

มัลติมีเดีย MULTIMEDIA

โดย

นาย นพรัตน์ วิจิตรวรรณกมล 34103168

นาย หฤษฎ์ อัครชัยราชันย์ 34108471

อาจารย์ที่ปรึกษา

รศ.ดร. ศรรชิต ไมตรี



วิทยานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2537

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง มัลติมีเดีย (Multimedia)

ผู้จัดทำ

1. นาย นพรัตน์ วิจิตรรณกมล 34103168
2. นาย หฤษฎ์ อัสวชัยราชันย์ 34108471



อาจารย์ที่ปรึกษา

(รศ. วิจารณ์ ใจดี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มัลติมีเดีย

นพรัตน์ วินิจวรรณกมล

หญิง อัครชัยราชันย์

รศ.ดร. ครรชิต โมตรี อาจารย์ที่ปรึกษา

ปีการศึกษา 2537

บทคัดย่อ

ในวิทยานิพนธ์ฉบับนี้ ได้นำเอาเทคโนโลยีทางด้านมัลติมีเดีย มาใช้ในการทำการเชื่อมต่อการสั่งงานคอมพิวเตอร์ (User Interface) เพื่อให้สามารถสั่งงานคอมพิวเตอร์ โดยใช้เสียงที่เป็นคำพูดภาษาไทย โดยจะทำการเก็บตัวอย่างเสียงผ่านทางซาวด์บลาสเตอร์ แล้วทำการวิเคราะห์เสียงตัวอย่าง เพื่อหาลักษณะเด่นของเสียงแต่ละแบบออกมาเป็นค่าพารามิเตอร์ที่ใช้ในการแยกแยะความแตกต่างของเสียงพูดแต่ละคำ โดยใช้วิธีการทางสถิติในการจดจำรูปแบบ (Pattern Recognition) เพื่อแยกแยะความแตกต่างของคำพูดแต่ละคำ ซึ่งการพูดแต่ละครั้งของแต่ละคนจะมีความแตกต่างกันทั้งในด้านของ จังหวะการออกเสียง ความถี่ และ ความดังในการออกเสียง ค่าพารามิเตอร์ที่ได้จะเก็บเป็นตัวอย่าง เพื่อใช้ในการเปรียบเทียบกับเสียงที่เป็น input ในการสั่งงานคอมพิวเตอร์โดยไม่ขึ้นกับผู้พูด ทั้งนี้คำต่างๆจะต้องเป็นคำสั้นๆ และ ผู้พูดจะต้องเว้นช่วงในการพูดแต่ละคำให้ชัดเจน เพื่อให้สามารถตัดคำได้

โปรแกรมนี้ได้สร้างขึ้นมา ให้ใช้งานบนระบบปฏิบัติการ WINDOWS 3.1 ที่ติดตั้งซาวด์บลาสเตอร์ซึ่งจะบันทึกเสียงในระบบ 8 บิต ความถี่ 11 kHz ซึ่งข้อมูลที่ได้จากการบันทึกนี้จะอยู่ในรูปของ Time Domain

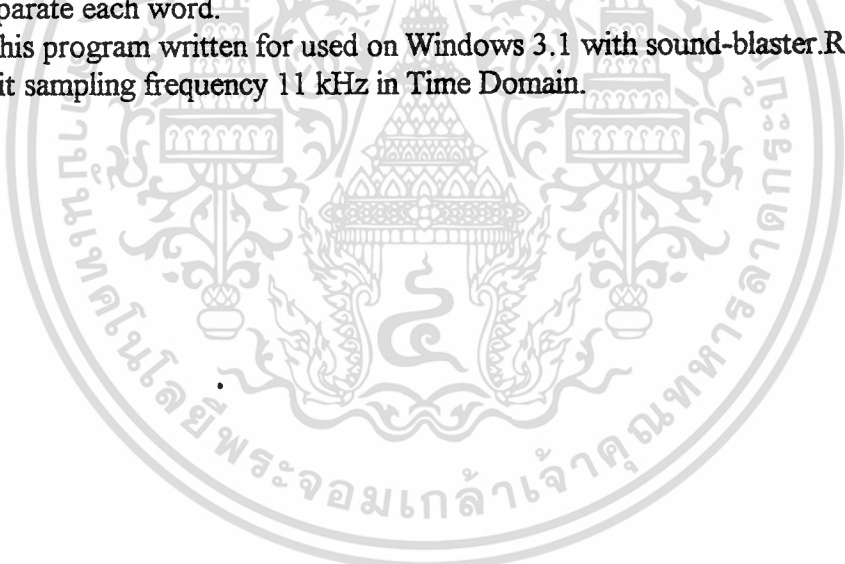
MULTIMEDIA

Noparut Vinijwonkmon
Harit Ussawachairachan
Associate Professor Dr. Kanchit Maitree Advisor
1994

ABSTRACT

This thesis is to bring the technology of multimedia to make the sound interface between user and computer in Thai language sound. To analysis the example sound that record by sound-blaster to extract feature of sound into the Parameter to distinguish the words. By use the statistical method of Pattern Recognition to distinguish the sound that different in the way of interval of vowel sound ,frequency and loudness. These Parameter will compare with the parameter of input sound that independent from speaker . Even though the word must has short sound and the speaker must say separatly word by word to easy to separate each word.

This program written for used on Windows 3.1 with sound-blaster. Record sound in digital 8 bit sampling frequency 11 kHz in Time Domain.



สารบัญ

content

บทที่	หน้า
1. บทนำ	1
ความเป็นมาของโครงการ	1
วัตถุประสงค์	2
ขอบเขตและข้อจำกัดของโครงการ	2
ขั้นตอนและวิธีดำเนินงาน	2
2. ทฤษฎีและหลักการ	3
หลักการออกเสียง	3
ลำดับขั้นการออกเสียง	4
อวัยวะในการออกเสียง	6
หลักการในการจดจำเสียง	6
ทฤษฎีและหลักการที่ใช้	7
เทคนิคทั่วไปของ pattern-recognition	8
Template matching	8
Statistical methods	11
Structural techniques	13
ทฤษฎีที่ใช้ในการประมวลผลสัญญาณ	16
Median filtering	16
The Laplacian	18
3. การเขียนโปรแกรมบน WINDOWS	20
รูปแบบของแอฟลิเคชัน	22
ไลบรารีของ WINDOWS	23
ขั้นตอนในการสร้างแอฟลิเคชัน	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Sound Blaster Card	25
ฮาร์ดแวร์ที่ใช้ในงาน	25
โครงสร้างทั่วไปของ Sound Blaster Card	26
การติดตั้ง Sound Blaster Card	27
5. การคำนวณและการสร้าง	28
การเรียนรู้	29
การอ่านข้อมูลจากไฟล์เสียง	30
ส่วนประมวลผล	32
หลักเกณฑ์ในการหาของเขต	34
6. การทดลองและผลการทดลอง	40
วิธีทำการทดลอง	40
ตารางบันทึกผลการทดลองและกราฟ	
จำนวนแถวที่มีข้อมูลของแต่ละเสียง	41
จำนวนจุดที่มีค่า laplacian มากกว่า 200	42
จำนวนจุดที่มีค่า laplacian ระหว่าง -1 และ 101	43
จำนวนจุดที่มีค่า laplacian ระหว่าง 100 และ 200	44
จำนวนจุดที่มีค่า laplacian ระหว่าง 100 และ 245	45
จำนวนจุดที่มีค่า laplacian ระหว่าง 200 และ 245	46
จำนวนจุดที่มีค่า laplacian ระหว่าง 0 และ 200	47
จำนวนจุดที่มีค่า laplacian ระหว่าง 0 และ 30	48
จำนวนจุดที่มีค่า laplacian ระหว่าง 0 และ 20	49
จำนวนจุดที่มีค่า laplacian ระหว่าง 0 และ 10	50
จำนวนจุดที่มีค่า laplacian ระหว่าง 5 และ 30	51
จำนวนจุดที่มีค่า laplacian ระหว่าง 200 และ 253	52
จำนวนจุดที่มีค่า laplacian ระหว่าง 244 และ 253	53
จำนวนจุดที่มีค่า laplacian ระหว่าง 100 และ 253	54
สรุปผลการทดลอง	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.บทวิจารณ์และสรุป	56
สรุปลักษณะของข้อมูลที่ใช้เป็นค่าสถิติของแต่ละค่า	57
สรุปผลโครงการ	58

ภาคผนวก

เอกสารอ้างอิง

กิตติกรรมประกาศ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปภาพ

รูปที่ 2.0	อวัยวะที่ใช้ในการออกเสียง	5
รูปที่ 2.1	สาขาการศึกษาถึงการใช้งาน pattern-recognition	7
รูปที่ 2.2	แสดงการทำ template matching	9
รูปที่ 2.3	Statistical pattern recognizer	11
รูปที่ 2.4	การแบ่ง feature space	13
รูปที่ 2.5	ตัวอย่าง structural classification	14
รูปที่ 2.6	ไดอะแกรมของ structural recognition	15
รูปที่ 3.1	แสดงส่วนประกอบต่างๆของวินโดว์	25
รูปที่ 3.2	การรับข้อมูลผ่านทางเมตเสจคิวของสองแอปพลิเคชัน	26
รูปที่ 3.3	ขั้นตอนการสร้างแอปพลิเคชันบน WINDOWS	27
รูปที่ 4.1	แสดงโครงสร้างของ Sound Blaster Card	28
รูปที่ 5.1	ขั้นตอนการทำงานส่วนเรียนรู้	29
รูปที่ 5.2	โมดูลที่ใช้อ่าน input	31
รูปที่ 5.3	ขั้นตอนส่วนประมวลผล	32
รูปที่ 5.4	ลักษณะข้อมูลและการหาขอบเขต	33
รูปที่ 5.5	โมดูลที่ใช้ในการหาขอบเขต	36
รูปที่ 5.6	โมดูลการหา median	37
รูปที่ 5.7	โมดูลการหา laplacian	39
รูปที่ 6.1	แสดงกราฟที่ได้จากตารางที่ 6.1	41
รูปที่ 6.2	แสดงกราฟที่ได้จากตารางที่ 6.2	42
รูปที่ 6.3	แสดงกราฟที่ได้จากตารางที่ 6.3	43
รูปที่ 6.4	แสดงกราฟที่ได้จากตารางที่ 6.4	44
รูปที่ 6.5	แสดงกราฟที่ได้จากตารางที่ 6.5	45
รูปที่ 6.6	แสดงกราฟที่ได้จากตารางที่ 6.6	46
รูปที่ 6.7	แสดงกราฟที่ได้จากตารางที่ 6.7	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.8	แสดงกราฟที่ได้จากตารางที่ 6.8	48
รูปที่ 6.9	แสดงกราฟที่ได้จากตารางที่ 6.9	49
รูปที่ 6.10	แสดงกราฟที่ได้จากตารางที่ 6.10	50
รูปที่ 6.11	แสดงกราฟที่ได้จากตารางที่ 6.11	51
รูปที่ 6.12	แสดงกราฟที่ได้จากตารางที่ 6.12	52
รูปที่ 6.13	แสดงกราฟที่ได้จากตารางที่ 6.13	53
รูปที่ 6.14	แสดงกราฟที่ได้จากตารางที่ 6.14	54



สารบัญตาราง

ตาราง		หน้า
ตารางที่ 6.1	แสดงจำนวนแถวที่มีข้อมูลของแต่ละเสียง	41
ตารางที่ 6.2	แสดงจำนวนจุดที่มีค่า laplacian มากกว่า 200	42
ตารางที่ 6.3	แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง -1 และ 101	43
ตารางที่ 6.4	แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 100 และ 200	44
ตารางที่ 6.5	แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 100 และ 245	45
ตารางที่ 6.6	แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 200 และ 245	46
ตารางที่ 6.7	แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 0 และ 200	47
ตารางที่ 6.8	แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 0 และ 30	48
ตารางที่ 6.9	แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 0 และ 20	49
ตารางที่ 6.10	แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 0 และ 10	50
ตารางที่ 6.11	แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 5 และ 30	51
ตารางที่ 6.12	แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 200 และ 253	52
ตารางที่ 6.13	แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 244 และ 253	53
ตารางที่ 6.14	แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 100 และ 2536	54

บทที่ 2

ทฤษฎีและหลักการ

หลักการออกเสียง

การเปล่งเสียงของมนุษย์ ด้วยอวัยวะที่ใช้ในการออกเสียง(Organ of Speech) โดยเปล่งเสียงตามที่ระบบภาษาที่ใช้อยู่ แม้ว่ามนุษย์ที่อยู่ในสังคมเดียวกัน ใช้ภาษาเดียวกันก็ตาม ถ้าพิจารณาการเปล่งเสียงออกมาในแต่ละครั้งจริงๆ ก็จะสังเกตเห็นลักษณะที่แตกต่างกันได้ เราจึงสามารถจดจำเสียงผู้ที่เรารู้จักได้ ในทางวิทยาศาสตร์ถึงแม้จะใช้ภาษาแตกต่างกันก็สามารถแบ่งลักษณะการออกเสียงพูดของมนุษย์เป็นดังนี้

1. ความก้องหรือไม่ก้องของเสียง(Voiced & Voiceless)

อวัยวะส่วนที่ทำให้เกิดความก้องหรือไม่ก้องของเสียง คือ เส้นเสียง(Vocal Cords) เสียงที่ก้องเกิดจากการออกเสียงในขณะที่เส้นเสียงปิด ส่วนเสียงที่ไม่ก้องเกิดจากการออกเสียงเมื่อเส้นเสียงเปิด

2. ความยาวในการออกเสียง (Lenght) คือ ช่วงจังหวะเวลาในการออกเสียง ซึ่งขึ้น

กับเสียงสระในคำคำนั้นด้วย

3. ระดับเสียงสูงต่ำ (Pitch Of Voice)

ระดับสูงต่ำของเสียง อยู่ที่ความถี่ของเสียง เสียงแหลมเป็นเสียงที่มีความถี่สูง เสียงทุ้มคือเสียงที่มีระดับความถี่ต่ำ

4. ความดังของเสียง (Loundness)

ขึ้นอยู่กับปริมาณของลมที่เปล่งออกมา และการลงน้ำหนักของคำ ประกอบกับความยาวของเสียง

5. รอยต่อของเสียง (Juncture)

