

โครงการวิทยาศาสตร์

โปรแกรมถอดเสียงดนตรีเป็นโน้ตดนตรีโดยใช้การแปลงฟูรีเย
Music transcribing program using Fourier Transform

นายปิยวัฒน์ อานันทคุณ
นายวิชา สกุลยืนยง
นายภัทรพล มุลแก่น

สาขาวิชาวิทยาการคอมพิวเตอร์
กลุ่มวิชาวิทยาศาสตร์และเทคโนโลยี
โรงเรียนมหิดลวิทยานุสรณ์ (องค์การมหาชน)
ปีการศึกษา 2557



ใบรับรองโรงงานวิทยาศาสตร์
โรงเรียนมหิตลวิทยานุสรณ์ (องค์การมหาชน)

มัธยมศึกษาตอนปลาย

วิทยาการคอมพิวเตอร์

หลักสูตร

สาขาวิชา

โปรแกรมถอดเสียงดนตรีเป็นโน้ตดนตรีโดยใช้การแปลงฟูรีเย
(Music transcribing program using Fourier Transform)

นามผู้ทำโครงการ	นายปิยวัฒน์ อานันทคุณ	ม.5/9	เลขประจำตัว 06628
	นายวิช สกุลยืนยง	ม.5/9	เลขประจำตัว 06625
	นายภัทรพล มูลแก่น	ม.5/9	เลขประจำตัว 06630

ได้พิจารณาเห็นชอบโดย

ประธานกรรมการ..... วันที่.....เดือน..... พ.ศ.

(อาจารย์ พิษณุตม์ อุปพันธ์)

กรรมการ..... วันที่.....เดือน..... พ.ศ.

(อาจารย์ เลขาวิญ งามประสิทธิ์)

กรรมการ..... วันที่.....เดือน..... พ.ศ.

(อาจารย์ ศิริพร ศักดิ์บุญญารัตน์)

รักษาการหัวหน้าสาขาวิทยาการคอมพิวเตอร์..... วันที่.....เดือน..... พ.ศ.

(อาจารย์ ศิริพร ศักดิ์บุญญารัตน์)

หัวข้อโครงการ	โปรแกรมถอดเสียงดนตรีเป็นโน้ตดนตรีโดยใช้การแปลงฟูรีเย <i>Music transcribing program using Fourier Transform</i>		
ผู้ทำโครงการ	นายปิยวัฒน์ อานันทคุณ		
	นายวิชา สกุลยืนยง		
	นายภัทรพล มูลแก่น		
อาจารย์ที่ปรึกษา	นายพิชญุตม์ อุปพันธ์ และ รศ.ดร.อติวงศ์ สุชาโต		
สาขา	วิทยาการคอมพิวเตอร์		
โรงเรียน	มหิดลวิทยานุสรณ์	ปีการศึกษา	2557

บทคัดย่อ

ในปัจจุบันการแปลงเสียงเพลงให้กลายเป็นโน้ตดนตรีมีความจำเป็นและมีความสำคัญอย่างยิ่งสำหรับนักดนตรีมือใหม่ เนื่องจากเพลงบางเพลงไม่สามารถหาโน้ตดนตรีได้บนอินเทอร์เน็ต ผู้จัดทำจึงได้พัฒนาอัลกอริทึมและสร้างโปรแกรมที่ใช้ในการแปลงเสียงเพลงที่อยู่ในรูปของไฟล์ wave (.wav) ให้เป็นโน้ตดนตรีที่อยู่ในรูปของไฟล์ MIDI (.mid) โดยอัลกอริทึมหลักที่ใช้คือ Fourier Transform ซึ่งสามารถแปลงข้อมูลเสียงจาก time-domain ให้อยู่ในรูปของ frequency-domain ได้ คณะผู้จัดทำได้คิดค้นอัลกอริทึมในการหาโน้ตดนตรีขึ้น ซึ่งสามารถแบ่งออกเป็น 2 ส่วน ได้แก่ การแบ่งช่วงข้อมูล และการวิเคราะห์โน้ตดนตรี ในส่วนของการแบ่งช่วงข้อมูล จะพิจารณาจากโน้ตที่เปลี่ยนไปเพื่อหาจุดเริ่มต้นของช่วงต่างๆ โดยดูจากความถี่และ power ของโน้ตแต่ละตัว และในส่วนของการวิเคราะห์โน้ตดนตรี จะทำการวิเคราะห์ frequency-domain ที่ได้จาก Fourier transform ในแต่ละช่วงและนำมาแปลงเป็นโน้ต ผู้จัดทำได้ใช้ Word Error Rate ในการวิเคราะห์ข้อมูล ซึ่งได้ว่าข้อผิดพลาดเฉลี่ยที่ได้จากการทดสอบอัลกอริทึมจากเพลง monophonic และเพลง polyphonic มีค่า 8.92% และ 62.90% ตามลำดับ

Research Title	Music transcribing program using Fourier Transform		
Researchers	Mr. Piyawat Anantakhun		
	Mr. Chawit Sakulyuenyong		
	Mr. Pattarapon Moonkaen		
Advisors	Mr. Pichayoot Oupaphan and Assc. Prof. Dr. Atiwong Suchato		
Department	Computer Science		
School	Mahidol Wittayanusorn School	Academic Year	2014

Abstract

Music transcription is necessary and useful for many musicians around the world, especially for amateur musicians. There are many computer programs that are good at analyzing monophonic music but not for polyphonic music; they have many limitations and cannot analyze the notes accurately. We proposed an algorithm that can analyze both monophonic and polyphonic music. The main algorithm that we used to analyze the music data is the Fourier transform, or more specifically, the Fast Fourier transform, which can separate the original sound wave to a summation of infinitely many sinusoidal waves and convert them into frequency-domain data. Our music transcribing program consisted of two main parts: digital signal processing and graphic user interface. In the digital signal processing part, we also divided our algorithm into two parts: interval determining and pitch analyzing. We used "Word Error Rate," a common method to analyze errors in speech recognition, to calculate the error rate of the results from our program. We found that our program was satisfactory at detecting synthesized monophonic music and moderately acceptable at detecting synthesized polyphonic music. However, our program still needs improvements to be able to work with real music data.

