



## 저작자표시-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

**Thesis for the Degree of Master of Engineering**

# **Design of Music Learning Assistant Based on Music and Score Recognition**

**by**

**Ahmad Wisnu Mulyadi**

**Interdisciplinary Program of Information Systems**

**The Graduate School**

**Pukyong National University**

**February 2016**

# **Design of Music Learning Assistant Based on Music and Score Recognition**

**음악 및 악보 인식을 통한  
음악학습 보조 시스템의 설계**

**Advisor: Prof. Bong-Kee Sin**

**by  
Ahmad Wisnu Mulyadi**

**A thesis submitted in partial fulfillment of the requirements  
for the degree of**

**Master of Engineering**

**in Interdisciplinary Program of Information Systems,  
The Graduate School,  
Pukyong National University**

**February 2016**

**Design of Music Learning Assistant  
Based on Music and Score Recognition**

**A thesis  
by  
Ahmad Wisnu Mulyadi**

Approved by:

---

(Chairman) ***Man-Gon Park***

---

(Member) ***Carmadi Machbub***

---

(Member) ***Bong-Kee Sin***

**26 February 2016**

# Contents

Contents .....	<i>i</i>
List of Tables .....	<i>iii</i>
List of Figures.....	<i>iv</i>
요약 .....	<i>v</i>
Abstract.....	<i>vi</i>
 <b>Chapter 1. Introduction.....</b>	<b>1</b>
1.1. Background .....	1
1.2. Problem Statements .....	4
1.3. Related Works.....	4
1.4. Research Objective .....	5
1.5. Outline.....	5
 <b>Chapter 2. Hidden Markov Model .....</b>	<b>6</b>
2.1. Three Problems for Hidden Markov Models .....	8
2.1.1. Evaluation Problem .....	8
2.1.2. Decoding Problem.....	11
2.1.3. Training Problem.....	13
2.2. HMM Scaling.....	16
2.3. Continuous Observation Densities in HMM.....	18
 <b>Chapter 3. Proposed Method .....</b>	<b>20</b>
3.1. Overview .....	20
3.2. Music Score Feature Extraction .....	21
3.2.1. HOG Features .....	21
3.3. Music Signal Feature Extraction.....	22
3.3.1. Chroma Features .....	22
3.3.2. Onset Detection .....	25
3.4. Optical Music Recognition (OMR).....	27
3.5. SVM Classifier.....	30
3.6. HMM Training.....	31
3.7. Viterbi Decoding.....	31
 <b>Chapter 4. Experimental Result .....</b>	<b>34</b>
4.1. Music Score Recognition.....	34
4.2. Music Signal Recognition.....	37
4.2.1. Chroma Features.....	37
4.2.2. Onset Detection .....	39
4.2.3. Training .....	40
4.2.4. Decoding.....	41

<b>Chapter 5. Conclusions.....</b>	<b>43</b>
5.1. Summary .....	43
5.2. Future Directions .....	43
<b>References.....</b>	<b>44</b>
<b>Acknowledgements .....</b>	<b>48</b>



## List of Tables

Table 3.1.	Set of music notation .....	29
Table 4.1.	Music symbols training data .....	34
Table 4.2.	Music symbols recognition result .....	35



## List of Figures

Figure 1.1.	Music score-following workflow .....	2
Figure 2.1.	Forward lattice diagram .....	9
Figure 2.2.	Backward lattice diagram .....	11
Figure 2.3.	Viterbi lattice diagram .....	12
Figure 2.4.	Xi lattice diagram.....	14
Figure 3.1.	Proposed method workflow .....	20
Figure 3.2.	(a) Music symbol image and its (b) HOG features with 2 x 2 cell size, (c) 4 x 4 cell size, and (d) amplification of particular cell .....	21
Figure 3.3.	12 Semitones circle in Western music notation .....	22
Figure 3.4.	Shepard's Helix of pitch perception .....	24
Figure 3.5.	Chroma features extraction workflow .....	24
Figure 3.6.	Music events in single note of music recording (a) wavelet and its (b) envelope .....	26
Figure 3.7.	Typical architecture of an OMR processing system .....	27
Figure 3.8.	Decoding the observation buffer.....	33
Figure 4.1.	Music symbols missclassification cases .....	36
Figure 4.2.	(a) A music score and (b) the MIDI from the score.....	36
Figure 4.3.	(a) Music scales recording and (b) its spectrogram .....	37
Figure 4.4.	C Matrix.....	37
Figure 4.5.	Chromagram .....	38
Figure 4.6.	2D PCA of Chroma vectors .....	39
Figure 4.7.	(a) Music Scales Recording and its (b) onset estimation .....	40
Figure 4.8.	Chroma vectors with ellipses as states of HMM .....	40
Figure 4.9.	(a) Chromagram and (b) Viterbi decoding result of music scale performance.....	41
Figure 4.10	(a) Chromagram and (b) Viterbi decoding result of "London Bridge is Falling Down" music performance .....	42



## 음악 및 악보 인식을 통한 음악학습 보조 시스템의 설계

아흐메드 위스누 몰야디

부경대학교 대학원 정보시스템협동과정

### 요약

악기 연주를 마스터하는 과정에서 연습은 가장 중요한 단계이다. 이 단계에서, 미숙한 학습 초보자들에게는 어려움이 있을 수 있다. 예를 들어, 학습자는 음정과 박자를 정확하게 잡을 수 있도록 노력해야 하기 때문이다. 이러한 문제를 해결하기 위해 다음과 같은 특징을 갖는 음악학습 보조기를 디자인 하였다.

그 체계의 두가지 주요한 업무에는 음악 점수 인식과 음악 전사가 있다. 음악 점수 인식 업무에서는 분절된 음악 기호 이미지로부터 HOG의 특징들을 추출하게 된다. 기호들을 인식하기 위하여 이 시스템은 SVM 분류자를 만들어내고, 피아노 초보자 음악 점수를 고려하게 되면 분류자의 평균 성적의 정확도는 96.02%로 나타나게 된다.

음악 전사 작업에서 제안된 방법으로는 박자를 추적하기 위한 음악의 파형의 채도 이미지를 사용하는 것이다. 채도 이미지는 소리의 신호를 어떠한 옥타브 형태의 12 단계의 반음들로 분류하기 위하여 유용한 음악적 정보를 획득할 수 있다. 그러한 특징들의 집합을 고려해보면 HMM 모델은 Baum-Welch 방법을 사용하여 고안되었다. 그리고 안내 시스템은 실시간으로 현재 상태를 추정하기 위하여 단기간의 연속된 채도 이미지에서 Viterbi 알고리즘을 운영한다. 그 결과는 음악 점수의 측정 결과와 비교되기도 한다. 그러므로, 실시간 박자 추적과 점수 측정은 가능해지게 되며 음악적인 보조 시스템이 실현 가능해질 수 있다.

## **Design of Music Learning Assistant Based on Music and Score Recognition**

**Ahmad Wisnu Mulyadi**

**Interdisciplinary Programs of Information Systems, Graduate School  
Pukyong National University**

### **Abstract**

In the journey of mastering musical instruments, practice is the most important step. In this phase, an unskilled beginning learner might be having difficulties. For example, they struggles to play the musical notes and catch the tempo accurately. Practicing musical instruments could be more effective if there is a music learning assistant that listens and gives feedback to the learner. To solve this problem, this paper proposes a design of music learning assistant that follows music scores while listening to the performance.

There are two main tasks of the system, music score recognition and music transcription. In the music score recognition task, the proposed method extract the histogram of oriented gradients (HOG) features from segmented music symbols image. In order to recognize the symbols, the system employ Support Vector Machine (SVM) Classifier. Given beginner piano music score, the classifier average performance accuracy is 96,02 %.

In music transcription task, the proposed method uses chroma features of the waveform music to track the pitches. Chroma features captures musical info useful for classifying the audio signal into 12 semitones of any octave. Given a collection of such features, a Continuous Hidden Markov Model (HMM) has been designed using the Baum-Welch method. The guiding system runs Viterbi algorithm on short term sequence of chroma features to estimate the current note in real time. The result is compared to the reading of the music score. Realtime pitch tracking and score reading is possible and musical assistance is feasible.

# **Chapter 1**

## **Introduction**

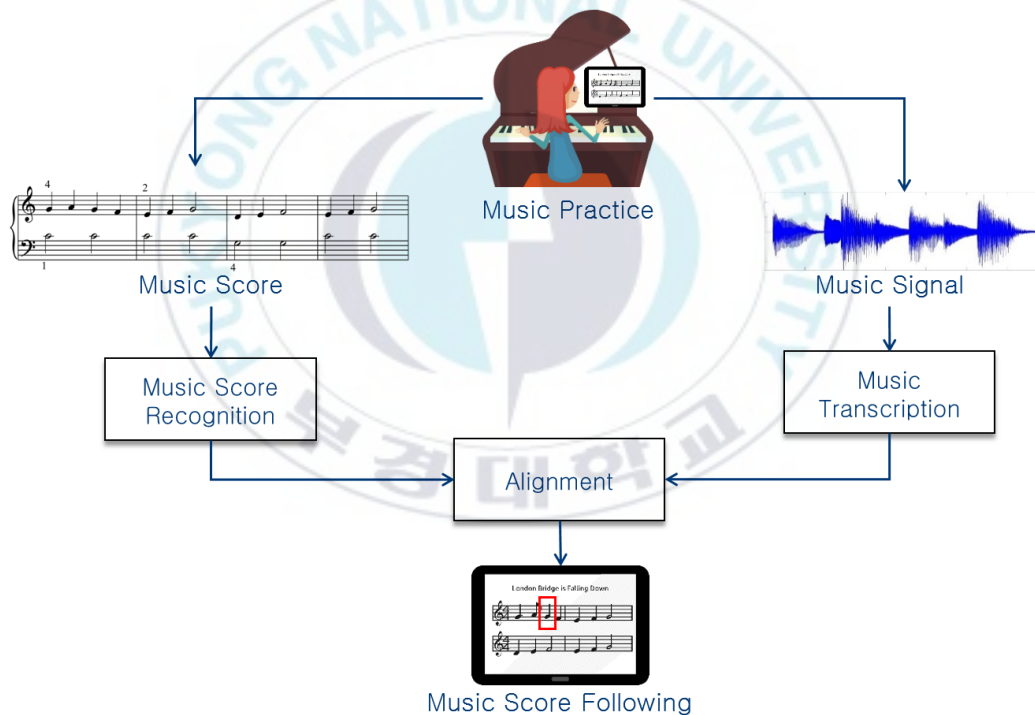
### **1.1. Background**

Learning musical instrument benefit the learner in aspects in their life. There are researches that study about relation between musical instrument learning and their achievements [10]. In their finding, they stated that the process of learning can be deeply impact by music making, with cognitive and affective components working (Raimer, 2004).