เสียงที่เรียงกันมาในคำพูดของนั้นมีรอยต่อระหว่างเสียงไม่เหมือนกัน บางเสียงก็อยู่ชิดกันซึ่งเป็นการยากที่จะทำการตัดแยกคำหรือพยางค์ ออกจากกัน ส่วนเสียงที่มีช่วงต่อห่างกันมากก็จะทำให้สามารถตัดเสียงออกเป็นคำ หรือ พยางค์ได้ง่าย

6. การลงน้ำหนักของเสียง

ความแรงที่ใช้ในการเปล่งเสียงของพยางค์แต่ละพยางค์นั้นจะต้องมีการเน้น ให้ดังหรือค่อยกว่าพยางค์ที่อยู่ข้างเคียง

ลำดับขั้นของการเกิดเสียง แบ่งได้เป็น 3 ขั้นตอนใหญ่ ๆ คือ

ขั้นตอนที่ 1

จุดเริ่มต้น (initiation) อวัยวะที่ใช้ในขั้นตอนนี้ คือ ปอดเป็นขั้นตอนที่ลมเริ่มถูกขับออกจากปอด ผ่านเข้าสู่ขั้นตอนที่ 2

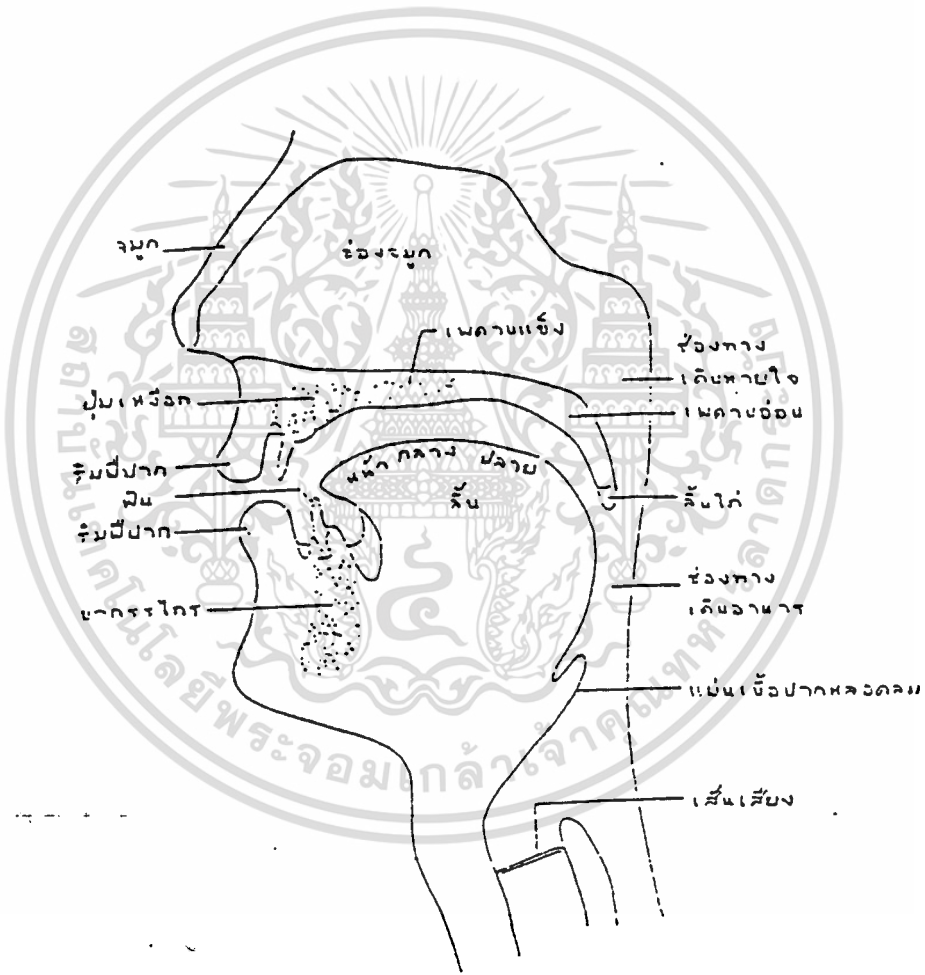
ขั้นตอนที่ 2

การดัดแปลงลมที่เส้นเสียง(Phonation) อวัยวะที่ใช้ในขั้นตอนนี้คือส่วนที่ต่อจากปอดขึ้นมาจนถึงกล่องเสียง เป็นขั้นตอนที่ลมจากปอดจะผ่านมายังหลอดลมและกล่องเสียง ซึ่ง ณ ที่กล่องเสียงนี้เส้นเสียงจะทำหน้าที่เป็นลิ้นเปิดปิดทำให้เกิดเสียงได้ 2 ชนิดคือ ถ้าเส้นเสียงเปิดตลอดเวลาที่ลมผ่าน ลมจะผ่านออกมาได้สะดวกซึ่งจะทำให้เกิดเสียงชนิดไม่ก้อง แต่ถ้าเส้นเสียงปิดกั้นลมไว้ ลมที่ผ่านออกมาจะเพิ่มแรงดันมากขึ้นจนเส้นเสียงเปิดและปิดสลับกันไป ทำให้เกิดเสียงก้องและเรียกความถี่ในการเปิดปิดของเส้นเสียงว่า ความถี่พื้นฐาน(Fundamental Frequency)

ขั้นตอนที่ 3

การเปลี่ยนแปลงลักษณะเสียง(Articulation) อวัยวะที่ใช้ในขั้นตอนนี้คือ ส่วนที่ต่อจากกล่องเสียงจนถึงริมฝีปาก และช่องว่างใหญ่อีกสองส่วนคือ ช่องปากและช่องจมูก ขั้นตอนนี้ลมผ่านออกมาจากกล่องเสียงจะถูกแปลงให้เกิดเสียงในลักษณะต่างๆ

การเปลี่ยนแปลงของลักษณะเสียงต่างๆ นั้น จะเกิดจากอวัยวะในช่องปากและช่องจมูกเป็นส่วนใหญ่ ทั้งนี้เพราะว่าภายในช่องว่างในปากจะมีการเปลี่ยนแปลงขนาด ตลอดเวลาที่เปลี่ยนแปลงเสียงและช่องจมูกก็จะถูกควบคุมโดยเพดานอ่อน (Velum) ดังนั้นถ้าช่องว่างๆเหล่านี้เปลี่ยนแปลงขนาดไป ก็ทำให้ความถี่กำทอน (Resonant Frequency) ของเสียงเปลี่ยนไปด้วย



รูปที่ 2.0 แสดงอวัยวะที่โตในการออกเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อวัยวะที่ใช้ในการออกเสียง

เสียงที่ออกมาที่ได้ยินเป็นคำๆนั้น กำเนิดมาจากหลอดเสียง(Vocal Tube) โดยปกติแล้วจะมีความยาวของหลอดเสียงประมาณ 17 ซม. มีพื้นที่หน้าตัดไม่แน่นอนมีขนาดตั้งแต่ 1 ถึง 20 ตารางเซนติเมตร เสียงสามารถออกมาได้ 2 ทางคือ ทางจมูก และทางปาก ซึ่งจะต้องผ่านอวัยวะที่ใช้ในการออกเสียงมีดังนี้

1. ริมฝีปาก เป็นอวัยวะส่วนที่เคลื่อนไหวได้มากในการออกเสียงพูดและมีอิทธิพลต่อการออกเสียงพูด
2. ฟัน เป็นอวัยวะที่ช่วยในการออกเสียง พยัญชนะ ส ซึ่งจะเป็นตัวกักลมที่จะออกจากปากทำให้เสียงที่ออกมานั้นเปลี่ยนไป
3. ปุ่มเหงือก เป็นส่วนที่นูนออกมาอยู่หลังฟันด้านบน จะทำให้เกิดเสียงที่เรียกว่า Alveolar Sound
4. เพดานแข็ง คือ เป็นส่วนของเนื้อที่ติดอยู่กับกระดูกแข็ง ด้านบน
5. เพดานอ่อน คือ ส่วนที่อยู่ถัดจากเพดานแข็งเข้าไปด้านใน เป็นกระดูกอ่อนที่ขยับขึ้นลงได้

หลักการในการจดจำเสียง

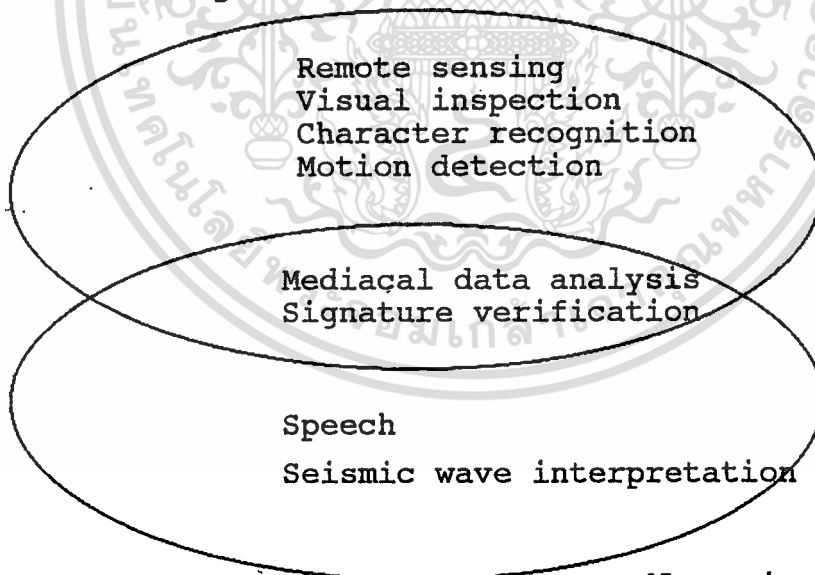
ในส่วนนี้จะกล่าวถึงการจดจำแบบอัตโนมัติ โดยนำเอาเทคโนโลยีทางด้านคอมพิวเตอร์มาใช้วิเคราะห์ คุณสมบัติทางกายภาพของสัญญาณ ซึ่งสามารถนำไปใช้ได้ทั้งกับสัญญาณที่มองเห็นได้ เช่น สัญญาณภาพ หรือ สัญญาณที่ไม่สามารถมองเห็นได้ เช่น สัญญาณเสียง วิธีการต่างๆได้ถูกพัฒนาขึ้น โดยพยายามที่จะเลียนแบบ ความสามารถในการแยกแยะเสียงต่างๆของมนุษย์ ซึ่งมีความซับซ้อนและละเอียดอ่อน ยากเกินกว่าที่จะเข้าใจขบวนการต่างๆในการได้ยินของมนุษย์ จึงเป็นความหวังที่พยายามจะพัฒนาวิธีการในการแยกแยะเสียงให้ใกล้เคียงกับมนุษย์ และมีเปอร์เซ็นต์ของความผิดพลาดลดลง วิธีการที่จะใช้ในการจดจำเสียงดังที่ได้กล่าวมาแล้วนั้นจะใช้หลักการของ Pattern recognition และทฤษฎีทางด้าน Image Processing เข้ามาช่วย ซึ่งจะได้กล่าวถึงในส่วนต่อไป

ทฤษฎีและหลักการที่ใช้

ในโครงการนี้ได้ใช้หลักการของทางด้าน Pattern-recognition ซึ่งในปัจจุบันมีการใช้กันในด้านต่าง ๆ มากมายหลายด้าน เช่นในด้านของ Image processing มีการใช้งานในการสื่อสารทางไกล (telecommunication) และในเรื่องของระบบสารสนเทศ (information systems) เป็นต้น ประโยชน์ในด้านการสื่อสารและระบบสารสนเทศนั้น เช่น การจัดเก็บและส่งออกสารทางแฟกซ์ มีการพัฒนาเทคนิคที่ใช้ในการ recognize pattern โดยแทนที่จะกระทำกับข้อมูลทั้งหมดในการส่งและจัดเก็บ ก็มีการเข้ารหัสที่แทน pattern ของข้อมูลนั้นได้ทำให้ปริมาณข้อมูลที่ต้องจัดการข้อมูลมีน้อยลงเป็นผลทำให้ประหยัดเวลาในการส่งและเนื้อที่ในการจัดเก็บ

นอกจากนี้เทคนิคของ pattern-recognition ยังสามารถนำมาใช้ในการจดจำ (recognize) pattern ต่าง ๆ เช่น ตัวอักษร สัญลักษณ์ ภาพ เป็นต้น ซึ่งจะเห็นได้ว่ารูปแบบของ pattern นั้นมีมากมายหลายรูปแบบ ขึ้นอยู่กับการนำไปประยุกต์ใช้งาน ดังเช่นที่ในโครงการนี้ได้นำหลักการของ pattern-recognition มาใช้กับสัญญาณเสียงซึ่งเป็นการประยุกต์ใช้แบบหนึ่ง

Visual signals



Non-visual signals

รูปที่ 2.1 แสดงสาขาที่มีการศึกษาการใช้งาน pattern recognition

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.1 จะแสดงให้เห็นถึงการประยุกต์ใช้งาน pattern-recognition ในด้านต่าง ๆ จะเห็นได้ว่าสัญญาณเสียงจัดเป็นรูปแบบของสัญญาณที่ไม่อาจมองเห็นได้ แต่สามารถรับรู้และหาตัวแทนของสัญญาณนั้นได้โดยการผ่านอุปกรณ์ในการรับสัญญาณ เช่น ไมโครโฟน แล้วผ่านวงจรฮาร์ดแวร์ที่ใช้ในการแปลงสัญญาณที่รับได้ซึ่งเป็นสัญญาณอนาล็อก (สัญญาณที่มีความต่อเนื่อง) ให้เป็นสัญญาณที่คอมพิวเตอร์เข้าใจได้คือสัญญาณดิจิทัล ซึ่งเป็นตัวแทนของสัญญาณเสียงนั้นสำหรับใช้ในการประมวลผลต่อไป

ส่วนต่อไปจะพูดถึงเทคนิคโดยทั่วไปที่มีใช้กันในด้าน pattern-recognition ซึ่งเทคนิคแต่ละแบบก็เหมาะกับ input บางลักษณะ ดังนั้นในการจะเลือกใช้เทคนิคใดนั้นจึงต้องศึกษาถึงวิธีการและคุณสมบัติของแต่ละเทคนิค จึงจะสามารถนำไปใช้งานได้ถูกต้อง ดังจะได้เสนอในส่วนตัวต่อไป