กิตติกรรมประกาศ

ขอขอบพระคุณ สาขาวิทยาการคอมพิวเตอร์ โรงเรียนมหิดลวิทยานุสรณ์สำหรับโอกาสที่ดีในการทำ
โครงงานนี้อีกทั้งยังเอื้อเฟื้อสถานที่และเครื่องมือที่จำเป็นต่อโครงงาน

ขอขอบพระคุณ อาจารย์พิชญุตม์ อุปพันธ์ และ รศ.ดร.อดิวงค์ สุชาโต อาจารย์ที่ปรึกษาที่คอยให้
คำแนะนำที่เป็นประโยชน์ในการทำโครงงานจนโครงงานสำเร็จลุล่วงได้ด้วยดี

ขอขอบคุณเพื่อนๆ ที่ได้ให้ความช่วยเหลือในเรื่องต่างๆ

ท้ายที่สุดนี้ ขอกราบขอบพระคุณ คุณพ่อและคุณแม่ ที่ให้กำลังใจและโอกาสในการศึกษาอันมีค่าซึ่งส่งผล
ให้เกิดการพัฒนาโครงงานนี้

คณะผู้จัดทำ

20 กุมภาพันธ์ 2557

สารบัญ

เรื่อง	หน้า
ใบรับรองโรงงานวิทยาศาสตร์	ก
บทคัดย่อ.....	ข
Abstract	ค
กิตติกรรมประกาศ	ง
สารบัญ	จ
สารบัญตาราง	ช
สารบัญรูปภาพ.....	ซ
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ.....	1
1.4 ระยะเวลาในการทำโครงการ.....	2
1.5 สถานที่ทำโครงการ	2
1.6 ประโยชน์ที่คาดว่าจะได้รับ	2
1.7 คำสำคัญ	2
บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง	4
2.1 Fourier transform.....	4
2.2 Wave file	8
2.3 MIDI file.....	9
บทที่ 3 วิธีดำเนินการทดลอง	10
3.1 อุปกรณ์ที่ใช้ในการทดลอง.....	10
3.1.1. อุปกรณ์และฮาร์ดแวร์ที่ใช้ในการพัฒนา	10
3.1.2 ซอฟต์แวร์ที่ใช้	10
3.2. วิธีการทดลอง.....	10
Interval determining	10
Pitch analyzing.....	12
บทที่ 4 ผลการทดลอง	13
ผลการทดลอง.....	13

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง	15
5.1 สรุปผลการทดลอง	15
5.2 ข้อเสนอแนะ	15
บรรณานุกรม	16
ประวัติผู้ทำโครงการ.....	17

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 1.4.1 ตารางปฏิบัติงาน.....	2
ตารางที่ 4.1 ตารางแสดงความผิดพลาดของอัลกอริทึมในการถอดโน้ตจากเพลง monophonic.....	13
ตารางที่ 4.1 ตารางแสดงความผิดพลาดของอัลกอริทึมในการถอดโน้ตจากเพลง polyphonic.....	14

สารบัญรูปภาพ

ภาพ	หน้า
ภาพที่ 2.2.1 กราฟของข้อมูลเสียงภายในไฟล์ .wav.....	8
ภาพที่ 3.2.1 วิธีการทำ binary-search ในอัลกอริทึม.....	11

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ไฟล์ .wav เป็นไฟล์เสียงที่เป็นมาตรฐานในระบบปฏิบัติการ Windows ของ Microsoft ที่ได้รับความนิยมสูงมาก และมีคุณภาพดี เนื่องจากเก็บข้อมูลมากถึง 44,100 sample ใน 1 วินาที จึงทำให้โปรแกรมเกือบทุกโปรแกรมรองรับมาตรฐานไฟล์เสียงชนิดนี้ ซึ่งทำให้ผู้ใช้สามารถหาไฟล์เสียงได้ง่ายจากอินเทอร์เน็ต และเนื่องด้วยการเล่นดนตรีได้รับความนิยมมากขึ้นในปัจจุบัน แต่นักดนตรีมือใหม่ไม่สามารถเล่นเพลงต่างๆที่ฟังจากอินเทอร์เน็ตได้ เนื่องจากบนอินเทอร์เน็ตอาจไม่มีซีพเพลงที่ต้องการ รวมทั้งยังมีความสามารถไม่เพียงพอที่จะถอดโน้ตดนตรีจากการฟังได้ จึงจำเป็นที่จะต้องใช้โปรแกรมช่วยฟังที่สามารถแปลงเสียงเพลงเป็นโน้ตดนตรีได้เพื่อพิจารณาแนวโน้ตของโน้ตของเพลงนั้นๆ

ผู้จัดทำจึงพัฒนาอัลกอริทึมและสร้างโปรแกรมที่ใช้ในการแปลงไฟล์ .wav ให้กลายเป็นไฟล์ MIDI หรือไฟล์ที่ระบุข้อมูลการเล่นของโน้ตหรือบรรทัด 5 เส้น และมีขนาดไฟล์ข้อมูลที่เล็กมากเมื่อเปรียบเทียบกับไฟล์ .wav เพื่อแก้ไขปัญหาดังกล่าวให้นักดนตรีกลุ่มนั้น จากการศึกษาพบว่าอัลกอริทึม Fast Fourier Transform เป็นอัลกอริทึมที่สามารถแปลงข้อมูลเสียงที่อยู่ในรูป time-domain ให้อยู่ในรูป frequency-domain ได้ ผู้จัดทำจึงเลือกอัลกอริทึมนี้ในการใช้วิเคราะห์หาข้อมูลของโน้ตที่ปรากฏในเพลงนั้นๆ