In the journey of mastering the musical instruments, practice is the most important activity. Unskilled beginning learner might be having difficulties in this phase. While looking at music scores sheets, the performer struggle to play individual notes correctly as well as keep tempo accurately. Besides that, in order to master the whole music score, the performer tends to repeat some sections in the music. Based on the difficulties faced by the beginner learner, musical instruments practice activities would be more effective if there is some music learning assistant that guide and give feedback to the performer.

Realize of this facts, developing the musical instruments learning assistant is needed. In this case, music learning assistant should have score-following feature. Score-following feature able to track performer's practice of their musical instrument and find current performance position in the music score. Score following consists of

two steps, transcription and matching. In the transcription step, audio signal of performance is analyzed into a sequence of events. In the matching step, we find the best alignment of events sequence in the transcription and its score [16]. Figure 1.1. below illustrate music learning assistant that provide score-following features. As shown in Figure 1, in music practice there are two component of interests. There are music score and music signal. So that in order to develop music learning assistant which have a score-following feature, the design consists of several steps. That are music score recognition music signal transcription, and alignment.



**Figure 1.1. Music score-following workflow**

In music score recognition process, the system means to extract music score events

using image processing. On the other side, in music signal transcription process, the system intends to analyze music signal into sequence of meaningful events such as pitches. After two processes above are performed, then matching or alignment step is required. In the matching step, we find the best alignment of events sequence in the transcription and its score. In this research in order to perform music score following, we employ audio-visual analysisist.

In visual analysisist part, given music score recognition, we tried to recognize the music symbol and generate MIDI file from it. This MIDI file will be used as ground truth in the alignment step. Using popular pattern recognition in the field of optical music recognition, we employ Support Vectorm Machine (SVM).

In audio analysisist part, in order to be able recognize the music in the music score following feature, time series model come to the rescue. One of the well known model is Hidden Markov Model (HMM). By using stochastic approach, HMM already proved in the speech recognition field so that it becoming popular also in the music recognition field [5,15,16,19].

With this motivation, in this thesis we tried to employ two popular pattern recognition method, SVM and HMM in order to design robust music score following feature in the music learning assistant.

## 1.2. Problem Statements

In this research, problems that highlighted in order to design the music learning assistant are stated as follow :

1. How to design robust music learning assistant using audio-visual analysis ?
2. In visual analysis part, how well is the performance of SVM in recognizing the music score?
3. In audio analysis part, how well is the performance of HMM in tracking the pitches given music signal?

## 1.3 Related Works

Researches about score following approach were started based on MIDI instruments that use technique based on classical approximate string matching and heuristics technique [15,22]. Using stochastic approach, then researchers develop score-following using HMM first started by Raphael which states emit the expected sound features [15,20]. Using HMM, each note in the score is modeled by a sequence of states [5,15,16].

In the context of real time or online decoding using HMM, there is implementation that employ Viterbi decoding using two buffers : L length of audio input buffer and observation buffer as introduced in [6]. Their research result show that the real-time system can be as good as the offline system.

In music score recognition part, [17,18] cover optical music recognition which consists of image pre-processing, music symbol recognition, musical notation reconstruction and final representation construction. In the music symbol recognition there are various methods published to date. One of popular method exhibiting best performance among others is SVM to classify the musical symbol [17,18].

#### **1.4. Research Objective**

Aim of this research is to design and develop music learning assistant that will guide and give feedback to the performer in musical instruments practice in real time. This is done by applying music score following feature to transcribe the music signal, music score recognition and alignment of those.

#### **1.5. Outline**

Chapter 1, this chapter cover introduction of the research. Then, we will cover Hidden Markov Modeling theory in Chapter 2. Using Hidden Markov Modeling, we try to propose method in order to develop music learning assistant as we will describe in Chapter 3. After implement the proposed method, then we analyze and presents the results in Chapter 4. Finally, Chapter 5 concludes the research, brief discussion and future directions.

## Chapter 2

### Hidden Markov Model

A system is considered described as being in one of  $N$  distinct states  $S_1, S_2, \dots, S_N$  at time  $t = 1, 2, \dots, T$ . In case of a first order Markov chain, the state transition probabilities does not depend on the whole history of the process, instead only the preceding state is taken into account [14,19]. Given time  $t$  and states  $q$ , first order markov chain can be defined as

$$P(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \dots) = P(q_t = S_j | q_{t-1} = S_i) \quad (2.1)$$

Where  $P$  is the probability and  $i, j, k$  are states indexes.

In above right hand side of above equation is independent of time, thereby leading to the state transitions probabilities  $a_{ij}$  that

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i), \quad 1 \leq i, j \leq N \quad (2.2)$$

with stochastic constraints

$$a_{ij} \geq 0 \quad \forall i, j \quad (2.3)$$

$$\sum_{j=1}^N a_{ij} = 1$$

The probability to start in a state we denote the initial state probabilities as follow :

$$\pi = P(q_1 = S_i), \quad 1 \leq i \leq N \quad (2.4)$$

That also have stochastic constraints

$$\sum_{i=1}^N \pi_i = 1 \quad (2.5)$$



Equations above is defined for the discrete-time Markov model which cannot be applicable to all problems of interest. Therefore, the Hidden Markov Model (HMM) is introduced. This extensions implies that every state will be probabilistic and not deterministic. This means that every state generates an observation at time  $t$ ,  $o_t$ , according to a probabilistic function  $b_j(o_t)$  for each state  $j$  as defined below

$$b_j(o_t) = P(o_t | q_t = S_j), 1 \leq j \leq N \quad (2.6)$$

According to [19], an HMM is characterized by :

1.  $N$  number of hidden states in the model. Although the states are hidden, for many practical applications there is often some physical significance attached to the states or to sets of states of the model. Individual states defined as  $S = \{S_1, S_2, \dots, S_N\}$  and the state at time  $t$  as  $q_t$ .
2.  $M$  number of distinct observation symbol per state. The observation symbols correspond to the physical output of the system being modeled. Individual symbols defined as  $V = \{v_1, v_2, \dots, v_M\}$ .
3.  $A = \{a_{ij}\}$  denoted as state transition probability from  $S_i$  to  $S_j$  as described in follow equation :

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i)$$

4.  $B = \{b_j(k)\}$  denoted as the observation symbol probability distribution that will emit  $o_t$  in state  $j$  as described in follow equation :

$$b_j(o_t) = P(o_t | q_t = S_j)$$

5.  $\pi = \{\pi_i\}$  denoted as the initial state probability which is probability of being  $S_i$  in the initial state as follow :

$$\pi = P(q_1 = S_i)$$

The complete model parameter notation commonly written as  $\lambda = (A, B, \pi)$ .

## 2.1. Three Problems for Hidden Markov Models

There are three basic problems that the model can be applied in real-world applications [14,19].

### ***Problem 1: Evaluation Problem***

Given a model  $\lambda$  and a sequence of observations  $O = O_1 O_2 \dots O_T$ , we compute the  $P(O | \lambda)$ .

### ***Problem 2: Decoding Problem***

Given a model  $\lambda$  and a sequence of observations  $O = O_1 O_2 \dots O_T$ , we compute the most likely or optimal state sequence  $Q = Q_1 Q_2 \dots Q_T$ .

### ***Problem 3: Training Problem***

In this problem, we attempt to optimize the model parameter  $\lambda = (A, B, \pi)$  so as to best describe how a given observation sequence comes about and to maximize  $P(O | \lambda)$ .

### 2.1.1. Evaluation Problem

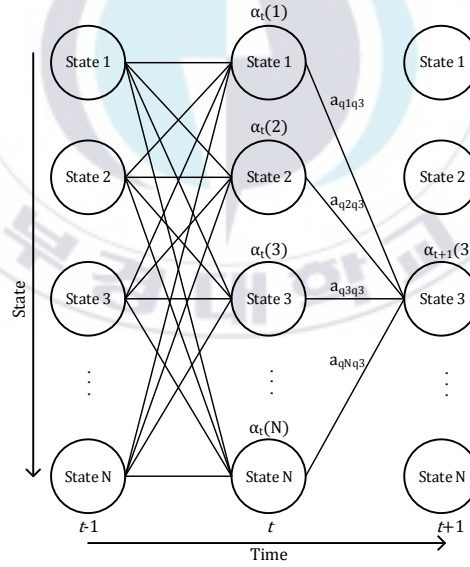
In this evaluation problem, we want to calculate  $P(\mathbf{O} | \lambda)$  which is the probability of observation  $\mathbf{O} = \mathbf{O}_1 \mathbf{O}_2 \dots \mathbf{O}_T$  given the model  $\lambda$ . To calculate  $P(\mathbf{O} | \lambda)$ , the efficient procedure is required. Such procedure exists and is called forward-backward procedure.

### Forward Algorithm

Consider the forward variable  $\alpha_t(i)$  defined as

$$\alpha_t(i) = P(\mathbf{O}_1 \mathbf{O}_2 \dots \mathbf{O}_t, q_t = S_i | \lambda) \quad (2.7)$$

Equation 2.8 describe that  $\alpha_t(i)$  is probability of partial observation sequence  $\mathbf{O}_1 \mathbf{O}_2 \dots \mathbf{O}_T$  (until time t) when being in states  $S_i$  at time t given model. Figure 2.1. shown diagram of Forward lattice.



**Figure 2.1. Forward Lattice diagram**

$P(\mathbf{O} | \lambda)$  can be calculate using Forward Algorithm as follows :

- **Initialization**

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (2.8)$$

- **Induction**

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1 \quad (2.9)$$

$$1 \leq j \leq N$$

- **Termination**

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (2.10)$$

### **Backward Algorithm**

Similar to forward algorithm, if we use backwards recursion in time then we can using backward algorithm as illustrated in Figure 2.2. Consider the backward variable  $\beta_t(i)$  defined as

$$\beta_t(i) = P(O_{t+1} O_{t+2} \dots O_T | q_t = S_i, \lambda) \quad (2.11)$$

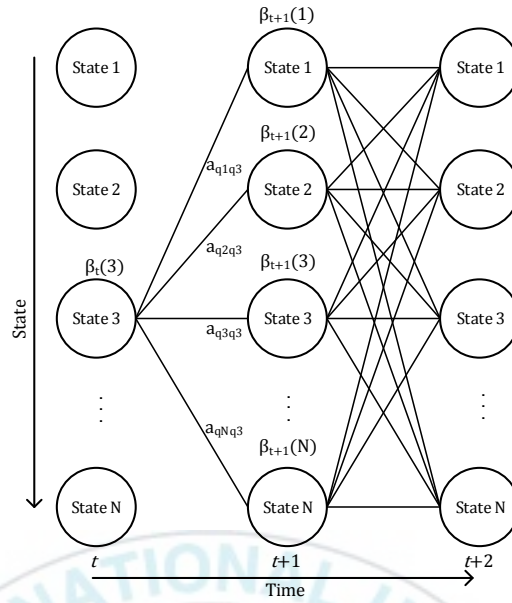
The backward algorithm is defined as follows :

- **Initialization**

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \quad (2.12)$$

- **Induction**

$$\beta_t(j) = \sum_{i=1}^N \beta_{t+1}(i) a_{ij} b_j(O_{t+1}), \quad 1 \leq j \leq N \quad (2.13)$$



**Figure 2.2. Backward Lattice diagram**

### 2.1.2. Decoding Problem

In this problem, using the model, we mean to find the single best state sequence  $\mathbf{Q} = \{q_1 q_2 \dots q_T\}$  to given observation sequence  $\mathbf{O} = \{O_1 O_2 \dots O_T\}$  as shown in Figure 2.3. The thick lines represents the path from the best state in time  $t$  to another best state in time  $t+1$ .