เทคนิคทั่วไปของ pattern-recognition

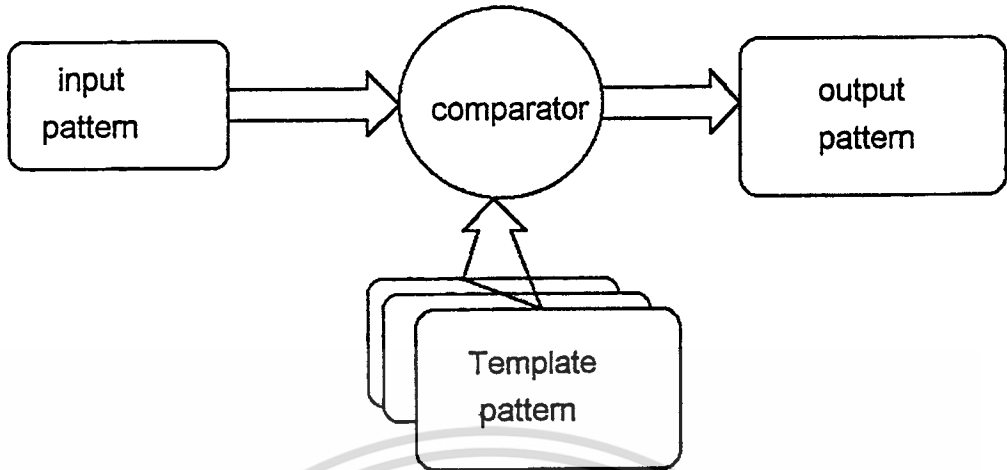
มีเทคนิคพื้นฐานหลายเทคนิคซึ่งใช้ได้เกือบทุกปัญหาของ pattern-recognition ไม่ว่าจะเป็นสัญญาณที่มองเห็นได้เช่นภาพ (image) หรือสัญญาณประเภท 1 มิติซึ่งเปลี่ยนแปลงตามเวลา (time-varying) เช่นเสียง (speech) เป็นต้น

เทคนิคพื้นฐานต่าง ๆ เหล่านั้น มีดังนี้

2.1 Template matching

เทคนิคนี้จัดเป็นเทคนิคที่ง่ายที่สุดของเทคนิคทางด้าน pattern-recognition การแยกแยะ pattern ทำได้โดยการนำ pattern ที่จะตรวจมาเปรียบเทียบกับ pattern ตัวอย่างที่เก็บไว้ pattern ตัวอย่างที่เก็บไว้นี้เรียกว่าเป็น template วิธีการของเทคนิคนี้แสดงไว้ในรูปที่ 2.2 ซึ่งแสดงให้เห็นถึงการนำ input pattern ที่ต้องการทราบผ่านไปยังตัวเปรียบเทียบ หรือ comparator ซึ่งจะทำการวัดความคล้ายคลึงระหว่าง input pattern นั้น กับแต่ละชุดของ template ที่เก็บไว้ ถ้าผลการเปรียบเทียบพบว่ามีผลของการจับคู่ (match) ระหว่าง input pattern กับ template เกินค่าระดับที่ตั้งไว้ (threshold) ค่าหนึ่งจะถือว่า input pattern นั้นสามารถ recognized ได้ ถ้าไม่มีผลของการจับคู่ใด ๆ เกินค่าระดับที่ตั้งไว้ก็จะถือว่า pattern ไม่มีความคล้ายคลึงกับ template ใด ๆ ที่มีอยู่เลย หรือหมายถึงเป็น pattern ที่ไม่เคยรู้จักนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดง Block Diagram ของการทำ Template Matching กำหนดให้

P เป็น input pattern และ

R เป็น template pattern

โดยที่

$P = \{ p(x_1, y_1), p(x_2, y_2), \dots, p(x_n, y_n) \}$ และ

$R = \{ r(s_1, t_1), r(s_2, t_2), \dots, r(s_n, t_n) \}$ มีมิติเดียวกันกับ P

วิธีที่ใช้ในการเปรียบเทียบความเหมือนหรือแตกต่างระหว่าง pattern มีดังนี้ คือ

2.1.1 Euclidean distance

มีสมการ คือ

$$D_E(P, R) = \sum_n \sqrt{(x_n - s_n)^2 + (y_n - t_n)^2}$$

จากสมการของ Euclidean distance คือการหาค่าผลต่างของแต่ละจุดยกกำลังสอง แล้วจึงหารากที่สอง (square root) นั้นเอง

โดยมากแล้วจะนำเครื่องหมายยกกำลังสอง และเครื่องหมาย square root ออก แล้วหาเพียงขนาดของความแตกต่างระหว่างจุดแทน จึงได้เป็นวิธีที่สองคือ 'city block' หรือ D4 distance

2.1.2 D4 distance or 'city-block'

มีสมการ คือ

$$D_c(P,R) = \sum_n (|x_n - s_n| + |y_n - t_n|)$$

จากสมการของ D4 distance จะเห็นว่าเป็นการหาค่าผลบวกของขนาดความแตกต่างของแต่ละจุด ซึ่งถ้า 2 pattern นั้นจับคู่ (match) กันได้พอดีก็จะได้อาผลบวกเท่ากับ 0

2.1.3 Hamming distance

มีสมการ คือ

$$D_H(P,R) = \sum_n e_n$$

โดยที่

$$\begin{aligned} e_n &= 1 \text{ if } p(x_n, y_n) \neq r(s_n, t_n) \\ &= 0 \text{ if } p(x_n, y_n) = r(s_n, t_n) \end{aligned}$$

จากสมการของ Hamming distance หมายความว่า ถ้า จุดใด ๆ ของ pattern ที่นำมาเปรียบเทียบกันมีค่าเท่ากัน จะได้ e (error) เป็น 0 แต่ถ้าไม่เท่ากันจะได้ค่า e เป็น 1 ดังนั้นถ้า 2 pattern ใด ๆ จับคู่ (match) กันได้พอดี 100% ก็จะได้ค่า summation ตามที่แสดงในสมการเป็น 0 สรุปจากวิธีการหาค่า distance ของทั้ง 3 วิธีข้างต้นนั้นได้ว่าถ้า pattern P และ pattern R เหมือนกัน (identical) พอดีจะได้ค่า $D_e(P,R) = D_c(P,R) = D_H(P,R) = 0$

ในการวัดค่า distance หรือ ความเหมือนด้วยวิธีการทั้ง 3 ข้างต้นนั้น จะได้ผลที่ถูกต้องก็ต่อเมื่อ template pattern และ input pattern นั้นมีขนาดเท่ากันทุกมิติ เช่น pattern แบบ 2 มิติก็ต้องมีขนาดเท่ากันทั้งในแกน x และ y ซึ่งในโลกความเป็นจริงแล้วกรณีดังกล่าวมีน้อยมาก และ จะต้องมีการผ่านกระบวนการ normalization เพื่อให้ได้ขนาดของ input pattern นั้นเท่ากับขนาดของ template pattern นอกจากนี้ยังอาจต้องมีการผ่านกระบวนการ preprocessing เพื่อที่จะลด quantization และ threshold noise ก่อนที่จะนำไปทำ template matching ต่อไป

ลด quantization และ threshold noise ก่อนที่จะนำไปทำ template matching ต่อไป

วิธีการของ template matching จะให้ได้ผลดีในกรณีดังนี้

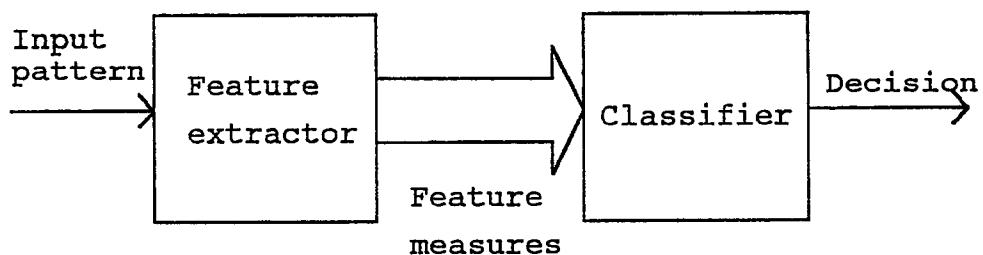
1. pattern ที่แตกต่างกันมีจำนวนน้อย
2. มีความแตกต่างกันเพียงเล็กน้อยภายใน คลาสเดียวกันของ pattern หรือ input ที่เข้ามานั้นควรจะคล้ายกับ template ที่มีอยู่
3. template ที่มีอยู่ของ pattern แต่ละแบบนั้นควรจะต้องแตกต่างกันพอสมควร เพื่อที่จะได้ตัดสินใจได้ว่า input pattern นั้นตรงกับ template pattern ตัวใดแน่

นอกจากเงื่อนไขต่าง ๆ ทั้ง 3 ข้อข้างต้นของ template matching แล้วยังมีสิ่งต้องคำนึงถึงอีก เช่นในเรื่องของขนาดนอกจากจะต้องเท่ากันแล้วยังต้องคำนึงถึงจุดระหว่าง pattern ที่เปรียบเทียบกัน ให้ตรงกันด้วย เพราะฉะนั้นในกรณีที่มีการ normalize เพื่อปรับขนาดให้เท่ากันแล้วยังต้องพยายามปรับตำแหน่งของจุดให้ตรงกันด้วย

จะเห็นว่าข้อจำกัดในการใช้งานมาก จึงเหมาะสมสำหรับปัญหาของ pattern-recognition แบบง่าย ๆ ซึ่ง input มีความแตกต่างจาก template ที่เก็บไว้ไม่มากนัก ในกรณีที่ เป็นปัญหาที่ซับซ้อนจึงต้องใช้ วิธีของ statistical หรือวิธีของ structural techniques เข้ามาช่วย

2.2 Statistical methods

Statistical pattern-recognition techniques หรือ decision-theoretic techniques หรือ เทคนิคทางสถิติ ใช้หลักการจัดว่า input ที่ต้องการตรวจสอบนั้นจัดอยู่ในคลาสใด โดยดูจาก ลักษณะต่าง ๆ ที่ดึงออกมาได้จาก pattern ในรูปที่ 2.3 จะแสดงให้เห็นถึงองค์ประกอบที่สำคัญของตัว recognizer ที่ใช้วิธีทางสถิติ โดยส่วนที่สำคัญจะมีอยู่ 2 ส่วน คือส่วนที่ใช้ดึงลักษณะต่าง ๆ หรือ Feature extractor และ ส่วนที่ใช้จัดจำแนก หรือ pattern classifier



รูปที่ 2.3 แสดงให้เห็นถึง Statistical pattern recognizer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

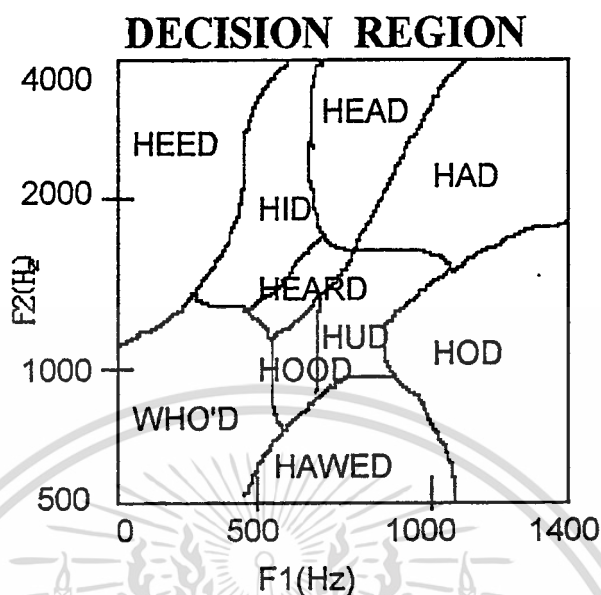
หลักการทำงานนี้คือ input จะถูกผ่านไปยัง Feature extractor ซึ่งจะทำการดึงลักษณะสำคัญที่เป็นลักษณะเด่น ของ pattern นั้นออกมาแล้วจะผ่านสิ่งถัดมา เรียกว่า feature vector ไปยังส่วน Classifier ซึ่งจะทำหน้าที่ในการตัดสินใจว่า pattern ที่ส่งเข้ามานั้นจัดอยู่ในคลาสใด

บ่อยครั้งที่วิธีการทางสถิติต่าง ๆ ที่ใช้อาจคล้ายกับเทคนิคของ template matching ที่ได้กล่าวไปแล้วในส่วนก่อน แต่ วิธีการสถิตินี้สามารถลดแนวโน้มของปัญหาจาก noise และความแตกต่างของ pattern (pattern variation) ซึ่งทำให้ไม่สามารถจับคู่ (match) กันได้หมดทั้ง pattern โดยใช้ชุดของลักษณะต่าง ๆ ที่ยืดหยุ่นต่อความแตกต่างของ pattern ดังนั้นวิธีการของ template matching ที่ได้กล่าวไปแล้วจึงจัดได้ว่าเป็นกรณีพิเศษของ statistical pattern matching

การเลือกลักษณะต่าง ๆ ออกมาเป็น feature vector นั้นเป็นเรื่องที่ค่อนข้างยาก และไม่มีกฎเกณฑ์ตายตัวที่จะใช้บอกว่าลักษณะใดบ้างที่ดึงออกมาได้ แต่โดยทั่วไปแล้วควรจะเลือกดึงลักษณะที่มีความอ่อนไหว (sensitive) ต่อผันแปรของ pattern น้อย หรืออีกนัยหนึ่งคือการเลือกลักษณะที่ ความแตกต่างของ pattern ไม่มีผลกระทบมากนัก ลักษณะที่ดึงออกมานั้น อาจไม่จำเป็นต้องมีความหมายในทางกายภาพ แต่ควรสามารถใช้ลักษณะนั้นในการวัดและจัดจำแนกได้ดี

ลักษณะของ feature vector ที่ดึงออกมาได้ N ลักษณะ สำหรับแต่ละ pattern จะเรียกว่าเป็น N -dimensional feature space ซึ่งลักษณะการแบ่ง feature space ดังกล่าวแสดงไว้ในรูปที่ 2.4

ข้อมูลทางสถิติที่เก็บได้จากชุดของ pattern ที่ใช้ในการ training (คือ pattern ต่าง ๆ ที่ใช้สอนคอมพิวเตอร์ให้รู้จัก) จะใช้ในการกำหนดลักษณะและขอบเขตที่เหมาะสมระหว่างคลาสของ pattern ต่าง ๆ