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1. เพื่อเขียนโปรแกรมที่สามารถแปลงเสียงดนตรีให้เป็นโน้ตดนตรีได้
- 1.2.2. เพื่อพัฒนาอัลกอริทึมในการถอดเสียงดนตรีเป็นโน้ตดนตรี

1.3 ขอบเขตของโครงการ

- 1.3.1 ใช้งานได้บนระบบปฏิบัติการ Windows
- 1.3.2. ข้อมูลนำเข้าจะต้องเป็นไฟล์ .wav ที่มีชื่อและที่อยู่ของไฟล์เป็นภาษาอังกฤษ
- 1.3.3. ข้อมูลนำเข้าจะต้องเป็นเสียงเปียโนสังเคราะห์

1.4 ระยะเวลาในการทำโครงการงาน

ขั้นตอน	เริ่ม	สิ้นสุด	วัน	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.
กำหนดปัญหาและตั้งหัวข้อ	1/3/2557	31/5/2557	90											
ศึกษาความเป็นไปได้	1/6/2557	7/6/2557	7											
วิเคราะห์ข้อมูล	1/6/2557	30/6/2557	30											
ออกแบบโปรแกรม	1/7/2557	31/7/2557	30											
สร้างโปรแกรม	1/8/2557	30/11/2557	120											
พัฒนาและปรับปรุงโปรแกรม	1/12/2557	31/12/2557	30											
เตรียมการนำเสนอโครงการงาน	1/12/2557	31/1/2557	60											

ตารางที่ 1.4.1 ตารางปฏิบัติงาน

1.5 สถานที่ทำโครงการงาน

สาขาวิชาวิทยาการคอมพิวเตอร์ โรงเรียนมหิดลวิทยานุสรณ์

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. นักดนตรีได้รับโน้ตที่มีความถูกต้อง แม่นยำ ตรงตามไฟล์เสียงต้นฉบับ
2. เพื่อช่วยเหลือผู้เล่นดนตรีให้สามารถฝึกเล่นดนตรีได้อย่างมีประสิทธิภาพ

1.7 คำสำคัญ

เพลง Monophonic หรือ Monophonic music

เพลงที่เล่นครั้งละโน้ต

เพลง Polyphonic หรือ Polyphonic music

เพลงที่เล่นหลายโน้ตพร้อมกัน

Word Error Rate (WER)

วิธีการทั่วไปที่ใช้วัดความถูกต้องของอัลกอริทึมในการรู้จำเสียงพูด (Speech Recognition) โดยสามารถคำนวณได้จากสูตร

$$WER = \frac{\text{deletion} + \text{insertion} + \text{substitution}}{\text{total number of words}}$$

การแปลงฟูรีเย หรือ Fourier Transform

อัลกอริทึมที่ใช้ในการแปลงข้อมูลจาก time-domain เป็น frequency domain

Sample

ข้อมูล 1 ตัว ใน wave file กล่าวคือ wave file เก็บข้อมูล 44,100 ตัวต่อวินาที นั่นคือ เก็บข้อมูล 44,100 sample ต่อวินาที

Peak ที่ x เพอร์เซ็นต์

sample ของข้อมูล frequency-domain ที่มีสมบัติคือ มี power มากกว่าหรือเท่ากับ x เพอร์เซ็นต์ของ power สูงสุดของข้อมูลนั้นๆ และจะต้องมีค่ามากที่สุดในช่วงที่มีความถี่ $2^{-1/30}$ ถึง $2^{1/30}$ เท่าของความถี่ที่ sample นั้นๆ

Time-domain

ข้อมูลของไฟล์เสียงที่มีแกน x เป็นค่าของเวลา และมีแกน y เป็นค่าของแอมพลิจูดของเสียง

Frequency-domain

ข้อมูลของไฟล์เสียงที่มีแกน x เป็นค่าความถี่ของคลื่น และมีแกน y เป็น power ของความถี่

Half-step

ระยะห่างครึ่งเสียงของโน้ตดนตรี เช่น C และ C# ห่างกัน 1 half-step

Fundamental frequency หรือ ความถี่มูลฐานของโน้ต

โน้ตแต่ละตัว จะมีความถี่มูลฐานที่คงตัวค่าหนึ่ง คำนวณได้จากสูตร

$$F_n = F_0 \times 2^{n/12}$$

โดยที่ F_n คือความถี่มูลฐานของโน้ตที่สนใจ, F_0 คือความถี่มูลฐานของโน้ตที่ทราบความถี่แล้ว และ n คือจำนวน half-step ระหว่างโน้ตที่สนใจกับโน้ตที่ทราบความถี่แล้ว

Harmonics หรือ ฮาร์โมนิกของโน้ต

ความถี่ที่มีค่าเป็น จำนวนเท่าของความถี่มูลฐาน เช่น 1 เท่า, 2 เท่า, 3 เท่า, ...

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

2.1 Fourier transform

ในปี ค.ศ. 1807 Joseph Fourier สามารถพิสูจน์ได้ว่า คลื่นที่มีคาบเกิดจากผลรวมของคลื่นที่มีความถี่ต่างกัน โดยแนวคิดพื้นฐานเริ่มจากการสังเกตว่า ฟังก์ชัน $\sin(x)$ และ $\cos(x)$ มีคาบเท่ากับ 2π หรือมีความถี่ $1/2\pi$ แต่ฟังก์ชัน $\sin(nx)$ และ $\cos(nx)$ มีคาบเท่ากับ $2\pi/n$ หรือมีความถี่ $n/2\pi$ ดังนั้น เมื่อนำคลื่นที่มีความถี่เป็นจำนวนเท่าของ $\sin(x)$ หรือ $\cos(x)$ ที่คูณด้วยค่าคงที่ต่าง ๆ กัน มาบวกกัน ยังคงได้คลื่นที่มีคาบ 2π หรือมีความถี่ $1/2\pi$ เช่นเดิม