A formal technique for finding single best state sequence based on dynamic programming methods is exists and called the Viterbi algorithm.

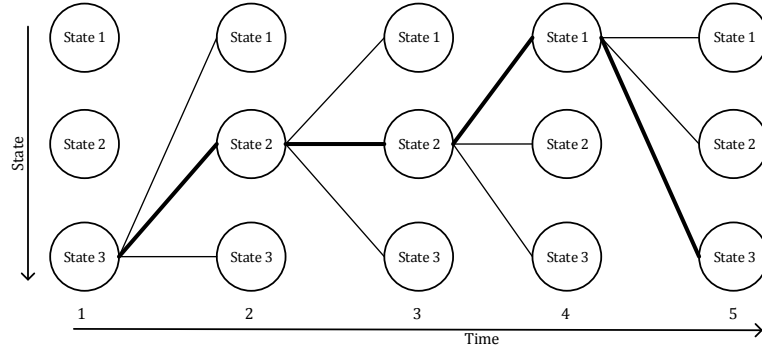


Figure 2.3. Viterbi lattice diagram

Consider

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1 q_2 \dots q_{t-1}, q_t = S_i, o_1 o_2 \dots o_t | \lambda) \quad (2.14)$$

is the maximum probability along a single path that ends in state  $S_i$  at time  $t$ , given the model  $\lambda$ . By using induction,  $\delta_{t+1}(i)$  can be define as

$$\delta_{t+1}(i) = b_j(o_{t+1}) \max_{1 \leq i \leq N} [\delta_t(i) a_{ij}] \quad (2.15)$$

To retrieve the state sequence, it is required to keep track of the argument that maximizes equation (2.16) in the variable  $\psi_t(j)$ , for each  $t$  and  $j$ .

The complete Viterbi algorithm is describe as follow :

- **Initialization**

$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N \quad (2.16)$$

$$\psi_1(i) = 0, \quad 1 \leq i \leq N \quad (2.17)$$

- **Induction**

$$\delta_t(j) = b_j(o_t) \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad 1 \leq j \leq N \quad (2.19)$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad 1 \leq j \leq N \quad (2.19)$$

- **Termination**

$$P(\mathbf{O}, q^*|\lambda) = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (2.20)$$

$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)] \quad (2.21)$$

- **Path Backtracking**

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots, 1 \quad (2.22)$$

As the result, we obtain  $q_T^*$  as an array that keep the best states sequence.

### 2.1.3. Training Problem

This third problem is the concerned with the estimation of the model  $\lambda = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$  that can be defined as

$$\lambda^* = \arg \max_{1 \leq i \leq N} [P(\mathbf{O}|\lambda)] \quad (2.23)$$

Above  $\lambda^*$  is denoted as the maximum probability of observation sequence given model. This problem is the most difficult among other problems in HMM, as there is no known way to analytically find the model that maximize the probability of the observation sequence. However, the model can be chosen to locally maximize the likelihood  $P(\mathbf{O}|\lambda)$  using an iterative procedure such as Baum–Welch method or using gradient techniques [19]. There are some advantages using Baum-Welch method as follow [14]:

- Baum-Welch is numerically stable with an increasing likelihood in every iteration

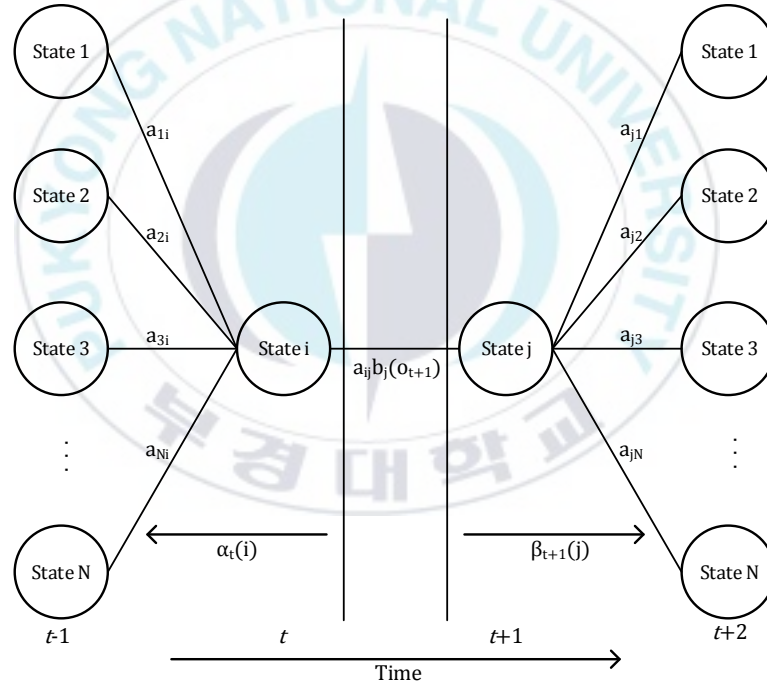
- Baum-Welch converges to a local optima
- Baum-Welch has linear convergence

Considering that facts, we will described the Baum-Welch method furthermore.

In order to describe the procedure for reestimation, we define probability of being in sate  $S_i$  at time  $t$  and state  $S_j$  at time  $t+1$ , given observation  $O$  and model, denote by  $\xi_t(i, j)$ .

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \quad (2.24)$$

For more clear description, see Figure 2.4 below.



**Figure 2.4. Xi lattice diagram**

Using forward variables, backward variables, transition probability matrix and



observation probability matrix , we can write  $\xi_t(i, j)$  as follow :

$$\xi_t(i, j) = \frac{P(q_t = S_i, q_{t+1} = S_j, O | \lambda)}{P(O | \lambda)} \quad (2.25)$$

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)} \quad (2.26)$$

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \quad (2.27)$$

The probability of being in state  $i$  at time  $t$ , given the entire observation sequence  $O$  and the model  $\lambda$  is defined as  $\gamma_t(i)$ . The relation between  $\gamma_t(i)$  and  $\xi_t(i, j)$  can be found as follow

$$\begin{aligned} \gamma_t(i) &= P(q_t = S_i | O, \lambda) \\ &= \sum_{j=1}^N P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \\ &= \sum_{j=1}^N \xi_t(i, j) \end{aligned} \quad (2.28)$$

Then we can calculate reestimation for model parameter  $\lambda = (A, B, \pi)$  :

$$\begin{aligned} \hat{\pi}_i &= \text{expected frequency (number of times) in state } S_i \text{ at time } t = 1 \\ &= \gamma_1(i) \\ &= \frac{\alpha_1(i) \beta_1(j)}{\sum_{i=1}^N \alpha_1(i)} \end{aligned} \quad (2.29)$$

$$\begin{aligned} \hat{a}_{ij} &= \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i} \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \end{aligned} \quad (2.30)$$

$$\hat{b}_j(k) = \frac{\text{expected number of transitions in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j}$$

$$= \frac{\sum_{t=1}^T \mathbf{o}_{t=V_k} \gamma_t(i)}{\sum_{t=1}^T \gamma_t(i)} \quad (2.31)$$

## 2.2. HMM Scaling

Based on previous explanation about three problems of HMM, the computation will require sequential multiplication of probabilities which have a value less than one. This means that  $\alpha_t(i)$  or  $\beta_t(i)$  tends to be zero exponentially as the time ( $t$ ) grow large. To overcome this underflow computation problem, we use scaling procedure as described in [14,19].

Consider the computation of  $\alpha_t(i)$  in forward algorithm which mentioned in Eq (2.9) and Eq (2.10) as follow :

$$\begin{aligned} \alpha_1(i) &= \pi_i b_i(\mathbf{o}_1), & 1 \leq i \leq N \\ \alpha_{t+1}(j) &= [\sum_{i=1}^N \alpha_t(i) a_{ij}] b_j(\mathbf{o}_{t+1}), & 1 \leq t \leq T-1 \end{aligned}$$

For each  $t$ , we first compute  $\alpha_t(i)$  according to induction formula and then multiply it by a scaling coefficient  $c_t$  which

$$c_t = \frac{1}{\sum_{i=1}^N \alpha_t(i)} \quad (2.32)$$

Using above scaling coefficient  $c_t$ , we can write scaled alpha  $\hat{\alpha}_t(i)$  as below

$$\hat{\alpha}_t(i) = c_t \alpha_t(i) \quad (2.33)$$

From above equation, it can be said that each  $\alpha_t(i)$  is effectively scaled by the

sum over all states of  $\alpha_t(i)$ . Forward algorithm using scaling procedure is modified as follow

- **Initialization**

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (2.34)$$

$$c_1 = \frac{1}{\sum_{i=1}^N \alpha_1(i)} \quad (2.35)$$

$$\hat{\alpha}_1(i) = c_1 \alpha_1(i), \quad 1 \leq i \leq N \quad (2.36)$$

- **Induction**

$$\tilde{\alpha}_{t+1}(j) = [\sum_{i=1}^N \hat{\alpha}_t(i) a_{ij}] b_j(O_{t+1}), \quad 1 \leq j \leq N \quad (2.37)$$

$$c_t = \frac{1}{\sum_{i=1}^N \tilde{\alpha}_t(i)} \quad (2.38)$$

$$\hat{\alpha}_t(i) = c_t \tilde{\alpha}_t(i), \quad 1 \leq i \leq N \quad (2.39)$$

- **Termination**

$$\log P(O|\lambda) = -\sum_{t=1}^T \log c_t \quad (2.40)$$

Next we need to compute the  $\beta_t(i)$  using same scale coefficients as  $\hat{\alpha}_t(i)$  for each time  $t$ . So that, scaled beta  $\hat{\beta}_t(i)$  can be written as

$$\hat{\beta}_t(i) = c_t \beta_t(i) \quad (2.41)$$

The backward algorithm using scaling coefficient is defined as follows :

- **Initialization**

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \quad (2.42)$$

$$\hat{\beta}_T(i) = c_T \beta_T(i), \quad 1 \leq i \leq N \quad (2.43)$$

- **Induction**

$$\tilde{\beta}_t(j) = \sum_{i=1}^N \hat{\beta}_{t+1}(i) a_{ij} b_j(\mathbf{o}_{t+1}), \quad 1 \leq j \leq N \quad (2.44)$$

$$\hat{\beta}_t(i) = c_T \tilde{\beta}_t(i), \quad 1 \leq i \leq N \quad (2.45)$$

### 2.3. Continuous Observation Densities in HMM

For some applications, the observations are continuous signals (or vector) including audio or music signal. Hence it would benefit to use continuous observation densities HMM as mentioned in [19].