รูปที่ 2.4 การแบ่ง feature space

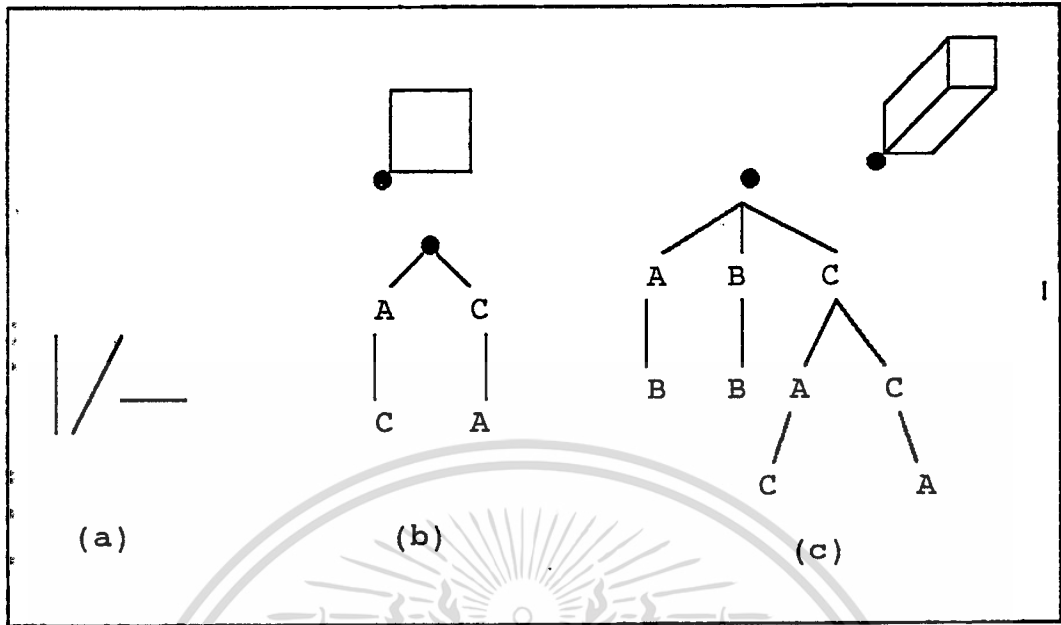
วิธีทางสถิตินี้จำเป็นต้องมีฐานข้อมูลของ pattern ที่ใช้ในการเรียนรู้จำนวนมาก และฐานข้อมูลของ pattern เหล่านั้นจะต้องสามารถกำหนดขอบเขตในการแบ่งลักษณะของคลาสของ pattern เหล่านั้นได้ จึงจะเหมาะแก่การใช้งาน

2.3 Structural or syntactic techniques

หลักการของวิธีนี้มาจากการที่มี pattern จำนวนมากมีลักษณะเป็นโครงสร้าง และสามารถแยกเป็นองค์ประกอบของ pattern ย่อย (subpatterns) หรือ pattern primitives ได้ กล่าวคือ pattern นั้นสามารถสร้างจาก ประโยคหรือแกรมมา (grammar) ของ pattern ง่าย ๆ (เช่น เส้นตรงหรือเส้นโค้ง) ใน pattern description language ได้ ซึ่งแสดงให้เห็นในรูปที่ 2.5

จากรูปที่ 2.5 ประกอบด้วย pattern อย่างง่าย 2 pattern คือรูปสี่เหลี่ยมและแท่งสี่เหลี่ยม (brick) สร้างขึ้นด้วยเส้นตรงพื้นฐาน 3 เส้น ซึ่งเป็น pattern primitives คือ A, B และ C การแสดง pattern เหล่านั้นในรูปของการเชื่อมต่อของเส้นตรงดังกล่าว อาศัย syntax ของ subpattern เช่น รูปสี่เหลี่ยมแสดงได้ด้วยประโยค (string) ของ pattern primitives 4 ตัว และแท่งสี่เหลี่ยมแสดงได้ด้วยประโยคของ pattern primitives 9 ตัว ดังรูป 2.5 (b) และ 2.5 (c) ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ตัวอย่างของ structural classification

(a) Pattern primitives. (b) Square. (c) Brick.

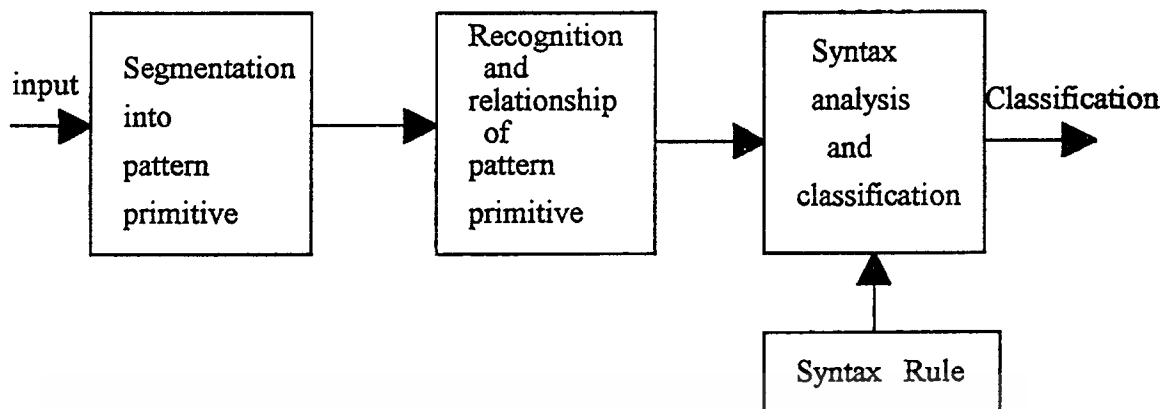
ลักษณะการทำงานของ Structural pattern recognizer แสดงไว้ในรูปที่ 2.6 เริ่มจากการนำ input pattern มาผ่าน segmentor ซึ่งจะทำการแยก pattern นั้นเป็นองค์ประกอบย่อยของ pattern primitives และ pattern primitives ที่แยกออกมาได้นี้จะถูกส่งต่อไปยังส่วน recognizer ซึ่งจะทำการจัดกลุ่ม และหาความสัมพันธ์ระหว่าง pattern primitives เหล่านั้น จากนั้นจะนำสิ่งที่ได้ทั้งหมดไปผ่าน syntax analyser ซึ่งจะทำการจัดกลุ่ม (classification) pattern input ที่ส่งเข้ามาโดยอาศัย syntax ของ pattern ที่เข้ามานั้นกับ ชุดของ syntax rules ต่าง ๆ ที่มีอยู่ซึ่งได้จาก pattern ที่รู้จักแล้ว

สิ่งที่ปัญหาสำคัญสำหรับวิธีการนี้คือ การหาสิ่งที่ เป็น pattern primitives ออกมา เนื่องจากไม่มีกฎแน่นอนที่ใช้ในการเลือก ดังนั้นการใช้วิธีนี้โดยมากจึงอาศัยประสบการณ์หรือความชอบของผู้ใช้เอง แต่อาจแนะนำแนวทางโดยทั่วไปในการเลือก primitive ได้ดังนี้

1. primitives จะต้องเป็นองค์ประกอบพื้นฐานของ pattern ที่มีขนาดกระทัดรัด (compact) แต่สามารถใช้อธิบายถึงข้อมูลนั้นในรูปของความสัมพันธ์ของโครงสร้างที่แน่นอนได้
2. primitives นั้นจะต้องถูกดึงออกมาได้โดยง่าย

จากกฎโดยทั่วไปนั้น ถ้าเลือก pattern primitives แบบง่าย จะทำให้ syntax grammars ยุ่งยากซับซ้อน และถ้าเลือก pattern primitives ซับซ้อนก็จะทำให้ syntax grammar ง่าย การเลือกแบบใดก็ขึ้นอยู่กับผู้ออกแบบเองที่จะต้องศึกษาถึงลักษณะของ pattern นั้น ๆ ก่อนว่าเหมาะสมจะใช้แบบใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 Structural recognition

จากวิธีการทั้งสามของ pattern-recognition ที่ได้กล่าวมาแล้วทั้งหมด มีหลักเกณฑ์ในการเลือกใช้ดังนี้

1. ถ้า pattern ที่แตกต่างกันมีจำนวนไม่มาก ควรเลือกใช้วิธี template matching
2. ถ้า pattern นั้นมีลักษณะเป็นโครงสร้างบ้างแต่ไม่มากนัก หรือ เป็นประเภทที่มีสัญญาณรบกวนมาก ควรเลือกใช้วิธี statistical methods
3. ถ้า pattern มีลักษณะเป็นโครงสร้างชัดเจน และมีความผันแปรเป็นจำนวนที่แน่นอน ควรเลือกใช้วิธีของ structural techniques

จากลักษณะของเทคนิคต่าง ๆ และ วิธีการของแต่ละเทคนิค ในโครงการนี้จึงเลือกใช้วิธีของ statistical method เนื่องจากมีลักษณะยืดหยุ่นต่อสัญญาณเสียงมากที่สุด กล่าวคือสัญญาณเสียงที่เข้ามาในแต่ละครั้งนั้นแม้จะเป็นคำ ๆ เดียวกัน แต่ก็มีความผันแปรของข้อมูลที่แปลงแล้วไม่เท่ากัน จึงไม่อาจใช้วิธีของ template matching และสัญญาณเสียงที่ได้นี้ก็ไม่อาจหาลักษณะของโครงสร้างที่แน่นอนได้ นอกจากนี้สัญญาณเสียงยังถูกรบกวนจากเสียงอื่นที่อยู่รอบ ๆ ได้โดยง่าย ดังนั้นการใช้วิธีของ statistical method จึงมีความยืดหยุ่นและเหมาะกับการใช้กับสัญญาณเสียงมากที่สุด

ข้อมูลที่ใช้เป็นค่าสถิติซึ่งดึงออกมาจากสัญญาณเสียงนี้อาจทำได้หลายวิธี แต่ในโครงการนี้ได้เลือกใช้ทฤษฎีทางด้าน image processing มาใช้ในการประมวลผลสัญญาณเสียงก่อนที่จะดึงลักษณะเด่นของแต่ละเสียงออกมาเพื่อใช้ในการจัดคลาสให้สัญญาณเสียงแต่ละเสียง

ในส่วนต่อไปจึงได้กล่าวถึงทฤษฎีโดยทั่วไป ทางด้าน image processing ที่มีใช้ใน
โครงการ ก่อนที่จะได้กล่าวถึงการนำทฤษฎีนั้นไปใช้งานจริงในส่วน ของการคำนวณและการสร้าง
ต่อไป

ทฤษฎีที่ใช้ในการ ประมวลผลสัญญาณ

1. Median Filtering

Median Filtering เป็นวิธีที่ช่วยลดสัญญาณรบกวน และ ปรับสภาพของสัญญาณให้
ดีขึ้นทำให้สัญญาณที่ได้นั้นใกล้เคียงความเป็นจริงมากขึ้น มีวิธีการคือ ทำการแบ่งข้อมูลออกเป็น
ชุดเล็ก ๆ

โดยจำนวนข้อมูลแต่ละชุดจะมีขนาดเท่าใดก็ขึ้นอยู่กับว่าจะทำ median ขนาดเท่าใด
เช่น ถ้าทำ median 3 จำนวนข้อมูลในแต่ละชุดก็จะมีจำนวน 3 ตัว หลักการแบ่งข้อมูลนั้นใช้การ
นับจำนวนโดยเริ่มจากตำแหน่งที่ต้องการหา median ให้ครบจำนวนที่จะทำ median เช่น ต้อง
การทำ median 3 จะทำได้ดังนี้

ตำแหน่งที่ 1 ของข้อมูลใหม่ได้จาก การนำข้อมูลเดิมตำแหน่งที่ 1,2,3 มาหา median
ตำแหน่งที่ 2 ของข้อมูลใหม่ได้จาก การนำข้อมูลเดิมตำแหน่งที่ 2,3,4 มาหา median
ตำแหน่งที่ 3 ของข้อมูลใหม่ได้จาก การนำข้อมูลเดิมตำแหน่งที่ 3,4,5 มาหา median

... ..
... ..
... ..

ทำเช่นนี้จนได้ข้อมูลครบทุกตำแหน่ง

หลักการหา median นั้นใช้หลักคณิตศาสตร์ทั่วไป คือ

- ทำการ sort ข้อมูลในชุดนั้นที่จะหา median ก่อน
- เลือกค่าข้อมูลที่อยู่ตรงกลางมาเป็นค่า median

เนื่องจาก median นั้นเลือกค่าที่อยู่ตรงกลางมาเป็นตัวแทน ดังนั้นการเลือกจำนวน
ข้อมูลที่จะทำ median จึงควรจะต้องเป็นจำนวนเลขคี่

ตัวอย่างเช่น การทำ median 3 ของสัญญาณขนาด 1 มิติ ทำได้ดังนี้

มีชุดข้อมูล คือ

0 0 0 6 0 0 1 2 1 0 1 2 1 0 2 4 2 0 2 4 2 0 0 1 2 3 4 4 4 4 0 0 0

|——|

|——|

|——|

|——|

จากข้อมูลดังกล่าวจะหา median 3 ของข้อมูลได้ คือ

0 0 0 0 0 1 1 1 1 1 1 1 2 2 2 2 2 2 0 0 1 2 3 4 4 4 4 0 0

ลักษณะของข้อมูลที่ได้ใหม่นี้จะเห็นว่า

- noise ถูกกำจัด คือ 6 ที่อยู่ในกลุ่มของ 0 ช่วงต้นจะถูกเปลี่ยนเป็น 0 ด้วย

- ลักษณะที่เป็น oscillations จะถูกกดให้แบนลง คือ ข้อมูลช่วงที่เป็น 0 1 2 1 0 1

2 1 0 ถูกแปลงเป็น 1 หหมด

- ยังคงลักษณะของ ramp และ step อยู่ คือข้อมูลช่วง 0 1 2 3 4 และ ช่วง 4 4 4

0 0 0

2. The Laplacian

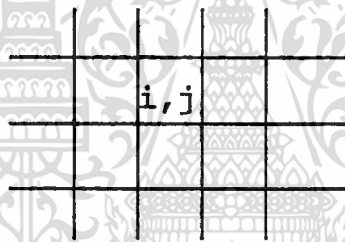
เป็นเทคนิคที่ใช้ในการเน้นขอบของสัญญาณ โดยมีหลักการว่า บริเวณขอบของสัญญาณคือ บริเวณที่มีการเปลี่ยนแปลงของสัญญาณมาก ซึ่งทฤษฎีของ laplacian นี้มีที่มาจากทฤษฎีของ Differentiation ซึ่งเป็นการหาค่าความแตกต่างระหว่างจุดใน pattern