จากแนวคิดดังกล่าว ได้ว่า สำหรับคลื่น $f(x)$ ที่มีคาบ T สามารถเขียนแทนได้ด้วย

$$f(x) = \frac{a_0}{2} + \sum_{m=1}^{\infty} \left(a_m \sin\left(m \frac{2\pi}{T} x\right) + b_m \cos\left(m \frac{2\pi}{T} x\right) \right)$$

โดยค่าของ a_0 , a_m , และ b_m สามารถพิสูจน์ ดังนี้

-> หา a_k คูณ $\sin\left(k \frac{2\pi}{T} x\right)$ โดยที่ $k \in N_0$ แล้วอินทิเกรตจาก $-\frac{T}{2}$ ถึง $\frac{T}{2}$ ซึ่งจากการที่

$$\int_{-T/2}^{T/2} \sin\left(m \frac{2\pi}{T} x\right) dx = 0$$

$$\int_{-T/2}^{T/2} \sin\left(m \frac{2\pi}{T} x\right) \sin\left(m \frac{2\pi}{T} x\right) dx = \begin{cases} 0, & m \neq n \\ \frac{T}{2}, & m = n \end{cases}$$

และ

$$\int_{-T/2}^{T/2} \sin\left(m \frac{2\pi}{T} x\right) \cos\left(n \frac{2\pi}{T} x\right) dx = 0$$

โดยที่ $m, n \in N$ ซึ่งจะได้ว่า

$$\int_{-T/2}^{T/2} f(x) \cos\left(k \frac{2\pi}{T} x\right) dx = a_k \frac{T}{2}$$

นั่นคือ

$$a_k = \frac{2}{T} \int_{-T/2}^{T/2} f(x) \cos\left(k \frac{2\pi}{T} x\right) dx$$

ในการทำงานเดียวกัน หา b_k คูณ $\cos\left(k\frac{2\pi}{T}x\right)$ โดยที่ $k \in N$ แล้วอินทิเกรตจาก $-\frac{T}{2}$ ถึง $\frac{T}{2}$ ซึ่งจากการที่

$$\int_{-T/2}^{T/2} \cos\left(m\frac{2\pi}{T}x\right) dx = 0$$

$$\int_{-T/2}^{T/2} \cos\left(m\frac{2\pi}{T}x\right) \cos\left(n\frac{2\pi}{T}x\right) dx = \begin{cases} 0, & m \neq n \\ \frac{T}{2}, & m = n \end{cases}$$

และ

$$\int_{-T/2}^{T/2} \sin\left(m\frac{2\pi}{T}x\right) \cos\left(n\frac{2\pi}{T}x\right) dx = 0$$

โดยที่ $m, n \in N$ ซึ่งจะได้ว่า

$$\int_{-T/2}^{T/2} f(x) \sin\left(k\frac{2\pi}{T}x\right) dx = b_k \frac{T}{2}$$

นั่นคือ

$$b_k = \frac{2}{T} \int_{-T/2}^{T/2} f(x) \sin\left(k\frac{2\pi}{T}x\right) dx$$

จาก

$$f(x) = \frac{a_0}{2} + \sum_{m=1}^{\infty} \left(a_m \sin\left(m\frac{2\pi}{T}x\right) + b_m \cos\left(m\frac{2\pi}{T}x\right) \right)$$

แทน $\omega = \frac{2\pi}{T}$ และ $x = t$ ได้ว่า

$$f(t) = \frac{a_0}{2} + \sum_{m=1}^{\infty} (a_m \sin m\omega t + b_m \cos m\omega t)$$

และจาก Euler's formula ได้ว่า

$$\cos m\omega x = \frac{e^{im\omega x} + e^{-im\omega x}}{2}$$

$$\sin m\omega x = \frac{e^{im\omega x} - e^{-im\omega x}}{2i}$$

ทำให้เขียน $f(t)$ ได้ในรูป

$$f(t) = \sum_{k=-\infty}^{\infty} c_k e^{ik\omega t}$$

$$\text{โดยที่ } c_k = \frac{2}{T} \int_{-T/2}^{T/2} f(t) e^{-ik\omega t} dt$$

แนวคิดนี้ได้ถูกขยายไปสู่คลื่นที่ไม่มีคาบ นั่นคือ สมมติให้ $T \rightarrow \infty$ แทนค่า c_k โดย เพื่อไม่ให้เกิดความสับสน จึงเปลี่ยนตัวแปร t ภายใน integrate เป็น τ ได้ว่า

$$f(t) = \sum_{k=-\infty}^{\infty} \frac{2}{T} \int_{-T/2}^{T/2} f(\tau) e^{-ik\omega \tau} d\tau e^{ik\omega t}$$

ซึ่งจากการที่ $T \rightarrow \infty$ ซึ่งทำให้ $\omega \rightarrow 0$ จึงได้ว่า $k\omega$ จะเป็นตัวแปรต่อเนื่อง จึงแทน $k\omega$ ด้วย ω และเปลี่ยนเครื่องหมาย sigma เป็นการ integrate และแทน $T = \frac{2\pi}{d\omega}$ ได้ดังนี้

$$f(t) = \int_{-\infty}^{\infty} \frac{d\omega}{\pi} \int_{-T/2}^{T/2} f(\tau) e^{i\omega(t-\tau)} d\tau$$

ซึ่งจะได้ว่า

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega t} d\omega \left[\int_{-\infty}^{\infty} f(\tau) e^{-i\omega \tau} d\tau \right]$$