Using continuous observation densities in HMM, we define each states as finite  $k$  mixtures of Gaussian density function  $\mathcal{N}$  as follow :

$$b_j(\mathbf{o}) = \sum_{k=1}^K c_{jk} \mathcal{N}(\mathbf{o}; \mu_{jk}, \Sigma_{jk}) \quad (2.46)$$

Where  $\mathbf{o}$  is the observation vector being modeled,  $c_{jk}$  is mixture coefficient for  $k$ th mixture in state  $j$ . Each Gaussian density for the  $k$ th mixture is defined by mean  $\mu_{jk}$  and covariance matrix  $\Sigma_{jk}$ .

The coefficient mixture satisfy the stochastic constraint :

$$\sum_{k=1}^K c_{jk} = 1, \quad 1 \leq j \leq N \quad (2.47a)$$

$$c_{jk} \geq 0, \quad 1 \leq j \leq N, 1 \leq k \leq K \quad (2.47b)$$

In relation with training problem, there will be modification in reestimation step for the mixtures as follow

$$\gamma_t(j, k) = \left[ \frac{\alpha_t(j) \beta_t(j)}{\sum_{j=1}^N \alpha_t(j) \beta_t(j)} \right] \left[ \frac{c_{jk} \mathcal{N}(\mathbf{o}_t; \mu_{jk}, \Sigma_{jk})}{\sum_{k=1}^K c_{jk} \mathcal{N}(\mathbf{o}_t; \mu_{jk}, \Sigma_{jk})} \right] \quad (2.48)$$

$$\bar{c}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j,k)}{\sum_{t=1}^T \sum_{k=1}^K \gamma_t(j,k)} \quad (2.49)$$

$$\bar{\mu}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j,k) o_t}{\sum_{t=1}^T \gamma_t(j,k)} \quad (2.50)$$

$$\bar{\Sigma}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j,k) (o_t - \mu_{jk})(o_t - \mu_{jk})'}{\sum_{t=1}^T \gamma_t(j,k)} \quad (2.51)$$

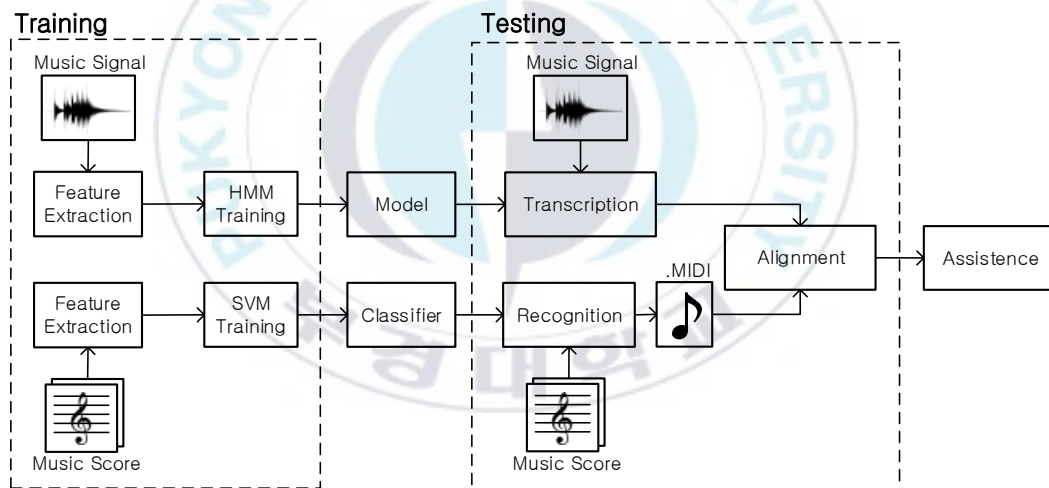


# Chapter 3

## Proposed Method

### 3.1. Overview

In order to develop music learning assistant that have music score following features, we propose design of music learning assistant which employ two popular pattern recognition. There are Hidden Markov Model (HMM) for music signal transcription and Support Vector Machine (SVM) for music score recognition as shown in Figure 3.1. belows



**Figure 3.1. Proposed method workflow**

There will be two type of features that used in this research. The first one is music signal feature extraction. In this type, given music recording, we extract pitch class profiles or chroma vectors out of it. This is done by apply pitch class profiles extraction

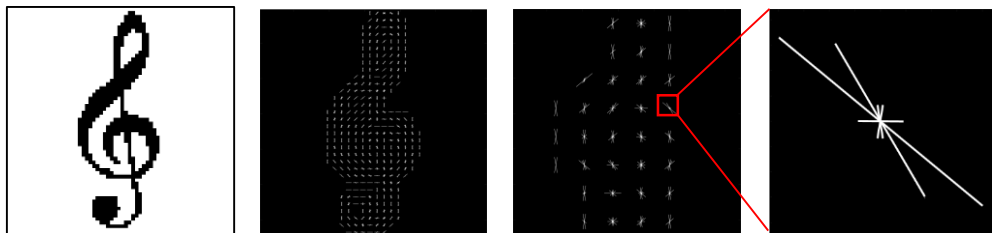
calculation described in next sub section. Then we use this features in order to train the HMM. We expect that the trained model can recognize pitch given another musical performance in the testing step.

Another features are obtained by apply image processing given music score. Here, Histogram of Oriented Gradients (HOG) features is extracted from each segmented music symbol of music score. Given HOG features of each music symbol then we train SVM classifier. The trained SVM classifier will be use to recognize another music score in the testing step.

## 3.2 Music Score Feature Extraction

### 3.2.1. HOG Features

The features to be extracted from the segmented musical symbol image is histogram of oriented gradients. The HOG is defines an occurrences counting of gradient orientation in part of an image [26]. HOG divides image into cells and computes the histogram of gradient directions or edge directions therein [26,27].



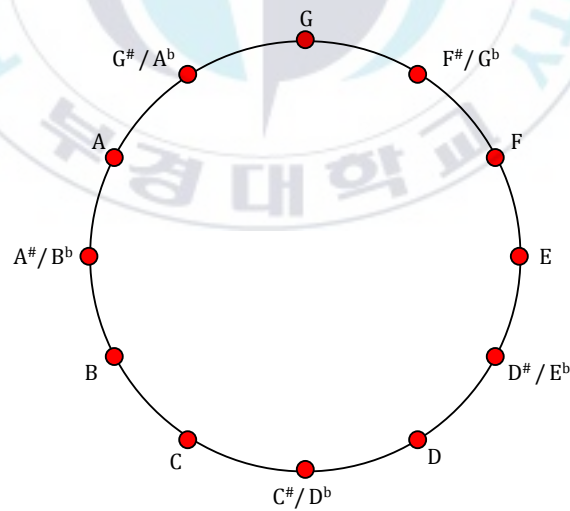
**Figure 3.2. (a) Music symbol image and its  
(b) HOG features with 2 x 2 cell size,  
(c) 4 x 4 cell size, and (d) amplification of particular cell**

Given segmented music symbol image, we extract HOG features with different size in Figure 3.2. Smaller cell size tends to have more spatial information but will increase the number of dimension, and vice versa. In this research, we use 2x2 cell size since it provides more spatial information and more accurate recognition rate. The HOG features extraction was done by applying method provided in [27].

### 3.3. Music Signal Feature Extraction

#### 3.3.1. Chroma Features

In Western music notation, the 12 pitches attributes are given by the set  $\{C, C^\sharp, D, \dots, B\}$  which repeat in the same sequence in the next octave [12,24] as shown in Figure 3.3. In Figure 3.3., the distance between two adjacent notes is known as a halfstep. The distance we perceive as a halfstep is always the same.



**Figure 3.3. 12 Semitones circle in Western music notation**



For instance, the interval between A and B<sup>b</sup> sounds like the same distance as the interval between E and F. Because of the way we perceive octaves in an exponentially doubling way, the frequency interval between halfsteps is not constant. Given a note that corresponds to frequency  $f_{min}$ , number of halfstep in octave  $b$ , the note  $k$  halfsteps above this note is at a frequency  $f_k$  can be calculate by equation

$$f_k = f_{min} 2^{m/b} \quad (3.1)$$

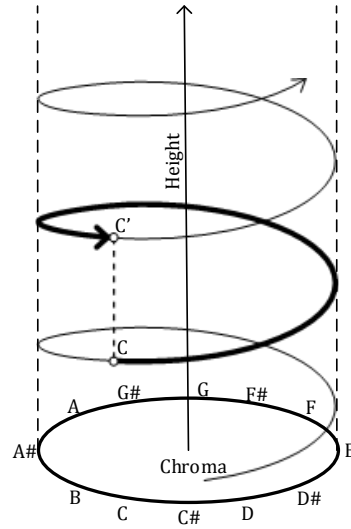
Using discrete Fourier transform then we define also frequency  $f_k$  as follow

$$f_k = \frac{k}{N} f_s \quad (3.2)$$

Given frequency  $f_k$ , number of halfstep in octave  $b$  the desired bin number can be calculated from the frequency using equation 3.3 [4]. Equation above come in handy when extract chroma features using log-frequency bin.

$$k = b \log_2\left(\frac{f_k}{f_{min}}\right) \quad (3.3)$$

In relation with octaves, two pitces can be perceived by human as similiar in “color” if they differ by one or several octaves as mentioned in [12]. Introduced by Shepard, human auditory system’s perception of pitch was better represented as a helix than as a one-dimensional line as shown in Figure 3.4. pitch helix. The vertical dimension is *tone height* while angular dimension is *chroma*. As the pitch of musical note increases from C to C’ in upper octave, its locus moves along the helix. It will rotating chromatically through all of the pitch classes before it returns to the initial pitch class C’ above one octave or cycle from the starting point C (shown as thick lines in Figure 3.4.).