- Differentiation ในกรณีของ pattern 2 มิติ การหา The first difference ในทิศทาง x และ y มีดังนี้ คือ

$$\Delta_x f(i,j) \equiv f(i,j) - f(i-1, j)$$

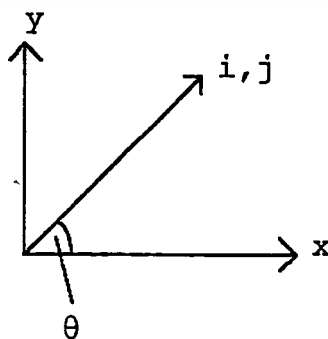
$$\Delta_y f(i,j) \equiv f(i,j) - f(i, j-1)$$

โดยที่ i, j เป็นตำแหน่งในแนวแกน x และ y ดังนี้



การหา first difference ในทิศทางใด ๆ สามารถหาได้จาก linear combination ของ x และ y ดังนี้

$$\Delta_\theta f(i,j) = \Delta_x f(i,j) \cos\theta + \Delta_y f(i,j) \sin\theta$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหา higher difference operator หาได้ โดยการทำ first difference ซ้ำอีกครั้ง เช่น second difference หาได้ดังนี้คือ

$$\begin{aligned} \Delta^2_x f(i,j) &= \Delta_x f(i+1,j) - \Delta_x f(i,j) \\ &= [f(i+1,j) - f(i,j)] - [f(i,j) - f(i-1,j)] \\ &= f(i+1,j) + f(i-1,j) - 2f(i,j) \end{aligned}$$

และทำนองเดียวกัน

$$\Delta^2_y f(i,j) = f(i,j+1) + f(i,j-1) - 2f(i,j)$$

- Laplacian

ค่าของ laplacian อาศัย ค่า difference ที่กล่าวมาแล้วในการหา ซึ่งหาได้ดังนี้คือ

$$\begin{aligned} \nabla^2 f(i,j) &\equiv \Delta^2_x f(i,j) + \Delta^2_y f(i,j) \\ &= [f(i+1,j) + f(i-1,j) + f(i,j+1) + f(i,j-1)] - 4f(i,j) \end{aligned}$$

จากสมการของ laplacian จะเห็นได้ว่าแท้จริงแล้วก็คือค่าผลบวกของ second difference ทั้ง 2 แกนนั่นเอง

เนื่องจากในโครงงานนี้ได้นำมาประยุกต์ใช้กับเสียง จึงคิดเพียงมิติเดียว ด้วยสมการเดียวกันกับ second difference ในแนวแกน x

ที่ได้กล่าวมาทั้งหมดนั้นเป็นทฤษฎีเท่าที่นำมาประยุกต์ใช้ในโครงงานนี้เท่านั้น ซึ่งในการประมวลผล pattern นั้น ยังมีอีกหลายวิธีซึ่ง แต่วิธีที่นำมาเสนอนี้เป็นวิธีที่ทำการทดลองแล้วพบว่าใช้งานได้และทำงานได้เร็วกว่าวิธีอื่น ๆ จึงนำเสนอแต่เพียงเท่านี้ และจะได้กล่าวถึงวิธีการนำไปประยุกต์ใช้งานในบทต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การเขียนโปรแกรมบน WINDOWS

จากการใช้งานระบบปฏิบัติการดอส (DOS) ในรุ่นแรกๆที่ใช้ FCB(File Control Block) ในการจัดการกับไฟล์ ดอสก็ได้รับการปรับปรุงเพิ่มความสามารถต่อมา ทำให้ผู้ใช้งานและโปรแกรมเมอร์ สามารถใช้งานดอสได้สะดวกสบายมากยิ่งขึ้น ไม่ว่าจะเป็นการจัดระบบไดเรคทอรี ไฟล์เฮนเดิล ฯลฯ แต่ที่ดอสยังขาดการสนับสนุนระบบกราฟิก และ ระบบการติดต่อกับผู้ใช้ จึงทำให้ระบบปฏิบัติการ WINDOWS ได้เข้ามามีบทบาทมากขึ้น และ ด้วยความสามารถที่มากขึ้นของระบบปฏิบัติการจึงทำให้การเขียนโปรแกรมนั้นง่ายขึ้น แต่ลักษณะการเขียนโปรแกรมบน WINDOWS นั้นจะแตกต่างไปจากการเขียนโปรแกรมบนดอส ความสามารถที่เพิ่มขึ้นมานบน WINDOWS มีดังนี้

- ระบบการติดต่อกับผู้ใช้แบบกราฟิก อันประกอบด้วย วินโดว์ เมนู กรอบข้อความ(dialog box) และ คอนโทรล(Controls) ชนิดต่างๆ
- ระบบรับข้อมูลที่สะดวกกว่าเดิม โดยรับเป็น เมสเสจ (Message)
- การจัดระบบงานแบบมัลติทาสกิง (multitasking)
- มีการใช้ทรัพยากรร่วมกันระหว่างแอปพลิเคชัน
- สนับสนุนการใช้งานอุปกรณ์ มัลติมีเดีย (Multimedia)

การเขียนโปรแกรมบนดอส ด้วยภาษา C ส่วนมากมักจะติดต่อกับ I/O และหน่วยความจำโดยตรงผ่านฟังก์ชันมาตรฐานในไลบรารีภาษา C ซึ่งจะทำเช่นนั้นบน WINDOWS ไม่ได้ เพราะ แอปพลิเคชันแต่ละแอปพลิเคชัน ต้องแบ่งกันใช้งานทรัพยากรบน WINDOWS อันได้แก่ คีย์บอร์ด หน่วยความจำ พอร์ตต่างๆ จอภาพ และ CPU ความแตกต่างระหว่างการเขียนโปรแกรมบนดอสกับบน WINDOWS ได้แก่

1. ระบบติดต่อกับผู้ใช้
2. ระบบการรับข้อมูลเข้า
3. ระบบกราฟิก
4. ระบบจัดงาน

1. ระบบการติดต่อกับผู้ใช้ ผู้ใช้สามารถเปิดใช้งานหลายๆแอปพลิเคชันได้พร้อมๆกัน โดยแต่ละแอปพลิเคชันจะมีกรอบสี่เหลี่ยมที่เรียกว่า window เป็นของตัวเองและจะแสดงผลเฉพาะภายใน window ของตัวเองเท่านั้น เสมือนกับเป็น จอภาพหนึ่งจอสำหรับ แอปพลิเคชันนั้นเลย สามารถเห็นการทำงานของแอปพลิเคชันได้พร้อมๆกัน และ สามารถปรับขนาด เคลื่อนย้าย และ ลดขนาดของ window ได้ตามต้องการ ภายใน window จะประกอบด้วยส่วนต่างๆ เช่น เมนู ปุ่มบังคับ สโคลลบาร์ (scroll bar) และ กรอบข้อความ (dialog box) เป็นต้น แอปพลิเคชันที่ทำงานบน WINDOWS จะต้องสร้าง window ขึ้นมาเอง

2. ระบบการรับข้อมูลเข้า ในระบบดอสรรมดาเมื่อแอปพลิเคชันจะรับข้อมูลจากผู้ใช้ ตัวแอปพลิเคชันจะต้องมีฟังก์ชันที่รอจนกว่าผู้ใช้จะใส่ข้อมูลเข้ามาเสียก่อน แต่ใน WINDOWS จะรับข้อมูลเข้ามาไว้ใน เมสเสจคิว(Messsage Queue) เมื่อแอปพลิเคชันต้องการอ่านข้อมูลก็เพียงแต่อ่านจากเมสเสจคิวเท่านั้น แอปพลิเคชันจะได้รับข้อมูลจากผู้ใช้โดยไม่มีการสูญเสียของข้อมูล โดยที่แอปพลิเคชันไม่ต้องมารอรับข้อมูลเอง

ข้อมูลที่อยู่ในเมสเสจคิวจะอยู่ในรูปของเมสเสจที่มีขนาดในการเก็บเท่ากันและมีรูปแบบการใช้งานคล้ายกัน ไม่ว่าจะเป็น เมสเสจจากคีย์บอร์ด หรือ เมาส์ ในการอ่านค่าแต่ละครั้ง จะใช้โปรแกรมที่คล้ายกันทำให้การเขียนแอปพลิเคชันง่ายขึ้น โดยใช้ฟังก์ชันเดียวเท่านั้น คือ GetMessage()

3. ระบบกราฟิก จากระบบดอสที่ไม่มีการสนับสนุนระบบกราฟิกเลย มาสู่ WINDOWS ที่มีคำสั่งมากมายและมีประสิทธิภาพ สำหรับการจัดการกับระบบกราฟิกไม่ว่าจะเป็นการวาด เส้นตรง วงกลม เส้นโค้ง หรือรูปภาพบิตแมพโดยไม่ขึ้นกับอุปกรณ์แสดงผลไม่ว่าจะเป็น จอภาพ เครื่องพิมพ์ พลอตเตอร์ ฯลฯ เพราะ WINDOWS ได้อาศัยดีไวซ์ไดรเวอร์เพื่อติดต่อกับอุปกรณ์แสดงผลให้เหมาะกับแต่ละแบบ

4. ระบบจัดงาน การทำงานในดอสนั้นจะไม่มีระบบจัดการปันส่วนทรัพยากรใดๆ เมื่อแอปพลิเคชันหนึ่งๆ ทำงานแอปพลิเคชันนั้นจะเป็นยึดครองทรัพยากรทั้งหมดเอาไว้ อันได้แก่ คีย์บอร์ด จอภาพ หน่วยความจำ หรือแม้กระทั่ง CPU แต่สำหรับ WINDOWS จะมีการปันส่วนกันใช้ทรัพยากร

ใน WINDOWS ซึ่งเป็นระบบมัลติทาสกิง(Multitasking) จะมีการแบ่งปันกันใช้หน่วยความจำซึ่งมีอยู่จำกัด และมีหลายๆ แอปพลิเคชันใช้งานร่วมกันอยู่ทำให้หน่วยความ จำไม่พอ จึงต้องมีการเคลื่อนย้าย ส่วนต่างๆของหน่วยความจำอยู่ตลอดเวลาเพื่อให้การทำงานของระบบมีประสิทธิภาพสูงสุด จะพบว่าตำแหน่งของแอปพลิเคชันจะไม่อยู่ที่เดิม จึงเป็นเหตุผลที่

เอกส WINDOWS ไม่ยอมให้เข้าถึงหน่วยความจำโดยตรง แต่จะต้องทำผ่าน WINDOWS เสมอ ยชนด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

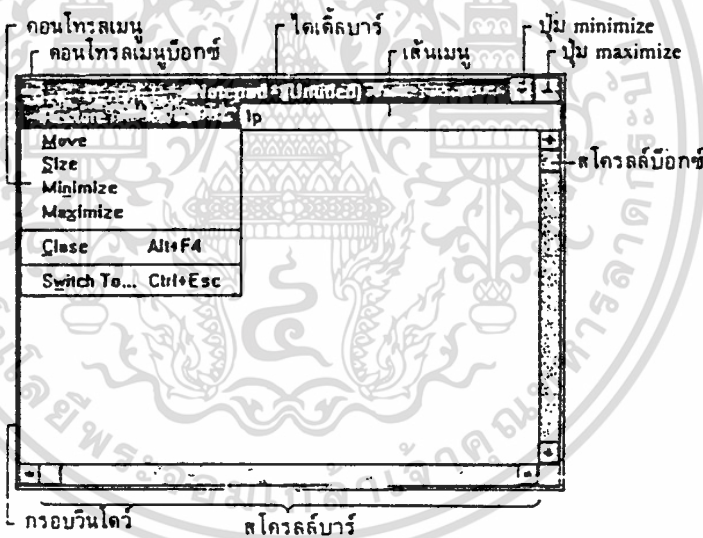
รูปแบบของแอปพลิเคชัน

แอปพลิเคชันที่สร้างขึ้นบน WINDOWS จะประกอบด้วยส่วนต่างๆต่อไปนี้

- วินโดว์
- เมนู
- กรอบข้อความ (dialog box)
- เมสเสจ (message)

1. วินโดว์ เป็นสิ่งที่ผู้ใช้จะติดต่อกับแอปพลิเคชัน ทุกแอปพลิเคชันจะต้องมีวินโดว์เสมอ

วินโดว์ประกอบไปด้วย ไตเติลบาร์ เมนูบาร์ สโคลล์บาร์ กรอบข้อความ ฯลฯ การสร้างวินโดว์ได้ โดยเรียกคำสั่ง CreateWindow () ซึ่งแอปพลิเคชันจะต้องเป็นตัวที่สร้างแต่การจัดการกับวินโดว์นั้น เป็นหน้าที่ของระบบ WINDOWS

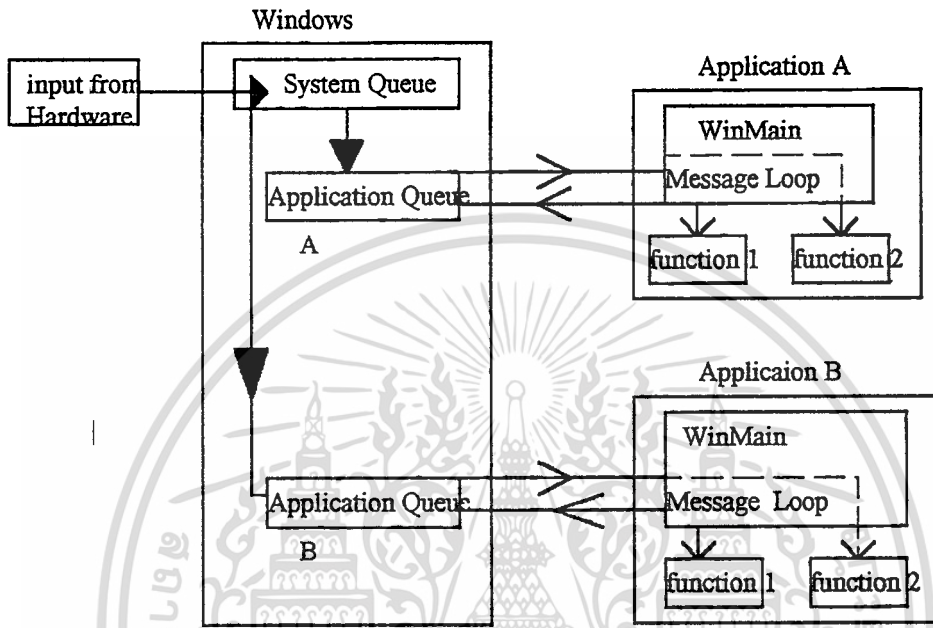


รูปที่ 3.1 แสดงส่วนประกอบต่างๆ ของวินโดว์

2. เมนู เป็นส่วนที่แสดงรายการคำสั่งให้ผู้ใช้เลือกสั่งงาน โดยคำสั่งที่เลือกจะถูกส่งผ่านทางเมสเสจคิว เพื่อให้กับแอปพลิเคชันนั้นๆ เมนูเป็น Resource อย่างหนึ่งที่ระบบ WINDOWS มีให้อยู่แล้ว