โดยเรานิยามเพิ่มเติมว่า

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

ทำให้สามารถลดรูปฟังก์ชันลงได้เป็น

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega$$

โดยจะเรียกฟังก์ชัน $F(\omega)$ และ $f(t)$ ว่า Fourier Transform และ Inverse Fourier Transform ตามลำดับ ซึ่งฟังก์ชันดังกล่าวนี้ สามารถใช้ได้กับข้อมูลที่ต่อเนื่องเท่านั้น ดังนั้น จึงมีการสร้าง Discrete Fourier Transform ขึ้นสำหรับข้อมูลที่ไม่ต่อเนื่อง โดยการแบ่งช่วงออกเป็น N ส่วน ยาวช่วงละ $\Delta T = T/N$ นั่นคือจะสนใจคลื่นที่มีความถี่ $\omega_k = 2\pi k/T$ ใช้สัญลักษณ์ t_k แทน $k(\Delta T)$ โดยที่ $k = 0, 1, \dots, N-1$ และถือว่า $f(t_k)$ ที่ $k \geq N$ มีค่าเข้าใกล้ 0 จึงสามารถเขียนออกมาได้เป็น

$$F(\omega_n) = \sum_{k=0}^{N-1} f(t_k) e^{-i\omega_n t_k} , n = 0, 1, \dots, N-1$$

และนิยามฟังก์ชัน Inverse Discrete Fourier transform เป็น

$$f(t_k) = \frac{1}{N} \sum_{n=0}^{N-1} F(\omega_n) e^{i\omega_n t_k} , k = 0, 1, \dots, N-1$$

2.2 Wave file

ไฟล์เสียงนามสกุล .wav เป็นไฟล์เสียงที่เป็นมาตรฐานใน Windows ของ Microsoft ที่ได้รับความนิยมสูงมาก ทำให้โปรแกรมเกือบทุกโปรแกรมรองรับมาตรฐานไฟล์เสียงชนิดนี้ ไฟล์ .wav ประกอบด้วย 3 ส่วนหลักๆ คือ Header chunk, Format chunk และ ส่วนข้อมูลอื่นๆ

- Header chunk หรือส่วนระบุประเภทของไฟล์ ประกอบด้วยอักขระทั้งหมด 3 ชุด คือ
 - Id หรือประเภทไฟล์ ในไฟล์ WAVE จะเป็น “RIFF”
 - Size หรือขนาดข้อมูลทั้งหมดของไฟล์ มีหน่วยเป็น byte
 - RIFF type หรือประเภทของ RIFF ในที่นี้จะเป็น “WAVE”
- Format chunk หรือส่วนข้อมูลที่ประกอบด้วยส่วนข้อมูลของไฟล์ wave ต่างๆ ตัวอย่างเช่น
 - Sample rate หรือความละเอียดของไฟล์เสียง ซึ่งก็คือความถี่ของข้อมูลที่เก็บ โดยทั่วไปจะเป็น 44100 samples/second
 - ประเภทของเสียง โดย “0” จะแทน mono และ “1” จะแทน stereo
- ส่วนข้อมูลอื่นๆ ประกอบด้วยรายละเอียดเพิ่มเติมของไฟล์เสียงนั้นๆ เช่น ชื่อผู้แต่ง หรือลิขสิทธิ์ เป็นต้น รวมทั้งมีข้อมูลของเสียงเพลงอยู่ด้วย

ไฟล์เสียง .wav ใช้การเก็บข้อมูลในหน่วย sample และ แอมพลิจูด ซึ่งโดยทั่วไปแล้ว 44100 samples จะเท่ากับ 1 วินาที หรือ 1 sample เท่ากับ 2.27×10^{-5} วินาที โดยในแต่ละ sample จะเก็บค่า แอมพลิจูด อยู่ในช่วง -2^{16} ถึง 2^{16} ซึ่งสามารถนำมาแสดงในรูปของกราฟ 2 มิติได้ดังรูป



ภาพที่ 2.2.1 กราฟของข้อมูลเสียงภายในไฟล์ .wav

2.3 MIDI file

MIDI หรือ ไฟล์ .mid เป็นไฟล์ดิจิทัลที่เก็บรายละเอียดการเล่นของโน้ตแต่ละตัว ทั้งความยาว และความดังในการเล่น ทำให้นักดนตรีสามารถเล่นเพลงต่างๆ จากไฟล์ MIDI ได้ง่าย นอกจากนี้ ไฟล์ MIDI ยังเป็นมาตรฐานที่ยอมรับกันทั่วไปของนักดนตรี และสามารถแสดงในรูปบรรทัด 5 เส้น หรือสามารถแปลงเป็นไฟล์เสียงอื่นๆ ได้ รวมทั้งยังมีขนาดเล็กทำให้สะดวกต่อการเก็บและเคลื่อนย้ายไฟล์ ไฟล์ MIDI ประกอบด้วย 2 ส่วนหลักๆ คือ header chunk และ track chunk

- Header chunk หรือส่วนระบุประเภทของไฟล์ ประกอบด้วยชุดอักขระทั้งหมด 3 ชุด คือ
 - Id หรือประเภทไฟล์ ในไฟล์ MIDI จะใช้อักขระ “Mthd”
 - ประเภทของเพลง ในโปรแกรมจะระบุเป็น “0” หรือ track เดียว
 - Division time ระบุหน่วยของเวลาในการเล่นเพลง ในโปรแกรมจะใช้ 96 ticks per beat หรือ 1 โน้ตตัวดำจะเท่ากับ 96 ticks
- Track chunk หรือส่วนระบุข้อมูลย่อยในแต่ละ track จะประกอบด้วยชุดอักขระทั้งหมด 3 ชุด คือ
 - Id หรือประเภทของข้อมูลใน track จะใช้อักขระ “Mtrk”
 - ส่วนระบุข้อมูลเบื้องต้นของเพลง เช่น ประเภทของเครื่องดนตรี ความยาวของแต่ละห้อง และ tempo เป็นต้น
 - Meta event จะระบุข้อมูลของโน้ตแต่ละตัวที่เล่น พร้อมระยะเวลาในการเล่น โดยมีหน่วยเป็น tick ซึ่งสอดคล้องกับค่า Division time นอกจากนี้ยังมีชุดคำสั่งอื่นๆ เช่น ชุดคำสั่งในการปิดเสียงที่เล่นอยู่ทั้งหมด การเปลี่ยน tempo ในการเล่น และการเปลี่ยนชนิดของเครื่องดนตรีที่เล่น
 - ชุดอักขระที่บ่งบอกถึงจุดสิ้นสุดของแต่ละ track โดยจะระบุในรูปแบบของ hex หรือเลขฐาน 16 ได้ว่า “FF 2F 00”