**Figure 3.4. Shepard's Helix of pitch perception**

Pitch Class Profiles or also commonly known as Chroma features (or Chromagram if it already represented visually), is representation of music audio that the entire spectrum is projected onto 12 bins that is 12 distinct semitones of chromatic scale [6,7,22]. In [23], chroma features calculation is described as shown in Figure 3.5.



**Figure 3.5. Chroma features extraction workflow**

The most popular tool for describing the time-varying energy across different frequency bands is the Short-Time Fourier Transform (STFT) [11]. STFT can be visualized its magnitude that well known as spectrogram.

Given music recording  $\mathbf{X}[n]$  can be converted to a STFT representation using

$$X_{STFT}[k, n] = \sum_{m=0}^{N-1} x[n - m] \cdot w[m] \cdot e^{-j2\pi km/N} \quad (3.4)$$

where  $k$  index the frequency axis with  $0 \leq k \leq N - 1$ ,  $n$  is the short-time window center, and  $w[m]$  is an  $N$ -point Hanning window. Frequency to pitch mapping is achieved using the logarithmic characteristic of the equal temperament scale.

Where  $b$  is the number of bins per octave,  $f_{min}$  is the reference frequency,  $f_s$  is the sampling rate, then STFT bins  $k$  are mapped to PCP bins  $p$  derived from equation 3.2, 3.3 as below

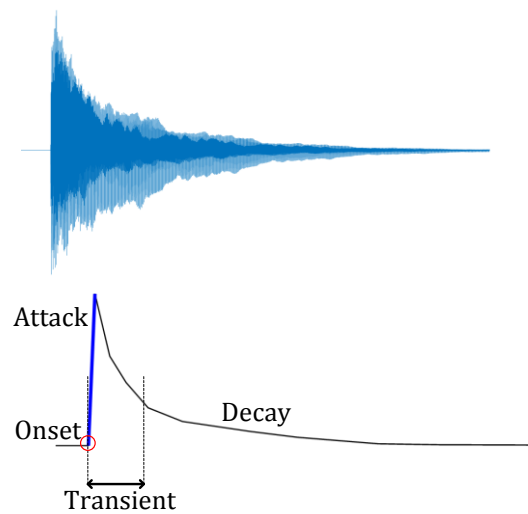
$$p(k) = \left\lfloor b \log_2 \left( \frac{k}{N} \cdot \frac{f_s}{f_{min}} \right) \right\rfloor \bmod b \quad (3.5)$$

For each time slice, we calculate the value of each PCP element by summing the magnitude of all frequency bins that correspond to a particular pitch class i.e. for  $p = 0, 1, \dots, b - 1$  using follow folding equation

$$PCP[p] = \sum_{k:p(k)=p} |X[k]|^2 \quad (3.6)$$

### 3.3.2. Onset Detection

Events detection in musical audio signal or recording needs clear distinctions since its different applications have its different needs. Relate to these events in musical signals, there are concepts of *transients*, *onsets* and *attacks* [2] [9] as shown in Figure 3.6.



**Figure 3.6. Music events in single note (a) wavelet and its (b) envelope**

According to [2] music events in above illustrations, we can distinguish each concepts as follow :

- *Attack*

Attack of the note is time interval during which the amplitude envelope increases, shown as blue lines in Figure 3.6.(b).

- *Transients*

Transients are short intervals during which the signal evolves quickly in some nontrivial or relatively unpredictable way. Release or offset of a sustained sound can also be considered as a transient period.

- *Onset*

The onset of the note is a single instant chosen to mark the temporally extended transient. In most cases, it will coincide with the start of the transient, shown as red circle in Figure 3.6.(b).

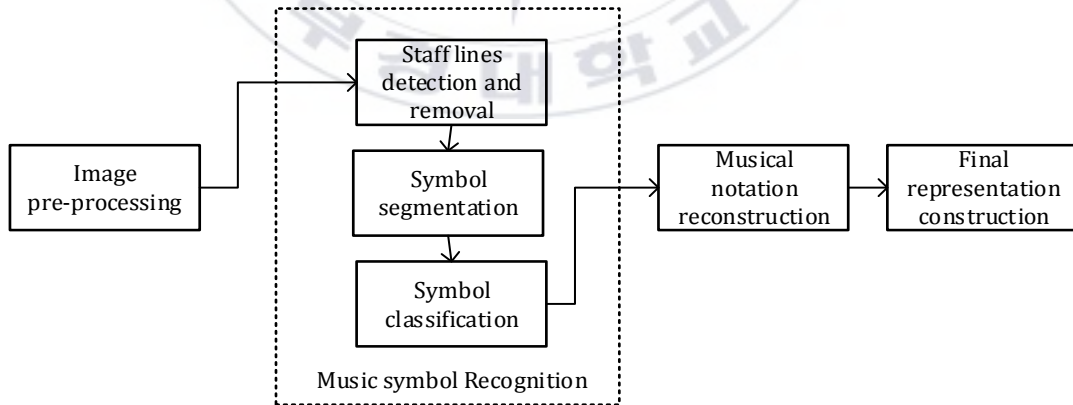
In this design of music score following, we concern about onset detection. Onset can be defined as the start of a signal event or a very moment where the start of an abrupt change in amplitude in a signal occurs [9]. Spectral difference or spectral flux is choosed as detection function since its ability to detect a change in pitch as well as a change ini energy that would be the best first estimate of any onset detection function for general purpose beat detection [9].

Spectral difference can be calculated by using following equation :

$$SD(n) = \frac{1}{N} \sum_{k=\frac{N}{2}}^{\frac{N}{2}-1} \{H(|X_k(n)| - |X_k(n-1)|)\}^2 \quad (3.7)$$

Where  $H(x)$  is zero for negative arguments and equal tot the resultfor positive arguments. This is calculated by  $H(x) = (x + |x|)/2$  in order to emphasize an increase in spectral content and is intended to emphasize onsets rather than offsets [9] [2].

### 3.4. Optical Music Recognition (OMR)



**Figure 3.7. Typical architecture of an OMR processing system [17]**

OMR processing system can be divided into three principal modules as shown in

Figure 3.7. In above illustration an OMR processing system that explained by [17] is consists of

- ***Image pre-processing***

In this models, given image of music sheet, there are several techniquese e.g. binarization, noise removal, blurring, deskewing, etc to make the recognition process more robust and efficient.

- ***Music symbol recognition***

There are three stages in this modules. Staff line detection and removal is performed in order to obtain an image containing only the musical symbols. In this part, staff lines spacing and thickness also provide the basic scale for relative size comparisons. After the removal of staff line, then the system can perform symbol primitives segmentation and classification.

- ***Musical notation reconstruction***

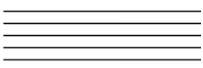

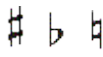




In this step, symbols primitives are merged in order to form musical symbols. Graphical and syntactic rules are used to introduce context information to validate and solve ambiguities from the previous module of music symbol recognition. Detected symbols are interpreted and assigned a musical meaning.


- **Final representation construction**

In this final step, a format of musical description is created with the information produced previously.

In relation of music symbol recognition, [17] describe set of music notation as presented in Table 3.1. below

**Table 3.1. Set of music notation [17] [25]**

Symbols	Description
	<b>Staff</b> An arrangement of parallel lines, together with the spaces between them
 treble    alto    bass	<b>Clef</b> The first symbols that appear at the beginning of every music staff and tell us which note is found on each line or space
	<b>Sharp, Flat and Natural</b> The signs that are placed before the note to designate changes in sounding pitch
	<b>Beams</b> Used to connect notes in notegroups; demonstrate the metrical and the rhythmic divisions
	<b>Accent and Staccatissimo</b> Symbols for special or exaggerated stress upon any beat, or portion of a beat
	<b>Crochet, Quaver and Minim</b> The Crochet (closed notehead) and Minim (open notehead) symbols indicate a pitch and the relative time duration of the musical sound. Flags (Quaver) are employed to indicate the relative time values of the notes with closed noteheads
	<b>Quarter, Eighth, Sixteenth and thirty-second rests</b> Indicate the exact duration of silence in the music; each note value has its corresponding rest sign; the written

	position of a rest between two barlines is determined by its location in the meter
	<p><b><i>Ties and Slurs</i></b></p> <p>Ties are a notational device used to prolong the time value of a written note into the following beat. The tie appears to be identical to slur, however, while tie almost touches the notehead centre, the slur is set somewhat above or below the notehead. Ties are normally employed to join the time value of two notes of identical pitch; Slurs affect note-group note-groups as entities indicating that the two notes are to be played in one physical stroke, without a break between them</p>

### 3.5. SVM Classifier

SVM is classification method that constructs hyper-plane in high order space which can be used as classification plane [26]. SVM is kernel-based classifier. And popular kernels are are linear, polynomial, RBF and sigmoid kernels [26].

SVM classified based on kernel. Among others, popular kernels are linear, polynomial, Radial Basis Function (RBF) and sigmoid kernels [9]. Given features  $x$ , kernel  $K$  can be defined as dot product of features  $\phi(x_a)$  with other  $\phi(x_b)$  [30], thus :

$$K(x_a, x_b) = \phi(x_a) \cdot \phi(x_b)$$

Using above kernel function, the classifier function  $f(x)$  can be notated as :

$$f(x) = \text{sgn}(\sum_i \alpha_i y_i (K(x_i, x)) + b)$$

with  $\alpha_i$  is the vector of  $l$  non-negative Lagrange multipliers to be determined,  $y_i$  are



values of support vector and  $b$  as bias.

In our case, we employ SVM classifier to recognize music symbols. There are number of musical symbol classes such as accidental, bar, braces, clef, digits, dot, note and rest. Given a set of extracted HOG features for the music symbols as training data, we train SVM classifier using a toolbox provided by MatLab [27].

### **3.6. HMM Training**

In order to do pitch recognition in transcription step, given music scales recording, we employ a continuous HMM that have 13 states, 12 states for notes events or pitches and 1 state for silences. Then we trained the HMM using Baum-Welch iterative algorithm as each iteration will guaranteed to increase the likelihood. K-Means clustering method is employed as initialization of Baum-Welch training.

### **3.7. Viterbi Decoding**

After the HMM has been trained using Baum-Welch algorithm, then we employ Viterbi algorithm as online decoding algorithm to do pitch recognition. Viterbi algorithm is the most popular technique for finding the optimal path along an HMM [15]. Viterbi algorithm purpose is to find the single best state sequence that most likely produced the observations [5,19].