3. กรอบข้อความ (Dialog boxes) ใช้รับข้อความจากผู้ใช้ ภายในกรอบข้อความ จะมี Control ต่างๆให้กดเลือก และมีปุ่มสำหรับสั่งงาน

4. เมสเสจ การรับข้อมูลเข้าแอฟพลิเคชันจะรับผ่านทางเมสเสจคิว ข้อมูลในเมสเสจคิวจะถูกแจกจ่ายไปยังแอฟพลิเคชันที่เหมาะสม โดย เมสเสจลูป (Message Loop) ซึ่งจะดึงข้อมูลจากคิวระบบ (System Queue) ไปใส่ไว้ใน คิวของแต่ละแอฟพลิเคชัน (Application Queue)



รูปที่ 3.2 การรับข้อมูลผ่านทางเมสเสจคิวของสองแอฟพลิเคชัน

ไลบรารีของ WINDOWS

ฟังก์ชันต่างๆของ WINDOWS ได้ถูกเก็บรวบรวมไว้เป็น ไลบรารี ซึ่งจะต่างกับไลบรารีในภาษา C คือ ระบบ WINDOWS จะ Link ไลบรารี ก็ต่อเมื่อ มีการเรียกใช้งาน ฟังก์ชันในไลบรารีเท่านั้น จึงทำให้ขนาดของแอฟพลิเคชันเล็กลงและกินเนื้อที่ในหน่วยความจำน้อยกว่า เพราะสามารถใช้งานไลบรารีร่วมกันได้

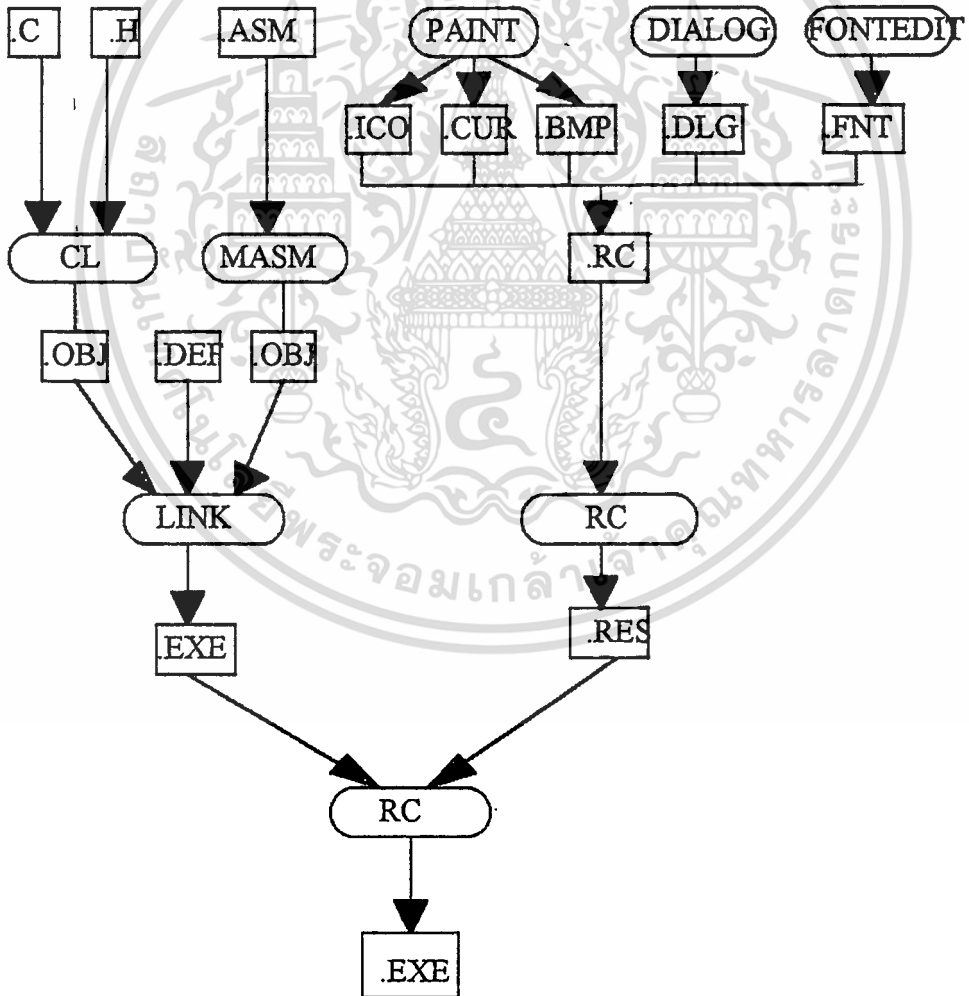
ไลบรารีที่สำคัญประจำ WINDOWS ได้แก่

1. USER.EXE เป็น ไลบรารีที่บรรจุฟังก์ชันในการจัดการวินโดว์และการจัดการสถานะแวดล้อมทั่วไปของ WINDOWS
2. KERNEL.EXE เก็บฟังก์ชันบริการ เกี่ยวกับการจัดระบบงาน การบริหารหน่วยความจำ และการจัดการทรัพยากร
3. GDI.EXE เก็บฟังก์ชัน การติดต่อกับอุปกรณ์แสดงผลกราฟิก

ขั้นตอนในการสร้างแอปพลิเคชัน

การสร้างแอปพลิเคชันบน WINDOWS ด้วยภาษา C มีขั้นตอนดังนี้

1. เขียนโปรแกรมภาษา C ที่ใช้ ฟังก์ชัน WinMain แทนฟังก์ชัน main ภายในประกอบด้วย Message Loop และฟังก์ชันในการสร้างวินโดว์
2. สร้าง resource ที่ต้องการใช้งาน
3. เขียน script ของ resource
4. เขียนไฟล์กำหนดลักษณะโมดูลของแอปพลิเคชัน (.DEF)
5. คอมไพล์โปรแกรมภาษา C แล้ว Link จะได้ โปรแกรม .EXE
6. ใช้ resource compiler คอมไพล์ resource script จะทำให้ resource รวมกับโปรแกรม .EXE เป็นแอปพลิเคชันที่พร้อมใช้งาน



รูปที่ 3.3 ขั้นตอนการสร้างแอปพลิเคชันบน WINDOWS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างทั่วไปของ Sound Blaster Card

Sound Capabilities

- มีระบบเสียง 12 ระบบ (12 C/MS)
- มี 11 เสียง สำหรับระบบ FM
- มีตัวแปลงสัญญาณจาก ดิจิตอลเป็น อะนาล็อก 1 ช่อง มีรายละเอียดดังนี้
 - มี Sampling Rate 4 kHz ถึง 23 kHz
 - Compressions schemes
 - 8 bit data no compression
 - 2 to 1 data compression 4 bit
 - 3 to 1 data compression 2.6 bit
 - 4 to 1 data compression 2 bit
- มีระบบ Stereo power amplifier กำลังขยาย 4 watt ต่อ 1 channel ระบบ output เป็น 4-Ohm
- RCA plug
- มีปุ่ม volume control

Voice Input (Digital Sampling) Capability

- 8 bit A/D conversion
- อัตราการ Sampling อยู่ระหว่าง 4 kHz ถึง 44.5 kHz
- ADC Transfer mode
 - Direct mode
 - DMA mode no CPU
- มีไมโครโฟน และระบบ amplifier พร้อมในตัว ซึ่งมีอัตราขยายแบบ auto

Joystick Port

- มี standard Game I/O port และมี standard IBM compatibility joystick

MIDI Interface

- มี MIDI Interface ในตัว สำหรับเชื่อมต่อกับเครื่อง MIDI หรือ Keyboard Hardware Data

ใน Sound Blaster จะใช้ I/O port เบอร์ระหว่าง 2x0H - 2xFH ซึ่ง x คือเบอร์ของ Jumper ระหว่าง 1 - 6

I/O address คือ 210H , 220H , 230H , 240H , 250H , 260H

FM music จะใช้ address ระหว่าง 388H - 389H

การติดตั้ง Sound Blaster Card

1. นำเอา Sound Blaster Card ไปเสียบลงใน slot ของเครื่องคอมพิวเตอร์
2. ต่อลำโพงเข้ากับ Sound Blaster
3. ปรับ Volume Control ของ Sound Blaster ให้อยู่ที่กึ่งกลาง
4. ใช้ File Test - SBC ที่มาพร้อมกับ Card ทดสอบ Sound Blaster โดยจะเช็คที่ I/O

Address , IRQ และ Feature ต่างๆของ Sound Blaster

5. ตั้ง Config ของ Sound Blaster ลงใน file Autoexec.bat ของ DOS

```
SET BLASTER = A220 I7 D1 T3
```

เป็นอันเสร็จสิ้นการติดตั้ง สามารถใช้งาน Sound Blaster ได้ทันที

บทที่ 5

การคำนวณและการสร้าง

สำหรับโครงงานนี้ จะประกอบด้วยส่วนที่สำคัญ 2 ส่วน คือส่วนที่เป็นฮาร์ดแวร์ (Hardware) และส่วนที่เป็น ซอฟต์แวร์ (Software) โดยส่วนที่เป็นฮาร์ดแวร์นี้จะเป็นส่วนที่ทำหน้าที่รับเสียงจากภายนอกเข้ามาโดยผ่านทางไมโครโฟน ซึ่งในโครงงานนี้จะตัดปัญหาทางด้านฮาร์ดแวร์ออกไปโดยการใช้ผลิตภัณฑ์จำพวกการ์ดเสียง (Sound Card) ซึ่งมีขายอยู่เป็นจำนวนมากหลายยี่ห้อในท้องตลาด แต่ในโครงงานนี้จะใช้การ์ดเสียง SOUND BLASTER ของบริษัท Creative Labs เวอร์ชัน 2.0

อีกส่วนหนึ่งที่เป็นซอฟต์แวร์นั้นจัดเป็นส่วนที่สำคัญของโครงงานนี้ จะมีลักษณะเป็นโปรแกรมที่ใช้สั่งงานการ์ดเสียงให้ทำการรับเสียงเข้ามาและนำเสียงที่รับได้ไปประมวลผลต่อไป

โดยในโครงงานในส่วนนี้มีวัตถุประสงค์ในการให้คอมพิวเตอร์เรียนรู้คำ เพื่อนำไปใช้กับหุ่นยนต์สำรวจได้ จึงเลือกสอนเฉพาะคำที่จำเป็นในการเคลื่อนย้ายหุ่นยนต์สำรวจนี้ 4 คำ ซึ่งจริง ๆ แล้วสามารถขยายให้มีการเรียนรู้จำนวนคำมากกว่านี้ได้ตามต้องการโดยใช้หลักการเดียวกัน

คำทั้งสี่ที่สอนให้คอมพิวเตอร์เรียนรู้ในโครงงานนี้ได้แก่ คำว่า

- "ซ้าย" หมายถึง ให้หันไปทางซ้าย
- "ขวา" หมายถึง ให้หันไปทางขวา
- "เดิน" หมายถึง ให้เคลื่อนที่ตรงไปข้างหน้า
- "หยุด" หมายถึง ให้หยุดอยู่กับที่

ซึ่งโปรแกรมนี้พัฒนาขึ้นมาโดยใช้ภาษา C และคอมไพเลอร์ของ บริษัท BORLAND คือ BORLANDC Version 3.1

ดังนั้นในส่วนที่จะกล่าวถึงต่อไปนี้เป็นขั้นตอนการสร้างส่วนโปรแกรมที่ใช้ประมวลผลเสียงเพื่อการรู้จัก ซึ่งมีรายละเอียดดังต่อไปนี้

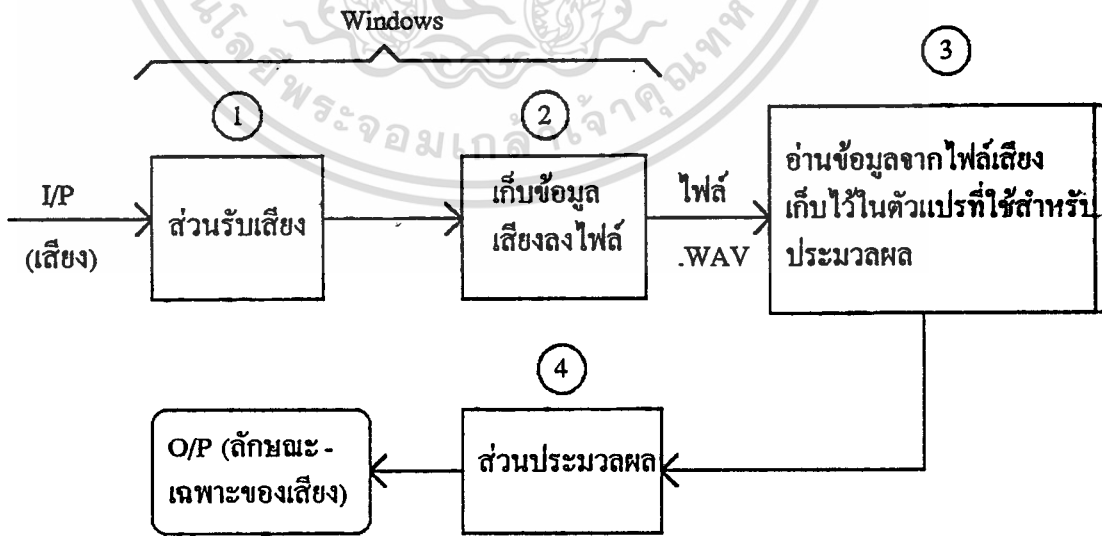
ขั้นตอนการรู้จำนี้อาจแบ่งออกได้เป็นขั้นตอนใหญ่ ๆ 2 ขั้นตอน คือ