บทที่ 3

วิธีดำเนินการทดลอง

3.1 อุปกรณ์ที่ใช้ในการทดลอง

3.1.1. อุปกรณ์และฮาร์ดแวร์ที่ใช้ในการพัฒนา

- Notebook SAMSUNG Series 5 NP510R5E-S01T56H
- Notebook DELL Vostro 5460-V560303TH
- SONY Vaio SVE14128CHB

3.1.2 ซอฟต์แวร์ที่ใช้

- Microsoft Windows 8
- Microsoft Visual Studio 2013 Community
- Code::Block v.13.12
- Aria Maestosa

3.2. วิธีการทดลอง

1. เตรียมไฟล์ MIDI ต้นฉบับที่เป็น monophonic music จำนวน 10 เพลง และเป็น polyphonic music จำนวน 10 เพลง
2. ทำการแปลงไฟล์จากไฟล์ MIDI ทั้งหมดให้เป็น ไฟล์ .wav
3. ในแต่ละไฟล์ ทำการรันอัลกอริทึมในส่วน Digital signal processing ได้แก่ Interval determining และ Pitch analyzing ตามลำดับ
4. สร้างไฟล์ MIDI ผลลัพธ์ขึ้น
5. นำไฟล์ MIDI ผลลัพธ์ไปเปรียบเทียบกับไฟล์ MIDI ต้นฉบับ โดยใช้ Word Error Rate ในการหาค่าความคลาดเคลื่อนของโปรแกรม
6. วิเคราะห์และสรุปผลการทดลองจากข้อมูลที่ได้จากการทดลอง

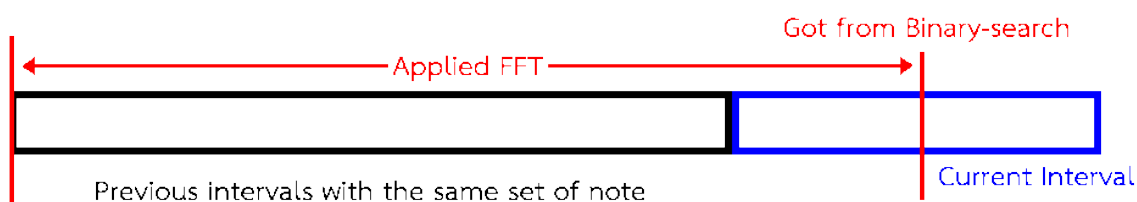
ในที่นี้ จะขออธิบายฟังก์ชันต่างๆที่เกี่ยวข้องกับการทดลอง ได้แก่ Interval determining และ Pitch analyzing

Interval determining

ในส่วนของ Interval determining เป็นส่วนที่ใช้ในการแบ่งเพลงออกเป็นช่วงๆ เพื่อทำการวิเคราะห์โน้ตในขั้นถัดไป โดยมีรายละเอียด ดังนี้

1. อ่านข้อมูลจาก ไฟล์ .wav ซึ่งได้ข้อมูลเสียงที่อยู่ในรูป time-domain

2. เลือกข้อมูลที่ละ 2,048 หรือ 2^{11} sample จากนั้น ทำการรันฟังก์ชัน Fast Fourier Transform
3. นำข้อมูล frequency-domain มาหา peak ที่ 50%
4. สำหรับทุกๆโน้ต จะเปลี่ยนความถี่ของ peak ทั้งหมดที่อยู่ในช่วง $2^{-1/24}$ ถึง $2^{1/24}$ เท่าของความถี่มูลฐานของโน้ตนั้นๆ ให้มีค่าเท่ากับความถี่มูลฐานของโน้ตนั้นๆ (ใช้ $2^{1/24}$ เนื่องจากการหาความถี่กึ่งกลางระหว่าง 2 โน้ต ตามสูตรการหาความถี่มูลฐาน)
5. ตรวจสอบจากโน้ตที่มีความถี่ต่ำสุด หากมี peak ของความถี่นั้นๆปรากฏ จะทำการหา power สูงสุดของ peak ที่เป็น harmonics แล้วนำมาเก็บไว้แทน power ค่าเดิม แล้วตัด harmonics ทั้งหมดของโน้ตทิ้ง ยกเว้นตัวมันเอง
6. ทำข้อ 5 ซ้ำเรื่อยๆจนครบทุกโน้ต
7. จะตรวจสอบว่า ช่วงที่กำลังพิจารณามีโน้ตใหม่เกิดขึ้นหรือไม่ โดยถือว่าโน้ตใหม่เกิดขึ้น เมื่อเงื่อนไขใดเงื่อนไขหนึ่งต่อไปนี้เป็นจริง
 - กรณีที่ 1: ถ้าพบว่า ในช่วงถัดไปพบโน้ตใหม่ที่ไม่ปรากฏในช่วงก่อนหน้านี้ และโน้ตที่ปรากฏมาใหม่นั้น มี power มากกว่าหรือเท่ากับ 75% ของ power สูงสุด จะถือว่าโน้ตใหม่เกิดขึ้น
 - กรณีที่ 2: ถ้าไม่มีโน้ตใหม่ปรากฏขึ้น แต่ power ของโน้ตเดิมนั้น มีการเพิ่มสูงขึ้นจน 70% ของ power ของโน้ตนั้นในช่วงใหม่ มากกว่าหรือเท่ากับ power ของโน้ตนั้นในช่วงก่อนหน้านี้ ก็ถือว่าโน้ตใหม่เกิดขึ้นเช่นกัน
8. หากในช่วงปัจจุบัน พบว่ามีโน้ตใหม่เกิดขึ้น จะใช้จุดเริ่มต้นของช่วงก่อนหน้านี้เป็นจุดเริ่มต้น และจะทำการ binary-search ช่วงปัจจุบัน เพื่อใช้เป็นจุดปลาย จากนั้นจะรัน Fourier Transform กับข้อมูลในช่วงจุดต้นถึงจุดปลาย แล้วนำข้อมูล frequency-domain มาวิเคราะห์ โดยจะทำการ binary-search จนกระทั่งเงื่อนไขต่อไปนี้เป็นจริง
 - 8.1. หากการเกิดโน้ตใหม่ตรงกับเงื่อนไขของข้อ 7 กรณีที่ 1 จะ binary-search จนกระทั่ง power ของความถี่ใหม่ที่ปรากฏขึ้น มีค่าเข้าใกล้ 70% ของ power สูงสุด
 - 8.2. หากการเกิดโน้ตใหม่ตรงกับเงื่อนไขของข้อ 7 กรณีที่ 2 จะ binary-search จนกระทั่ง 70% ของ power ของโน้ตเดิม มีค่าเข้าใกล้ power ของโน้ตเดิมในช่วงก่อนหน้านี้ (หรือนั่นคือ power ของโน้ตเดิมมีค่าเข้าใกล้ 1.43 เท่าของ power ของโน้ตเดิมในช่วงก่อนหน้านี้)