Using trained HMM parameter  $\lambda$  the system tried to find the single most likely pitches sequence  $\mathbf{q} = (q_1, q_2, \dots, q_T)$  given chroma features as observation  $\mathbf{O} = (o_1, o_2, \dots, o_t)$ . In this case we defines

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1 q_2 \dots q_t = S_i, o_1 o_2 \dots o_t | \lambda)$$

as the highest probability along a single path that ends in state  $S_i$  at time  $t$  given model parameter  $\lambda$  [19]. By using induction, we can calculate the probability at time  $t + 1$  as

$$\delta_{t+1}(i) = [\max \delta_t(i) a_{ij}] \cdot b_j(O_{t+1})$$

Figure 3.8. illustrate the workflow of online decoding using modified viterbi decoder. In term of online decoding system there will be no access to future information also to the entire signal. To be able to handle this situation, we will use observation buffer that keep L frames of the chroma features. Then the decoding algorithm will decode the buffer as illustrate in figure In the decoding step for each buffer decoding  $\delta_t(i)$  will be reuse previous buffer calculations  $\delta_{t-1}(i)$  (except for the first buffer that will be calculated as initialization  $\delta_1(i)$ ) that called as modified Viterbi decodier in [6]. Regarding the decoding result of each observation in the buffer we employ voting system so that each buffer has only one decoding result which is the pitch

estimation.

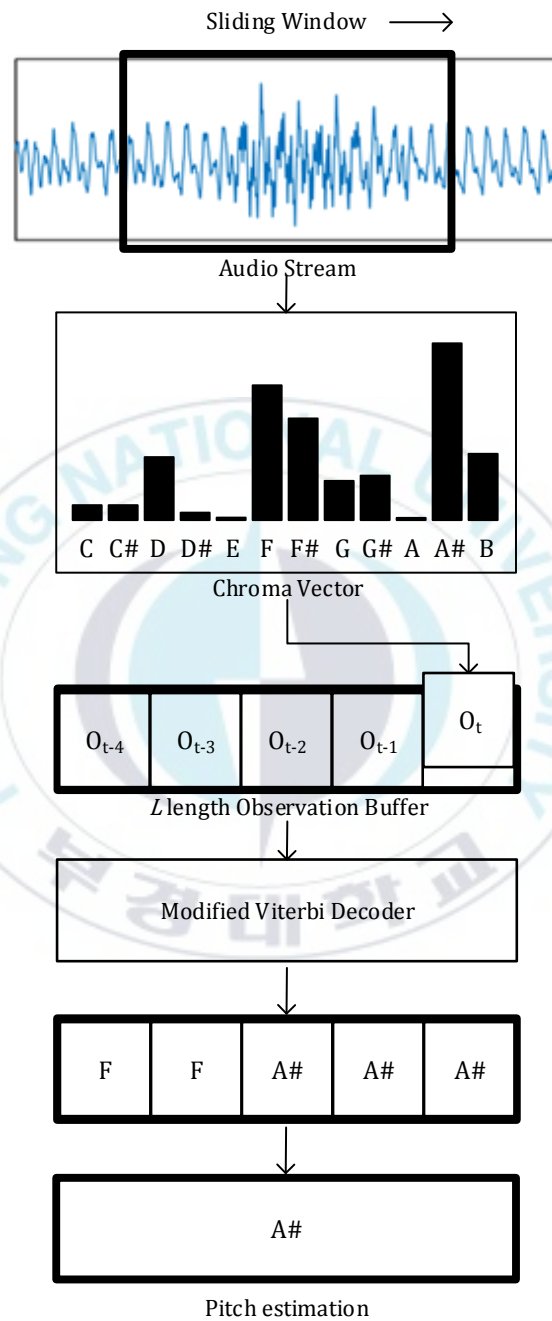


Figure 3.8. Viterbi Decoding the workflow [6]

## Chapter 4

### Experimental Result

#### 4.1. Music Score Recognition

In order to recognize the music symbols, we use music symbols training data described in Table 4.1. Music symbols in the training data are obtained from beginner piano music score which are already processed through line staff removal, gap stitching and segmentation. The system then extract HOG features out of it and use them as training data for SVM classifier training.

Table 4.1. Music symbols training data

Classes	Sub Classes	Number of Music Symbol
Accidental	Flat	61
	Natural	15
	Sharp	52
Bar	-	38
Brace	-	52
Clef	Clef F	36
	Clef G	58
Digit	Digit 0	16
	Digit 1	34
	Digit 2	23
	Digit 3	15
	Digit 4	26
	Digit 5	14
	Digit 6	16

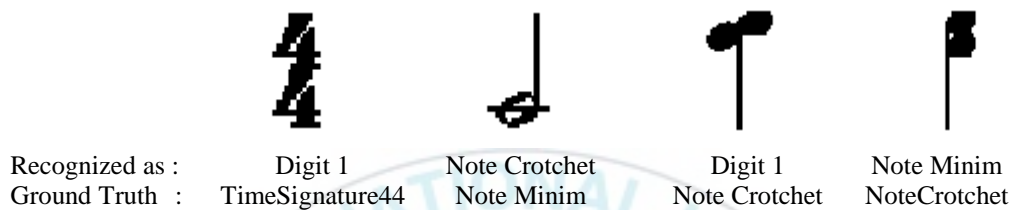
	Digit 7	12
	Digit 8	42
	Digit 9	9
Dot	-	71
Dynamic	Dynamic MF	3
	Dynamic MP	2
	Dynamic P	1
Note	Crotchet	109
	Crotchet Reverse	284
	Minim	28
	Minim Reverse	72
	Quaver	38
	Quaver Reverse	23
	Semibreve	66
Rest	Crotchet	26
	Minim	52
	Quaver	17
	Semiquaver	3
Timesignature	Timesignature 34	7
	Timesignature 44	10

Given HOG features, the trained SVM classifier is used to recognize the music symbols given music score. The result of recognition shown in Table 4.2. below :

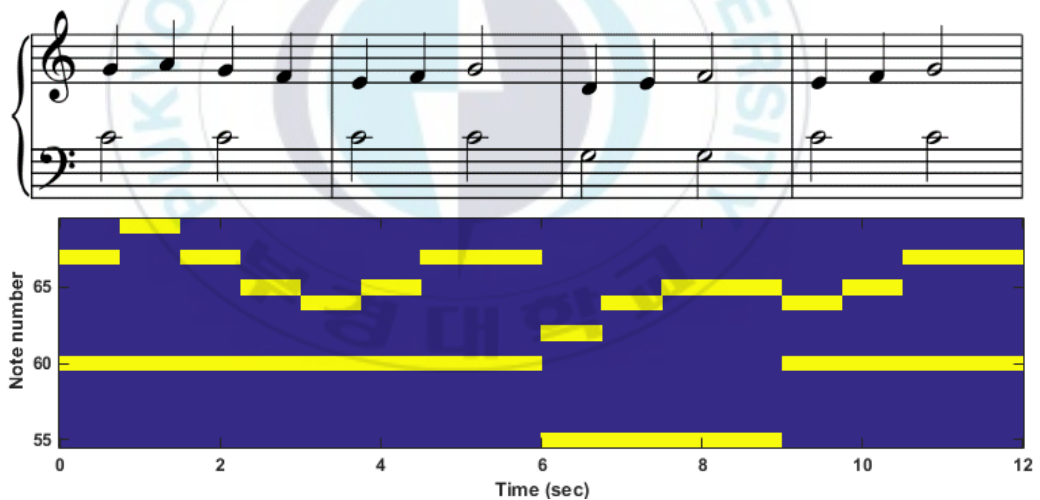
**Table 4.2. Music symbols recognition result**

Music Score Name	Number of Symbol	Correctly Recognized	Accuracy (%)
London Bridge is Falling Down	64	64	100
Twinkle Twinkle Little Stars	90	84	93.33
Peter Peter Piano	76	76	100
Au Clair De la Lune	99	98	98.99
Mary Had a Little Lamb	90	79	87.78

Given simple music score images, system able to perform good recognition with accuracy more than 85% with average accuracy 96,02%. Albeit, There are some misclassification cases that SVM fail to recognize the music symbols correctly as shown in Figure 4.1. below :



**Figure 4.1. Music symbols missclassification cases**



**Figure 4.2. (a) A music score and (b) the MIDI from the score**

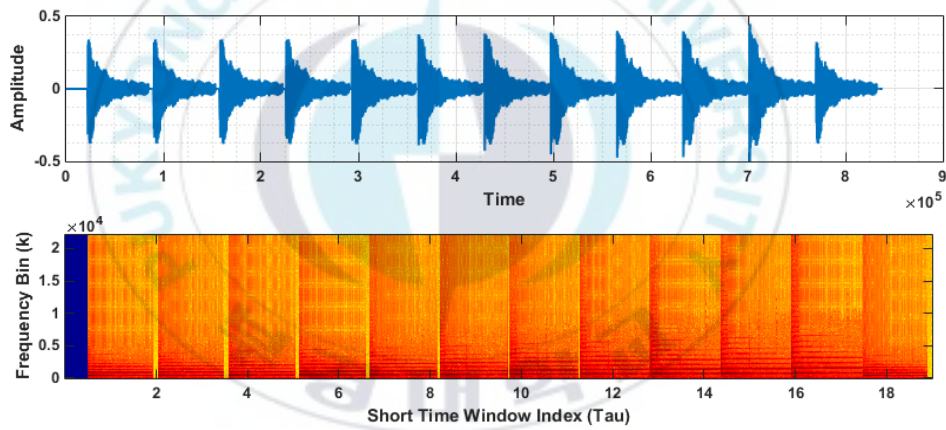
Given a simple music score in Figure 4.2.(a), system recognize all notes correctly. Then, we define construct reconstruction matrix and convert it into a MIDI file using

method provided in [12] as shown Figure 4.2.(b).

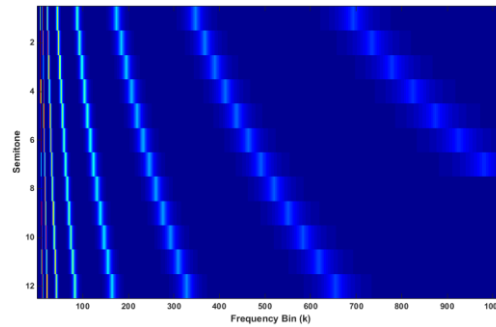
## 4.2. Music Signal Recognition

### 4.2.1. Chroma Features

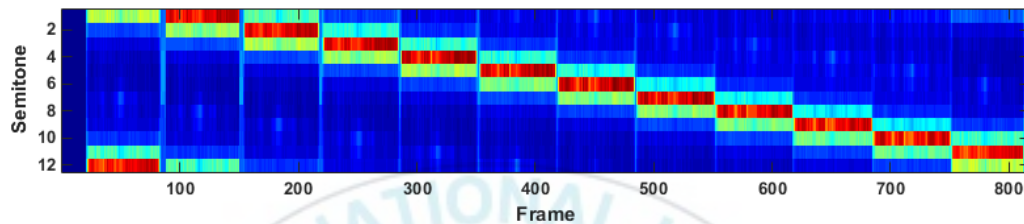
By using pitch class profiles extraction defined in 3.2. Music Feature Extraction before, we can obtain chromagram. Consider a music scales recoding in Figure 4.3.(a) below, we apply 2048 window size STFT calculation to make spectrogram in Figure 4.3.(b).