1. การเรียนรู้ (หรือ การฝึก) เป็นการสอนให้คอมพิวเตอร์รู้จัก คำต่าง ๆ ที่ต้องการให้เรียนรู้
2. การรู้จำ เป็นขั้นตอนหลังจากผ่านการเรียนรู้แล้วโดยใช้ลักษณะเฉพาะต่าง ๆ ของคำที่ได้ขั้นตอนการเรียนรู้มาตรวจสอบคำที่รับเข้ามาใหม่ว่าเป็นคำที่ตรงกับคำที่เคยเรียนรู้หรือไม่

รายละเอียดสำหรับแต่ละขั้นตอนมีดังนี้

5.1 การเรียนรู้

มีขั้นตอนการทำงานดังรูปที่ 5.1 โดยตัวเลขในวงกลมจะแสดงลำดับขั้นตอนการทำงาน ซึ่งแบ่งได้เป็น 4 ขั้นตอนย่อย ๆ ลำดับขั้นตอนที่ 1 และ 2 นี้ จะเป็นส่วนที่เกี่ยวข้องกับการรับ input เสียงเข้ามา โดยใช้โปรแกรมที่รันบนระบบปฏิบัติการ Windows เนื่องจาก Windows มีฟังก์ชันทางด้านมัลติมีเดียมาให้ซึ่งเกี่ยวข้องกับการรับเสียงและบันทึกเสียง โดยไม่ต้องไปสั่งงานที่การ์ดเสียงโดยตรง และ อีก 2 ส่วนที่เหลือคือส่วนที่ 3 และ 4 จะเป็นโปรแกรมบนระบบปฏิบัติการ Dos ซึ่งเป็นส่วนที่ใช้ในการประมวลผล input เสียง



รูปที่ 5.1 แสดงขั้นตอนการทำงานของส่วนเรียนรู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในเรื่องต้นนี้จะตัดปัญหาด้านการรับเสียงออกไปก่อนโดยการใช้โปรแกรม Sound Recorder ที่มีแถมมาให้ใน package ของ Windows อยู่แล้ว แต่จะทำการพัฒนาโปรแกรมในส่วนนี้ภายหลัง เพราะฉะนั้นในส่วนของการเรียนรู้จึงแสดงขั้นตอนการสร้าง ส่วนที่สำคัญที่สุดก่อนคือขั้นตอนที่ 3 และ 4 ในรูปที่ 5.1

5.1.1 ส่วนอ่านข้อมูลจากไฟล์เสียง

ในส่วนของการข้อมูลจากไฟล์เสียงนี้ ไม่ใช่ขั้นตอนที่ง่ายทีเดียวนัก เนื่องจากการจะอ่านข้อมูลขึ้นมาได้นั้นจำเป็นจะต้องรู้จักฟอร์แมตของไฟล์ประเภท .WAV นี้ก่อน และลักษณะการเก็บข้อมูลของไฟล์ชนิดนี้เพื่อที่จะได้ดึงเฉพาะส่วนที่เป็นข้อมูลของไฟล์เสียงขึ้นมาใช้งานได้ถูกต้อง

ลักษณะของไฟล์เสียงที่ได้จากโปรแกรม Sound Recorder นี้จะไม่ได้ถูกบีบอัด (Compress) ไว้ดังนั้นจึงสามารถดูได้ว่าไฟล์เสียงนี้เก็บอะไรอยู่บ้างโดยใช้โปรแกรม editor ใด ๆ ดูไฟล์ได้ เช่น q editor หรือ ส่วน view file ของ norton commander (nc) แต่ข้อมูลที่ view ดูนี้จะพบว่าไม่สื่อความหมายเท่าใด เนื่องจากข้อมูลที่เก็บในไฟล์นั้นเก็บเป็นกันณะของตัวอักษร (Character) ที่มีค่ารหัสแอสกี (ASCII Code) ตรงกับตัวเลขที่ได้จากการแซมปลิง (Sampling) กล่าวคือส่วนข้อมูลนี้จริง ๆ แล้วจะเป็นค่าตัวเลขระหว่าง 0 ถึง 255 เนื่องจากเป็นความสามารถของการ์ดเสียงนี้ที่ใช้จำนวนบิตต่อการแซมปลิง (Sampling) ได้เพียง 8 บิต เพราะฉะนั้นค่าตัวเลขที่ได้จากการแซมปลิงจึงมีค่าอยู่ 256 ระดับ คือ 0 ถึง 255 ซึ่งเท่ากับจำนวนรหัสแอสกีพอดี ตัวอย่างเช่น ถ้าตัวเลขที่ได้จากการแซมปลิง เป็น 60 61 62 63 64 65 จะมีการบันทึกค่าลงในไฟล์ดังนี้ คือ

60 แทนด้วย < (เนื่องจากรหัส ASCII ของ < คือ 60)

61 แทนด้วย = (เนื่องจากรหัส ASCII ของ = คือ 61)

62 แทนด้วย > (เนื่องจากรหัส ASCII ของ > คือ 62)

63 แทนด้วย ? (เนื่องจากรหัส ASCII ของ ? คือ 63)

64 แทนด้วย @ (เนื่องจากรหัส ASCII ของ @ คือ 64)

65 แทนด้วย A (เนื่องจากรหัส ASCII ของ A คือ 65)

เพื่อให้ข้อมูลที่อยู่ในไฟล์สื่อความหมายมากขึ้นจึงทำการแปลงให้อยู่ในรูปตัวเลข ก่อนที่จะนำไปประมวลผลต่อไป โดยได้แสดง ส่วนที่สำคัญของโปรแกรมที่ใช้ในการอ่านค่าขึ้นมาและเขียนลงไฟล์ใหม่ในรูปของตัวเลข ดังรูปที่ 5.2 ตัวแปร ch กำหนดให้มีชนิดเป็น unsigned char ทั้งนี้เพื่อให้ในการเขียนลงไฟล์เป็นตัวเลขจะได้มีค่าระหว่าง 0 ถึง 255 มิใช่ค่า -128 ถึง 127 และจากรูปจะเห็นว่าตอนแรกจะมีการเลื่อนไฟล์พอยน์เตอร์ไปยังไบต์ที่ 45 ก่อนเนื่องจากว่า 44 ไบต์แรกของ wave file นี้จะเก็บส่วนที่เป็น header ซึ่งเก็บรายละเอียดของ wave file นั้น ส่วนข้อมูลจริงจะเริ่มที่ประมาณไบต์ 45 นอกจากนี้จะเห็นว่ก่อนจะถึงส่วนตรวจสอบเงื่อนไขของ while loop จะมีการอ่านค่าข้อมูลขึ้นมาอีกครั้งซึ่งการอ่านนี้เป็นการอ่านทิ้ง คือไม่ได้นำข้อมูลนั้นไปใช้งาน เนื่องจากไฟล์เสียงนี้จะเก็บข้อมูลแต่ละค่าสลับกับช่องว่าง เพราะฉะนั้นจึงทำการอ่านข้อมูลส่วนที่เป็นช่องว่างนี้ทิ้งไป

```

handle = open("wav1.wav",O_RDONLY);
stream2 = fopen("wavtxt1.TXT", "w+");
lseek(handle, 45L, SEEK_SET);
i=0;
do
{
    read(handle, &ch, 1);
    i = i+1;
    if ( ((i%200) == 1) && (i != 1) ) {
        fprintf (stream2, "\n");
        i = 1;
    }
    if (ch == '\x1A') {
        ch = '\x19';
        lseek(handle, 1L, SEEK_CUR);
    }
    fprintf (stream2, "%4d",ch);
    read(handle, &ch, 1);
} while (!eof(handle));

```

รูปที่ 5.2 แสดงโมดูลที่ใช้อ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2 ส่วนประมวลผล

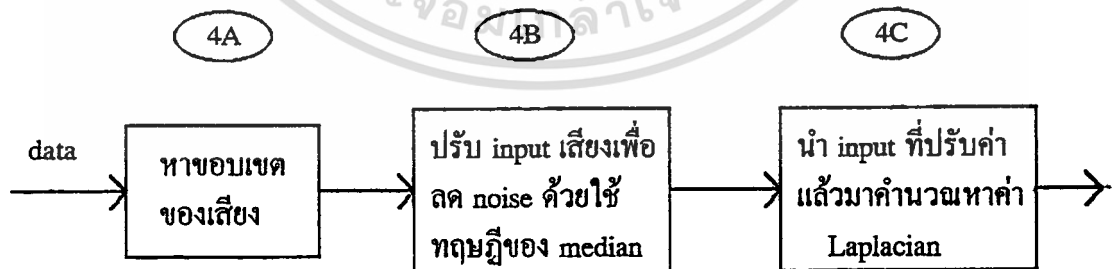
ประกอบด้วยขั้นตอนย่อย ดังแสดงในรูปที่ 5.3 มีรายละเอียดดังนี้

5.1.2.1 การหาขอบเขตของเสียง

เป็นการหาจุดเริ่มต้นและจุดสิ้นสุดที่แท้จริงของเสียงเนื่องจากเสียงที่รับเข้ามาได้ปรากฏว่าช่วงแรกของเสียงจะเป็นสัญญาณเงียบคือไม่มีข้อมูลเสียงใด ๆ เลยแล้วจึงเป็นข้อมูลเสียงและในช่วงท้ายเสียงก็เช่นกันหลังจากหมดส่วนที่เป็นข้อมูลเสียงแล้วก็จะเป็นสัญญาณเงียบที่ไม่มีข้อมูลเสียง ซึ่งเราจะสามารถนำประโยชน์ตรงนี้มาใช้ในการหาช่วงที่เป็นข้อมูลของเสียงที่แท้จริงได้

แต่ถ้าทดลอง view ไฟล์เสียงที่ได้แปลงเป็นตัวเลขด้วยโปรแกรมที่แสดงในหัวข้อที่ผ่านมาจะพบว่าช่วงที่เป็นสัญญาณเงียบนั้นจะไม่ได้มีค่าตัวเลขเป็นศูนย์ทั้ง ๆ ที่จริง ๆ แล้วควรจะเป็นศูนย์ เนื่องจากไม่มีข้อมูลของเสียง ทั้งนี้อาจจะเป็น noise ที่เกิดขึ้นขณะบันทึกเสียงก็ได้

เพราะฉะนั้นปัญหาที่สำคัญของการหาขอบเขตก็คือก็ควรให้ช่วงที่เป็นสัญญาณเงียบดังกล่าวมีค่าตัวเลขอยู่ระหว่างเท่าใด ค่านี้จึงจัดเป็นพารามิเตอร์ที่สำคัญตัวหนึ่งที่จะต้องทำการปรับให้เหมาะสมจึงจะทำให้การหาขอบเขตของเสียงได้ใกล้เคียงความเป็นจริงมากที่สุด



รูปที่ 5.3 แสดงขั้นตอนของส่วนประมวลผล

แต่เนื่องจากว่า input เสียงนี้มีขนาดใหญ่มากถ้าต้องประมวลผลข้อมูลเสียงทุกตัวเลขในไฟล์จะเสียเวลามาก ดังนั้นเพื่อเป็นลดเวลาในการหาขอบเขตลงจึงใช้วิธีการประมวลผลข้อมูลเป็นช่วง โดยในส่วนของโครงงานเลือกแบ่งข้อมูลออกเป็นช่วงละ 200 ตัว และแต่ละช่วง 200 ตัวนั้นจะนำข้อมูลเฉพาะ 20 ตัวแรกของแต่ละช่วงมาประมวลผลเพื่อหาขอบเขต พบว่าผลที่ได้นี้ก็ยังคงใกล้เคียงความถูกต้องกับการประมวลผลกับทุกตัวเลขในไฟล์ ซึ่งการประมวลผลในลักษณะเช่นนี้คล้ายกับการมองข้อมูลเสียงในลักษณะ 2 มิติ แล้วมี window จำกัดขอบให้มองเห็นเฉพาะ 20 ตัวแรก ดังตัวอย่างในรูปแบบที่ 5.4 ข

2	2	2	2	2	2	2	2	1	1	3	3	2	2	1	1	3	3	3	3	1	1	...	(n200)	
1	1	4	4	0	0	2	2	2	2	2	2	2	1	1	3	3	3	3	1	1	2	2	...	(n200)
4	4	2	2	3	3	4	4	2	2	5	5	3	3	3	3	2	2	3	3	4	4	...	(n200)	
2	2	0	0	255	255	2	2	0	0	1	1	0	0	1	1	2	2	0	0	255	255	...	(n200)	
3	3	2	2	2	2	3	3	3	3	2	2	3	3	4	4	3	3	2	2	3	3	...	(n200)	
10	10	11	11	11	11	10	10	10	10	10	10	8	8	5	5	4	4	0	0	252	252	...	(n200)	
19	19	20	20	22	22	23	23	20	20	16	16	12	12	7	7	2	2	0	0	252	252	...	(n200)	

รูปที่ 5.4 ก แสดงลักษณะข้อมูลของเสียงที่แปลงเป็นตัวเลขแล้ว

2	2	2	2	2	2	2	2	1	1	3	3	2	2	1	1	3	3	3	3				
1	1	4	4	0	0	2	2	2	2	2	2	2	1	1	3	3	3	3	1	1			
4	4	2	2	3	3	4	4	2	2	5	5	3	3	3	3	2	2	3	3				
2	2	0	0	255	255	2	2	0	0	1	1	0	0	1	1	2	2	0	0				
3	3	2	2	2	2	3	3	3	3	2	2	3	3	4	4	3	3	2	2				
10	10	11	11	11	11	10	10	10	10	10	10	8	8	5	5	4	4	0	0				
19	19	20	20	22	22	23	23	20	20	16	16	12	12	7	7	2	2	0	0				

รูปที่ 5.4 ข. แสดงลักษณะของข้อมูลที่มองเห็นในการหาขอบเขต

หลักเกณฑ์ในการหาขอบเขต

1. การหาช่วงหรือแถวเริ่มต้นของส่วนที่เป็นข้อมูล มีหลักดังนี้

- ให้มีค่า ≥ 10 อย่างน้อย 10 ตัวจาก 20 ตัวแรกที่ถูกจำกัดให้เห็นของแต่ละช่วง 200 ตัว กล่าวคือ พิจารณา 20 ตัวแรกแล้วข้ามไปอีกทีละ 200 ตัว คือตัวที่ 221, 421, 621, 821, ...

