004. *"Mixing with Mozart". In Proceedings of the International Computer Music Conference.*

Acknowledgment

First and foremost, I would like to give thanks and praise to the Almighty God for his grace and blessings throughout the entire project.

I would also like to express my deepest gratitude to my supervisor: Professor Sin Bong-Kee for guiding me throughout this project and giving me invaluable advice.

I am extremely thankful to my wonderful parents and my sisters for their love and support.

I thank all the Imedia Lab members, past and present for their kind corporation.

I also place on record my gratitude to Muheto Darcy-Dominique and Nkenyereye Lewis for helping me to get my admission in PKNU.