**Figure 4.3. (a) Music scales recording and (b) its spectrogram**



**Figure 4.4. C Matrix**

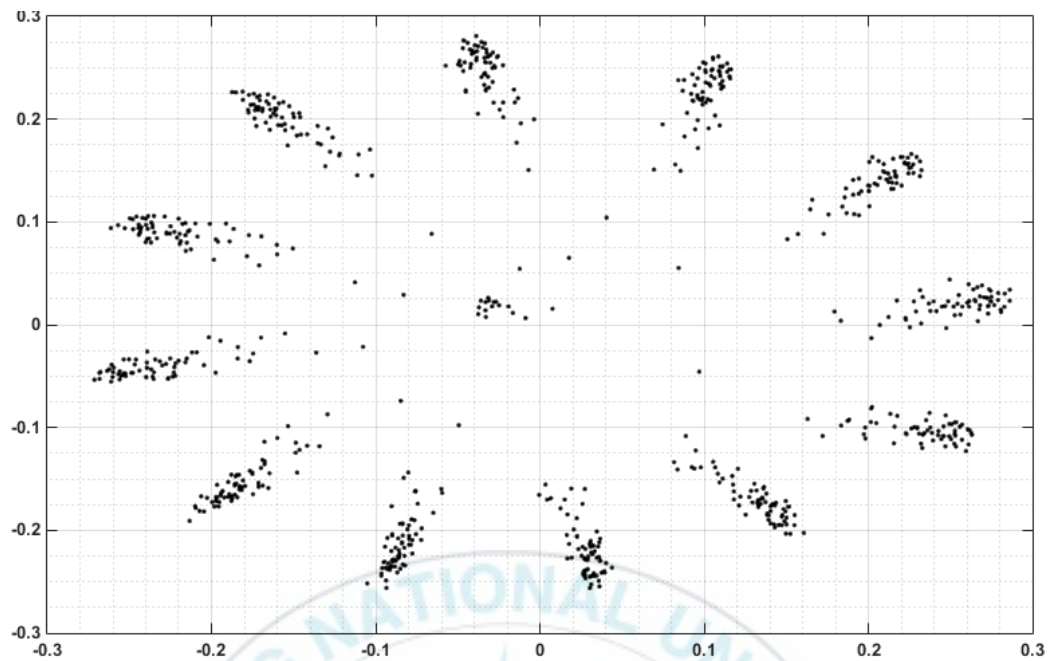


**Figure 4.5. Chromagram**

In order to obtain chromagram, then we multiply above spectrogram with C Matrix (Figure 4.4.). C Matrix is log frequency filter bank that have size : 12 semitones x number of frequency bin. After that, we can visualize 12 dimensional chroma features as chromagram in Figure 4.5 below

In above chromagram, it is clear enough by naked eyes to see 12 semitones in one octave. To see more chroma vector clearly, we also apply Principle Component Analysis (PCA) that only keep two components of chroma vectors as shown in Figure 4.6 below. At this point, we can use this features to train the HMM in order to recognize each pitch. HMM Training will be described later in this chapter.

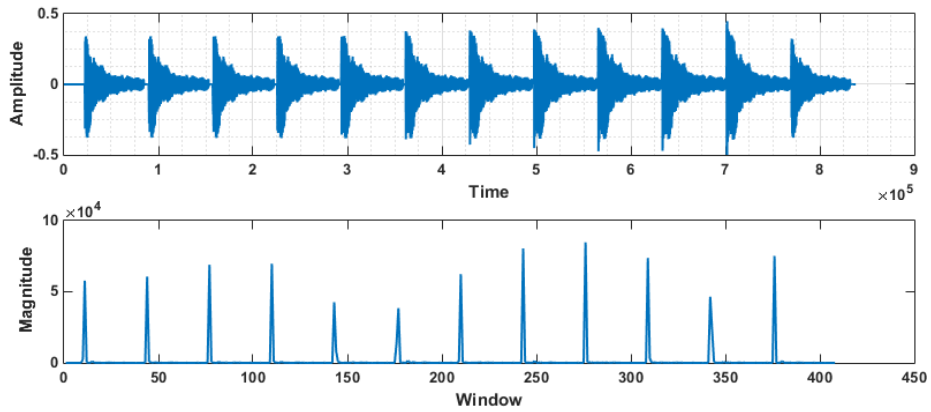




**Figure 4.6. 2D PCA of Chroma vectors**

#### **4.2.2. Onset Detection**

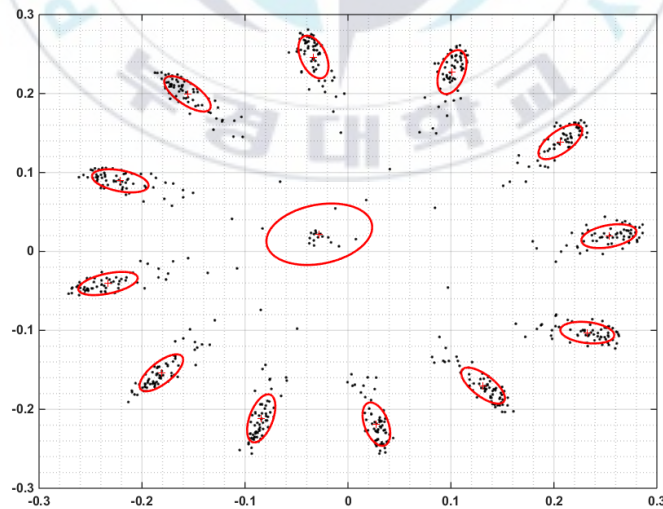
To identify musical events in music recording, we detect the onset using spectral flux. Spectral flux able to find onset by identify changes ini energy that would be the best first estimate of any onset detection function as shown in Figure 4.7. below



**Figure 4.7. (a) Music scales recording and its (b) onset estimation**

### 4.2.3. Training

In order to recognize the pitch, we train the continuous HMM with 13 states. Each states represent the tones including the silence. The training step is done by Baum-Welch algorithm. After hundred of iteration, the model tends to be able to recognize each tone in the chroma vectors as shown in Figure 4.8 below

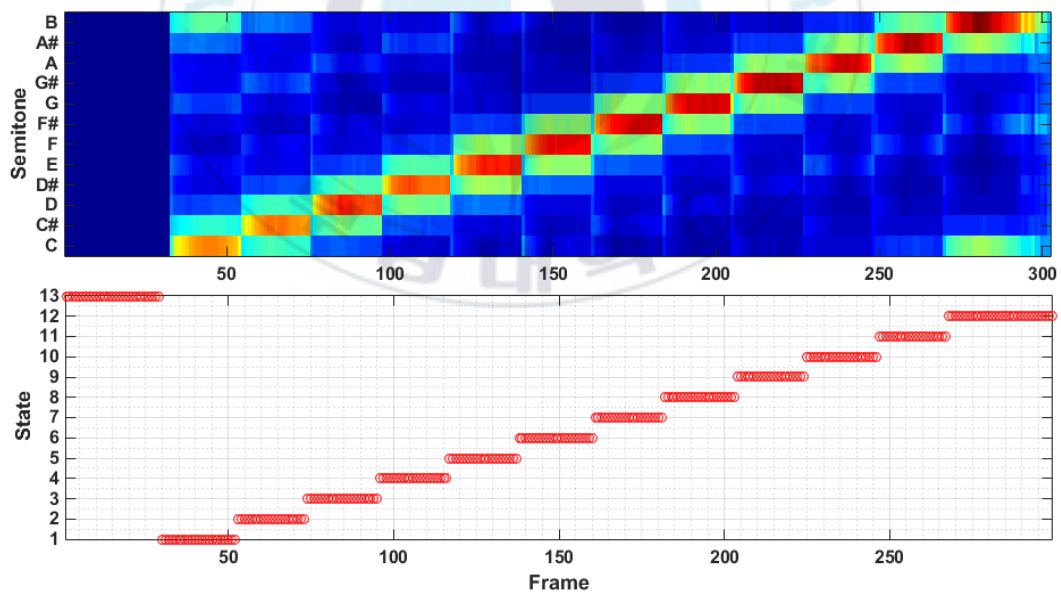


**Figure 4.8. Chroma vectors with ellipse as states of HMM**

In Figure 4.8., each ellipse represent the states which are gaussian distribution. Note that the chroma vector in the center part were the silence that also recognized by the HMM. Albeit more gaussian mixture will resulting more accurate model of a pitch given chroma features, a gaussian for each state is enough to recognize the pitch as shown in Figure 4.8. This training result then can be used to recognize the pitch of another music recording as will be describe in next sub section.

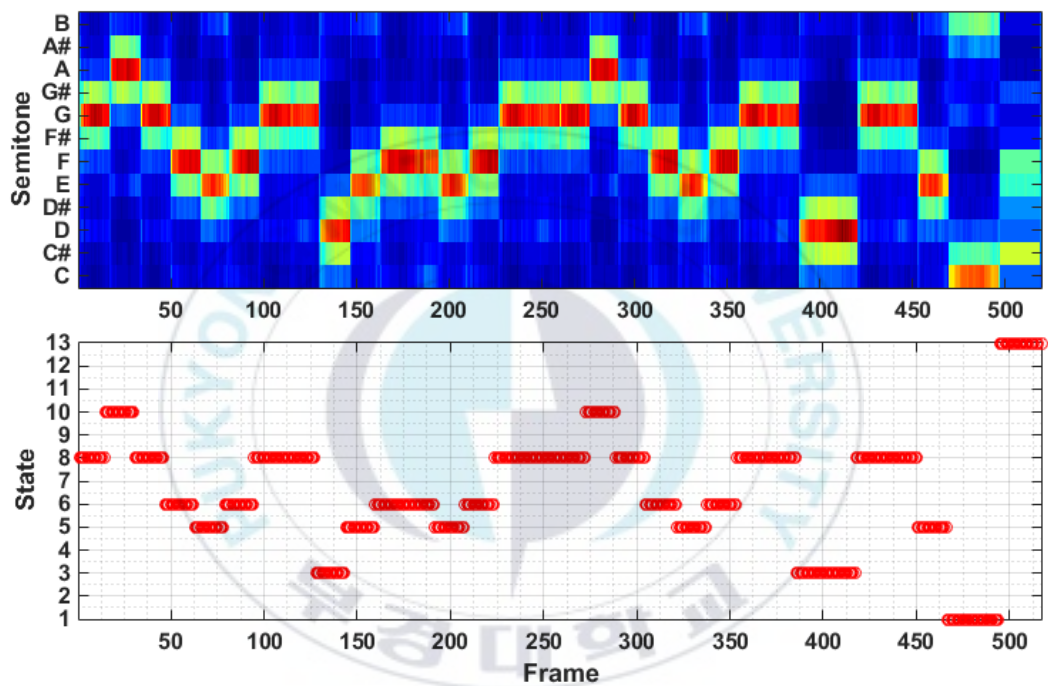
#### 4.2.4. Decoding

This step is mean to test the trained HMM wether it can recognize the pitch accurately or not. Decoding step is done by empolying the Viterbi decoding algorithm using trained HMM before.