ภาพที่ 3.2.1 วิธีการทำ binary-search ในอัลกอริทึม

Pitch analyzing

ในส่วนของ Pitch analyzing เป็นส่วนที่ใช้ในการวิเคราะห์หาโน้ตที่ปรากฏในช่วงต่างๆ ที่หาได้ในส่วนของ Interval determining โดยมีรายละเอียด ดังนี้

1. ในการพิจารณาแต่ละช่วงที่แบ่งได้ ก่อนการรันฟังก์ชัน Fast Fourier Transform จะทำการปรับแต่งข้อมูล โดยเลือกข้อมูลเฉพาะส่วนที่อยู่ตรงกลางของช่วง เนื่องจากตอนต้นและตอนท้ายจะก่อให้เกิดความคลาดเคลื่อนได้ โดยการเลือกข้อมูลตรงกลางจะพิจารณาจากความยาวของช่วง ดังนี้
 - 1.1. ในกรณีที่ความยาวช่วงน้อยกว่า 5,000 sample จะเลือกข้อมูล 90% กึ่งกลาง กล่าวคือ ตัด 5% ทั้งตอนต้นและตอนท้ายของช่วง
 - 1.2. ในกรณีที่ความยาวช่วงอยู่ระหว่าง 5,000 ถึง 16,000 sample จะเลือกข้อมูล 80% กึ่งกลาง กล่าวคือ ตัด 10% ทั้งตอนต้นและตอนท้ายของช่วง
 - 1.3. ในกรณีที่ความยาวช่วงที่อยู่ระหว่าง 16,000 ถึง 20,480 จะเลือกข้อมูลโดยตัดข้อมูล 1,600 sample แรก และ 1,600 sample ท้ายทิ้ง
 - 1.4. ในกรณีที่ความยาวช่วงมากกว่า 20,480 sample จะเลือกข้อมูลตั้งแต่ sample ที่ 1,600 ไปจนถึง sample ที่ 17,984 ซึ่งจะได้จำนวน sample เท่ากับ 16,384 หรือ 2^{14} พอดี
2. นำข้อมูล frequency-domain ที่ได้ในแต่ละช่วงมาหา peak ที่ 50%
3. เลือก peak ที่มีค่า power สูงสุด จากนั้นทำการเปลี่ยนค่าความถี่ของ peak ให้มีค่าเท่ากับความถี่มูลฐานของโน้ตที่อยู่ใกล้ที่สุด โดยจะถูกเปลี่ยนให้เป็นความถี่มูลฐานของโน้ตที่ต่ำกว่า เมื่อความถี่ของ peak น้อยกว่า $2^{1/24}$ เท่าของความถี่ของโน้ตที่ต่ำกว่า หากเงื่อนไขดังกล่าวไม่เป็นจริง จะถูกเปลี่ยนให้เป็นความถี่มูลฐานของโน้ตถัดไปแทน และทำการ mark ว่า โน้ตนั้นๆ ปรากฏอยู่ในช่วงที่พิจารณา
4. ทำการตัด harmonics ทั้งหมดของโน้ตที่เลือกในข้อ 3
5. ทำซ้ำข้อ 3-4 จนทุก peak ถูกตัด
6. นำโน้ตทั้งหมดที่เหลือ มาพิจารณาตามลำดับจากความถี่มูลฐานต่ำสุด ไปสูงสุด ถ้าโน้ตเดียวกันในออกเทพที่ต่ำกว่า ถูก mark ไว้แล้ว จะทำการตัดโน้ตในออกเทพปัจจุบันทิ้ง
7. บันทึกโน้ตทั้งหมดที่ได้เป็นโน้ตที่เล่นในช่วงนั้นๆ

บทที่ 4

ผลการทดลอง

ผลการทดลอง

จากการทดลองและคำนวณความผิดพลาดจาก “Word Error Rate” ได้ผลดังตารางต่อไปนี้

Song name	% Insert	% Delete	% Substitute	% ERROR
01 Mary Had A Little Lamb	0.00	0.00	3.22	3.22
02 Yankee Doodle	0.00	0.00	7.50	7.50
03 Twinkle Twinkle Little Star	0.00	0.00	4.44	4.44
04 Ode To Joy	0.00	0.00	7.09	7.09
05 Old MacDonald	3.33	0.00	5.70	9.04
06 Joy To The World	0.76	0.00	4.22	4.98
07 Jingle Bells	7.87	0.00	9.88	17.75
08 London Bridge	0.00	0.00	7.75	7.75
09 Happy Birthday	0.00	0.00	9.78	9.78
10 We Wish You A Merry Christmas	0.00	0.00	17.70	17.70
Average	1.20	0.00	7.73	8.92