**Figure 4.9. (a) Chromagram and (b) Viterbi decoding result of music scale performance**

Here we use 5 length of frames as buffer to be decode. Given the music performance, we extract the chroma features out of it then apply Viterbi decoding algorithm. Chromagram and its decoding result is shown in Figure 4.9. and Figure 4.10. In Figure 4.9. (b), Y axis represent the states of HMM. It is clearly to say that HMM able to recognize and tracks the pitch accurately.



**Figure 4.10. (a) Chromagram and (b) Viterbi decoding result of “London Bridge is Falling Down” music performance**

# **Chapter 5**

## **Conclusions**

### **5.1. Summary**

This research aim to design music learning assistant which have music score following features using audio-visual analysist. It consists of SVM and HMM. The SVM classifier recognizes music symbol in the simple music score. While, the HMM tracks the pitches given a music audio. If both of them are integrated with a proper synchronization, the system will be able to give feedback according to learner performance

### **5.2. Future Directions**

More data to be trained in training steps tends to obtain more accurate HMM. Thus, if there are more data from multiple instruments the HMM can be expect to recognize pitch from many instruments. In future research, it is necessary to provide multiple instruments as data training in order to design more robust music score following.

## References

- [1] Bartsch, M. A. and Wakefield, H., “Audio Thumbnailing of Popular Music Using Chroma-Based Representations” in *IEEE Transactions on Multimedia*, Vol. 7 No. 1. 2005.
- [2] Bello, J.P., Daudet, L., Abdallah, S., Duxbury, C. et all. “A Tutorial on Onset Detection in Music Signals” in *IEEE Transactions on Speech and Audio Processing* Vo. 13 No. 5, pp.1035-1047, 2005.
- [3] Bello, J.P. “Chroma and Tonality,” in *Music Information Retrieval Course*. Retrieved on August 2015 from <http://www.nyu.edu/classes/bello/MIR.html>
- [4] Brown, J. C., “Fundamental Frequency Tracking and Applications to Musical Signal Analysis” in *Analysis, Synthesis, and Perception of Musical Sounds* pp90-131, Springer New York, 2007.
- [5] Cano, P., Loscos, A., and Bonada, J. “Score-Performance Matching using HMMs,” in *Proc. Of the ICMC*, Sans Fransisco, pp.441-444, 1995.
- [6] Cho, T. and Bello, J.P. “Real-Time Implementation of HMM-Based Chord Estimation in Musical Audio,” in *Proc. of the International Computer Music Conference (ICMC)*, Canada, 2009.
- [7] Ellis, D. “Chroma Feature Analysis and Synthesis,” *MatLab Audio Processing Examples*. Retrieved on August 2015 from <http://www.ee.columbia.edu/~dpwe/resources/matlab/chroma-ansyn/>

- [8] Fujinaga, I., "Adaptive Optical Music Recognition" in *Ph.D Thesis*, McGill University, 1996.
- [9] Hess, A. "Beat Detection for Automated Music Transcription," in *Master Project of Masters of Science in Electrical Engineering, State University of New York, Birmingham*, 2011.
- [10] Horton, R., Bustamante, R.M., Edmonson, S.L., and Slate, J. R.. "Music and Student Performance: A Conceptual Analysis of the Literature", in *The Online Journal of New Horizons in Education (TOJNED)*, 2014.
- [11] Müller, M., Ellis, D.P., Klapuri, A., and Richard, G., "Signal Processing for Music Analysis," in *IEEE Journal of Selected Topics in Signal Processing*, Vol.5 No.6, pp.1088-1110, 2011.
- [12] Müller, M., "Short-Time Fourier Transform and Chroma Features" in *Lab Course of International Audio Laboratories Erlangen*, 2014.
- [13] Nguyen, T., and Lee, G. "A Lightweight and Effective Music Score Recognition on Mobile Phones," in *Journal of Information Processing Systems*, 2015.
- [14] Nilsson, M. "First Order Hidden Markov Model : Theory and Implementation Issues" in *Blekinge Institute of Technology Research Report*, 2005.
- [15] Orio, N. "An Automatic Accompanist Based on Hidden Markov Models," in *AI\*IA LNAI 2175*, Berlin, pp.64-69, 2001.
- [16] Pardo, B. and Birmingham, W., "Modeling Form for On-line Following of Musical Performances," in *Proceedings of the 20<sup>th</sup> National Conference on*



*Artificial Intelligence*, Pittsburgh, 2005.

- [17] Rabelo, A., Capela G., and Cardoso, J.S., “Optical recognition of music symbols” in *International Journal on Document Analysis and Recognition (IJDAR)*, 2010.
- [18] Rabelo, A., Fujinaga, I., Paskiewicz, F., et al, “Optical music recognition : State-of-the-art and open issues” in *International Journal of Multimedia Information Retrieval*, 2012.
- [19] Rabiner, L., “A Tutorial on Hidden Markov Modles and Selected Applications in Speech Recognition,” in *Proc. Of the IEEE*, Vol 77, pp.257-286, 1989.
- [20] Raphael, C. “Automatic Segmentation of Acoustic Musical Signals Using Hidden Markove Models,”in *IEEE Transactions on Pattern Analysis and Machine Intelligence Vol 21 No. 4*, 1999.
- [21] Raphael, C. “Aligning music audio with symbolic scores using a hybrid graphical model,” in *Machine Learning Vol 65 Issue 2-3*, pp.389-409, 2006.
- [22] Sako, S., Yamamoto, R., and Kitamura, T.. “Ryry : A Real-Time Score-Following Automatic Accompaniment Playback System Capable of Real Performances with Errors, Repeats and Jumps,” in *AMT LNCS 8610*, Switzerland, pp.124-145, 2014.
- [23] Sheh, A. and Ellis, D. “Chord Segmentation and Recognition using EM-Trained Hidden Markov Models,” in *4th International Symposium on Music Information Retrieval ISMIR-03*, pp. 185-191, Baltimore, October 2003.



- [24] Tralie, C., “Musical Pitches and Chroma Features”.
- Retrieved on August 2015 from
- [http://www.ctralie.com/Teaching/ECE381\\_DataExpeditions\\_Lab1/](http://www.ctralie.com/Teaching/ECE381_DataExpeditions_Lab1/)
- [25] “Musical Symbols”. Retrieved on August 2015 from
- <http://www.dolmetsch.com/musicalsymbols.htm>
- [26] Ebrahumzadeh, R., and Jampour, M.. “Efficient Handwritten Digit Recognition based on Histogram,” in International Journal of Computer Applications Vol. 104 No. 9, 2014.
- [27] \_\_\_\_\_. “Digit Classification Using HOG Features,” Retrieved on October 2015 from <http://www.mathworks.com/help/vision/examples/digit-classification-using-hog-features.html>
- [28] Schutte, K “MATLAB and MIDI” Retrieved online on October 2015 from <http://kenschutte.com/midi>
- [29] Zeng, A. “Optical Music Recognition” Retrieved online on October 2015 from <http://www.cs.princeton.edu/~andyz/ip/proj8/index.html>
- [30] D, Boswell. “Introduction to Support Vector Machines”, 2002.

## Acknowledgements

### (감사의 말씀)

First of all, I would like express my gratitude of blessing that given from our only God, Allah SWT who allow me seeking the knowledge abroad.

Study abroad was like a dream from me. Thanks to Dr. Gatot Hari Priowirjanto and Prof. Man-Gon Park that made my dream comes true through this 1<sup>st</sup> batch of ITB-PKNU Double Degree Program. Not to mention Dr. Abe Susanto, Mbak Cahya Ratih, Dr.techn. Ary Setijadi Prihatmanto, Prof. Dr. Carmadi Machbub, Mbak Any Siti Anisyah, SEAMOLEC and LSKK ITB staff who support us in the enrollment preparation and also during the study here.

For my beloved advisor professor, Prof. Bong-Kee Sin, I would like to express my highest gratitude for his gentle guidance, patience, and his caring. Not to mention, also for his dream and vision which motivated me a lot to study further and further after this step. I sincerely hope that your future dream will come true. I hope happiness and blessing will always be with you and your family.

I would like to express my sincere gratitude for Prof. Man-Gon Park and Dr. Myong-Hee Kim and their family who take care of us gently and received us likes their child with open and warm heart. May hapiness and blessing will always be with you and your family.

In the journey of seeking knowledge here, I would like to express my gratitude for all professor, Prof. Man-Gon Park, Prof. Bong-Kee Sin, Prof. Kyung-Hyune Rhee, Prof.

Chang-Soo Kim, and especially Dr. Myong-Hee Kim for their kindly and patiently teach us.

Above all, I would like to express my gratitude to my family that encourage and support me during my study here. My parents Atin Agustiani Widaningsih and Kurniadi, and my only grandmother Mbah Yati Suryati who I believe, their pray is incessant for me. Also my sister, Ratih Mulyawati and Herliana Pratiwi who support me and keep up my spirit until this thesis is done. Also my gratitude for Mr. Rumdi Raharja and his wife who support me during enrollment to this university.

Finally my college comrades, my labmates : Kokoy Siti Komariah, Maisevli Harika, 1<sup>st</sup> batch ITB-PKNU Double Degree students : Vandha Prawidyasma W, Diena Raudha R, Nurul Azhany, Taufiq Syahrir, Fairuz Iqbal Maulana, Wibby Aldriyani P, Sity Witty Aryanti, Kadek Restu Yani K, Rafinno Auliya, Heri Harum Nugroho, and specially my roommate Sandi Rahmadika to whom I would like to say thank you very much for your supports and thanks for paint colorful story of my life in here.

감사합니다