ตารางที่ 4.1 ตารางแสดงความผิดพลาดของอัลกอริทึมในการถอดโน้ตจากเพลง monophonic

Song name	% Insert	% Delete	% Substitute	% ERROR
01 Canon In D	2.53	50.71	10.75	63.99
02 Für Elise	6.29	15.47	23.41	45.17
03 Beethoven Virus	5.36	35.65	20.71	61.72
04 Moonlight Sonata 2nd Movement	14.41	39.84	10.76	65.00
05 Pathétique 2nd Movement	4.00	40.94	12.74	57.68
06 Mozart Turkish March	46.16	28.12	25.41	99.69
07 Flower Dance	11.73	27.76	9.01	48.50
08 River Flows In You	3.48	28.37	7.28	39.13
09 Winter 2nd movement	5.39	23.82	23.65	52.86
10 New world symphony	48.82	27.40	19.01	95.23
Average	14.82	31.81	16.27	62.90

ตารางที่ 4.2 ตารางแสดงความผิดพลาดของอัลกอริทึมในการถอดโน้ตจากเพลง polyphonic

บทที่ 5

สรุปและวิจารณ์ผลการทดลอง

5.1 สรุปผลการทดลอง

เราสามารถเขียนโปรแกรมที่ใช้ในการแปลงเสียงดนตรีให้อยู่ในรูปโน้ตดนตรีได้สำเร็จ และจากการทดสอบ อัลกอริทึมด้วยวิธีการ “Word Error Rate” พบว่าความผิดพลาดเฉลี่ยของเพลง monophonic และ เพลง polyphonic อยู่ที่ 8.92% และ 62.90 % ตามลำดับ

5.2 ข้อเสนอแนะ

- ควรเปลี่ยนโมเดลที่ใช้ในการตรวจสอบความผิดพลาดของโปรแกรม
- ควรเพิ่มฟังก์ชันการทำงานต่างๆให้มากขึ้น เช่น เพิ่มฟังก์ชันที่สามารถรันโปรแกรมบนช่วงหนึ่งของไฟล์ .wav ที่เป็นไฟล์ input ได้
- ควรปรับปรุงประสิทธิภาพของโปรแกรมให้ดีขึ้น เช่น ทำให้ระยะเวลาในการรันโปรแกรมสั้นลง
- ควรเพิ่มความสวยงามในส่วนของ Graphical user interface

บรรณานุกรม

- "Wave File Format" (n.d.). Retrieved January 18, 2015, from <http://www.sonicspot.com/guide/wavefiles.html>
- Bello, J., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., & Sandler, M. (2005). A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5), 1035-1047. Retrieved November 11, 2014, from http://www.nyu.edu/classes/bello/MIR_files/2005_BelloEtAl_IEEE_TSALP.pdf
- Fast Fourier transform — FFT. (n.d.). Retrieved December 17, 2014, from <http://www.librow.com/articles/article-10>
- Raphael, C. (2002). Automatic Transcription of Piano Music. Automatic Transcription of Piano Music. Retrieved June 14, 2014, from http://music.informatics.indiana.edu/~craphael/papers/ismir02_rev.pdf
- Standard MIDI-File Format Spec. 1.1, updated. (n.d.). Retrieved January 22, 2015, from <http://www.music.mcgill.ca/~ich/classes/mumt306/StandardMIDIfileformat.html>

ประวัติผู้ทำโครงการ

ชื่อ สกุล	นายปิยวัฒน์ อานันทคุณ
เลขประจำตัวนักเรียน	06628
วันเดือนปีเกิด	4 มิถุนายน พ.ศ. 2540
สถานที่อยู่	71 หมู่ 3 ถนนเพชรมาตุคลา ตำบลมะเร็ง อำเภอเมือง จังหวัดนครราชสีมา 30000
ประวัติการศึกษา	สำเร็จการศึกษาจาก ชั้นประถมศึกษาปีที่ 6 จาก โรงเรียนอนุบาลนครราชสีมา ชั้นมัธยมศึกษาปีที่ 3 จาก โรงเรียนราชสีมาวิทยาลัย ปัจจุบันศึกษาอยู่ที่ โรงเรียนมหิดลวิทยานุสรณ์
ชื่อ สกุล	นายวิชา สกุลยืนยง
เลขประจำตัวนักเรียน	06625
วันเดือนปีเกิด	10 พฤษภาคม พ.ศ. 2541
สถานที่อยู่	116 หมู่ 3 ถนนพหลโยธิน ตำบลสุขสำราญ อำเภอตากฟ้า จังหวัดนครสวรรค์ 60190
ประวัติการศึกษา	สำเร็จการศึกษาจาก ชั้นประถมศึกษาปีที่ 6 จาก โรงเรียนปรียาโชติ ชั้นมัธยมศึกษาปีที่ 3 จาก โรงเรียนตากลีประชาสรรค์ ปัจจุบันศึกษาอยู่ที่ โรงเรียนมหิดลวิทยานุสรณ์
ชื่อ สกุล	นายภัทรพล มูลแก่น
เลขประจำตัวนักเรียน	06630
วันเดือนปีเกิด	2 กุมภาพันธ์ พ.ศ. 2541
สถานที่อยู่	3/584 ซอยเทิดราชนัน 47 ถนนเทิดราชนัน แขวงสีกัน เขตดอนเมือง กรุงเทพมหานคร 10210
ประวัติการศึกษา	สำเร็จการศึกษาจาก ชั้นประถมศึกษาปีที่ 6 จาก โรงเรียนแสงอรุณ ชั้นมัธยมศึกษาปีที่ 3 จาก โรงเรียนหอวัง ปัจจุบันศึกษาอยู่ที่ โรงเรียนมหิดลวิทยานุสรณ์