2. การหาช่วงหรือแถวสุดท้ายของข้อมูล มีหลักดังนี้

- ต้องมีค่า ≥ 10 อย่างน้อย 10 ตัวจาก 20 ตัวแรกของแถว และ

- ถ้าแถวสุดท้ายเดิมอยู่ห่างจากแถวสุดท้ายที่หาได้ใหม่เกิน เกิน 3 แถว ให้ถือเอาแถวสุดท้ายเดิมเป็นแถวสุดท้ายจริง แล้วยกเลิกการหา เนื่องจาก

- ถ้าข้อมูลค่ามากกว่า 250 ขึ้นไปอยู่ติดกันทั้ง 20 ตัวที่มองเห็นตั้งแต่ 3 แถวขึ้นไป ให้เอาแถวสุดท้ายเดิมเป็นแถวสุดท้ายจริง แล้วยกเลิกการหาเช่นกัน

สาเหตุที่ต้องมีเกณฑ์การหา 2 ข้อหลังของการหาแถวสุดท้ายของข้อมูล ก็เนื่องจากลักษณะทั้งสองกรณีดังกล่าวจะเป็น noise คือจะมีแถวที่ห่างจากแถวอื่นมากจนผิดสังเกต หรือมีค่าข้อมูลที่มากกว่า 250 ขึ้นไปอยู่ติดกันมาก ๆ ประมาณ 3 แถวขึ้นไป จึงควรตัดแถวเหล่านี้ทิ้งไปไม่ใช่เป็นแถวสุดท้ายของข้อมูล ซึ่งค่าตัวเลขต่าง ๆ ที่แสดงในหลักเกณฑ์การหาแถวแรกและแถวสุดท้ายของข้อมูลนั้นจะเป็นพารามิเตอร์ที่ต้องมีการปรับค่าให้เหมาะสมจึงจะใกล้เคียงความเป็นจริง

รูปที่ 5.5 จะแสดงพารามิเตอร์ต่าง ๆ ที่กล่าวมาแล้ว และตัวอย่างของ module ที่ใช้ในการหาขอบเขตของเสียง

```

void find_border(unsigned int k)
{
    int i,gt_treshole,cnt,gt_ovbound,line_ovb,result;
    int old_lineb;
    unsigned int length;
    unsigned char ch;
    unsigned int loop=0;

    cnt = 1; i = 0; gt_treshole = 0; gt_ovbound = 0;
    fst_line = 0; lst_line = 0; line_ovb = 0; length = 0;
    old_lineb = 0; sec_line = 0;

    do
    {
        /* read a char from the file */
        ch = data[length+i];
        i = i+1;
        if (i == 21) {
            length = SPAN * cnt;
            i=0;
            ch = data[length+i];
            i=1;
            cnt = cnt+1;
            gt_treshole = 0;
            gt_ovbound = 0;
        }
        if ((int)ch >= TRESHOLE)
            gt_treshole = gt_treshole+1;
        if ((int)ch >= OVER_BOUND)
            gt_ovbound = gt_ovbound+1;
    }

```

ต่อนหน้าถัดไป

5.1.2.2 การปรับอินพุท

การปรับอินพุทในขั้นตอนนี้เป็นกรลด noise และปรับค่าบางอย่างของข้อมูลเสียงที่หาช่วงที่เป็นข้อมูลได้เรียบร้อยแล้ว โดยใช้ทฤษฎีของ Median Filtering ซึ่งได้กล่าวไว้ในส่วนของทฤษฎีแล้ว

ขั้นตอนการทำ median มีดังนี้ คือ

1. กำหนดว่าจะ Median กับข้อมูลขนาดกี่ตัว เช่น 3 ตัว 5 ตัว โดยจำนวนตัวที่เลือกนี้ควรเป็นเลขคี่
2. ทำการเรียงลำดับ (sort) ข้อมูล จากมากไปน้อย หรือจากน้อยไปมาก
3. เลือกข้อมูลที่อยู่ตรงกลางระหว่างจำนวนข้อมูลทั้งหมดที่ sort แล้ว จะได้ค่า median เช่นเลือกข้อมูลที่ทำ Median ขนาด 3 ตัว เมื่อ sort แล้ว ค่าที่จะเป็น Median คือค่าข้อมูลตัวที่ 2

โมดูลของส่วนการหา median นี้แสดงไว้ในรูปที่ 5.6 ก. และ 5.6 ข.

```

unsigned char median3(unsigned int i)
{
    int loop;
    unsigned char dat1,dat2,dat3,tmp;
    dat1 = data[i]; dat2 = data[i+1]; dat3 = data[i+2];

    for (loop=0; loop<3; loop++) {
        if (dat1 > dat2) { tmp = dat1; dat1 = dat2; dat2 = tmp; }
        if (dat2 > dat3) { tmp = dat2; dat2 = dat3; dat3 = tmp; }
    }
    return dat2;
}

```

รูปที่ 5.6 ก แสดงโมดูลที่ใช้หา median 3

```

unsigned char median5(unsigned int i)
{
    unsigned char dat1,dat2,dat3,dat4,dat5,tmp;
    dat1 = data[i]; dat2 = data[i+1]; dat3 = data[i+2];
    dat4 = data[i+3]; dat5 = data[i+4];

    do {
        if (dat1 > dat2) { tmp = dat1; dat1 = dat2; dat2 = tmp; }
        if (dat2 > dat3) { tmp = dat2; dat2 = dat3; dat3 = tmp; }
        if (dat3 > dat4) { tmp = dat3; dat3 = dat4; dat4 = tmp; }
        if (dat4 > dat5) { tmp = dat4; dat4 = dat5; dat5 = tmp; }

    } while (dat1>dat2 || dat2>dat3 || dat3>dat4 || dat4>dat5);

    return dat3;
}

```

รูปที่ 5.6 ข แสดงโมดูลที่ใช้หา median 5

5.1.2.3 การหาค่า Laplacian

หลังจากผ่านการหาขอบเขต และ ปรับ input แล้วก็ถึงขั้นตอนสำคัญคือการหาค่า Laplacian ของข้อมูลแต่ละตัวในไฟล์ ซึ่งการหาค่า Laplacian นั้นแท้จริงแล้วก็คือการเน้นจุดที่มีการเปลี่ยนแปลงระหว่างจุดที่อยู่ติดกันของ input นั้นเอง ซึ่งค่า Laplacian นี้จะมีทั้งค่าบวกและลบ โดยถ้าตัวเลขยังมีขนาดมากเท่าใดก็ยิ่งแสดงให้เห็นว่าระหว่างจุดใด ๆ นั้นมีค่าของการเปลี่ยนแปลงแอมพลิจูดมากเท่านั้น ซึ่งจากการทดลองพบว่าค่า Laplacian ทั้งค่าบวกและลบนั้นจะมีจำนวนใกล้เคียงกัน เพราะฉะนั้นในโครงงานนี้จึงเลือกค่า Laplacian เฉพาะค่าบวกไปเก็บเป็นค่าสถิติ ของเสียงแต่ละเสียงว่ามีค่าดังกล่าวเป็นอย่างไร โดยค่าที่จะนำไปเก็บเป็นสถิตินั้นจะเป็นการหาช่วงของค่า laplacian ค่าต่าง ๆ ของแต่ละเสียงซึ่งจะมีค่าแตกต่างกัน จึงนำค่าเหล่านี้ไปใช้ในการจำแนกเสียงแต่ละเสียงได้

โดยสมการที่ใช้ในการหาค่า Laplacian นั้นจะใช้แบบ 1 มิติ เนื่องจากคลื่นเสียงมีลักษณะเป็นมิติเดียว ดังที่ได้เสนอไปแล้วในส่วนของทฤษฎี แต่จะนำมาเสนอที่ตรงนี้อีกครั้ง คือ

$$\Delta_x^2 f(i,j) = f(i+1,j) + f(i-1,j) - 2f(i,j)$$

และโมดูลในการหาค่า aplacian ดังกล่าวแสดงไว้ในรูปที่ 5.7

```
int laplacian(unsigned int i)
{
    return ( ((int)data[i+1] + (int)data[i-1] - 2*(int)data[i] ) );
}
```

รูปที่ 5.7 แสดงโมดูลที่ใช้ในการหาค่า Laplacian



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การทดลองและผลการทดลอง

การทดลองนี้เป็นการทดลองเพื่อหาค่าลักษณะต่าง ๆ ของแต่ละเสียง โดยดึงเอาลักษณะเด่นของแต่ละเสียงออกมาเพื่อใช้เป็นข้อมูลทางสถิติ ในการแบ่งคลาสของ pattern ต่อไป

ในการทดลองขั้นต้นนี้จะทำการบันทึกเสียงของคำแต่ละคำด้วย โปรแกรม Sound recorder ที่มีมาให้ บน Windows แล้ว โดยเสียงที่ทำการบันทึกมีดังนี้คือ คำว่า

- "หยุด" ในตารางบันทึกผลและกราฟ จะแทนด้วยคำว่า 'stop'
- "เดิน" ในตารางบันทึกผลและกราฟ จะแทนด้วยคำว่า 'walk'
- "ซ้าย" ในตารางบันทึกผลและกราฟ จะแทนด้วยคำว่า 'left'
- "ขวา" ในตารางบันทึกผลและกราฟ จะแทนด้วยคำว่า 'right'

โดยแต่ละคำนั้นจะทำการบันทึกเสียง 30 ครั้ง แล้วนำมาหาค่าต่าง ๆ ดังนี้คือ

1. จำนวนช่วงหรือแถว (กล่าวไว้ในบทความสร้าง) ของแต่ละเสียงที่มีข้อมูลอยู่
2. จำนวนจุด laplacian มีค่าเกิน 200
3. จำนวนจุดของlaplacian ที่มีค่าอยู่ระหว่าง -1 และ 101
4. จำนวนจุดของlaplacian ที่มีค่าอยู่ระหว่าง 100 และ 200
5. จำนวนจุดของlaplacian ที่มีค่าอยู่ระหว่าง 100 และ 245
6. จำนวนจุดของlaplacian ที่มีค่าอยู่ระหว่าง 200 และ 245
7. จำนวนจุดของlaplacian ที่มีค่าอยู่ระหว่าง 0 และ 200
8. จำนวนจุดของlaplacian ที่มีค่าอยู่ระหว่าง 0 และ 30
9. จำนวนจุดของlaplacian ที่มีค่าอยู่ระหว่าง 0 และ 20
10. จำนวนจุดของlaplacian ที่มีค่าอยู่ระหว่าง 0 และ 10
11. จำนวนจุดของlaplacian ที่มีค่าอยู่ระหว่าง 5 และ 30
12. จำนวนจุดของlaplacian ที่มีค่าอยู่ระหว่าง 200 และ 253
13. จำนวนจุดของlaplacian ที่มีค่าอยู่ระหว่าง 244 และ 253
14. จำนวนจุดของlaplacian ที่มีค่าอยู่ระหว่าง 100 และ 253

ส่วนต่อไปจะแสดงตารางบันทึกผลการทดลองของข้อมูลที่เก็บได้ และ นำไป

เขียนกราฟให้เห็นลักษณะการแบ่งแยกกลุ่มของแต่ละคำได้ชัดเจนขึ้น

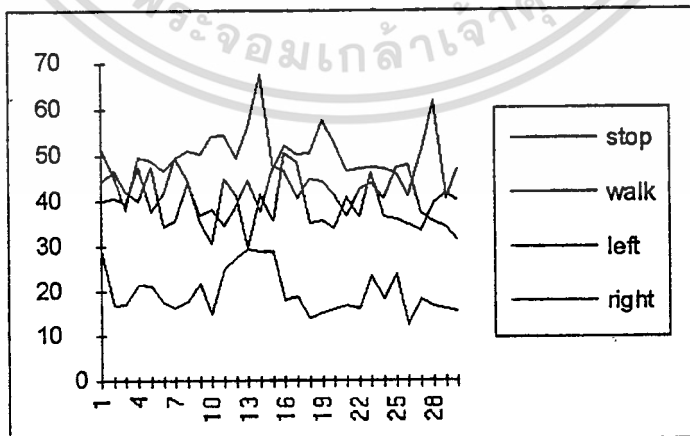
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9	10
stop	23	15	16	19	19	16	15	17	19	14
walk	38	38	37	43	34	38	43	40	33	30
left	44	47	43	42	47	36	38	45	40	40
right	48	44	38	47	47	45	46	46	46	50

	11	12	13	14	15	16	17	18	19	20
stop	21	22	24	22	23	16	17	13	14	15
walk	40	38	29	37	34	43	43	34	34	33
left	36	42	45	41	46	47	43	46	45	44
right	51	47	54	60	45	50	47	48	53	47

	21	22	23	24	25	26	27	28	29	30
stop	16	15	20	17	21	12	17	15	15	15
walk	36	34	41	35	33	34	34	37	38	36
left	39	43	45	43	48	47	38	37	36	34
right	46	45	47	45	45	42	49	55	40	45

ตารางที่ 6.1 แสดงจำนวนแถวที่มีข้อมูลอยู่ของแต่ละเดือน



รูปที่ 6.1 แสดงกราฟที่ได้จากตารางที่ 6.1

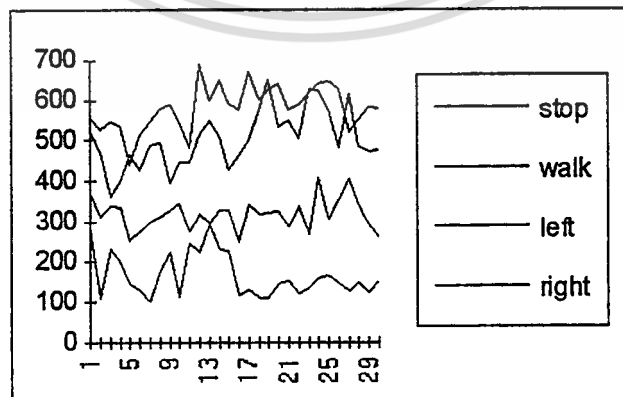
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9	10
stop	286	110	233	196	144	130	101	176	222	111
walk	372	314	338	335	253	274	300	314	329	347
left	559	526	546	532	440	514	546	579	589	548
right	524	472	361	402	464	428	488	495	393	447

	11	12	13	14	15	16	17	18	19	20
stop	244	224	293	229	225	116	128	110	108	144
walk	276	319	296	328	329	249	343	319	322	327
left	481	689	598	649	593	576	668	601	625	639
right	445	514	550	507	426	463	501	577	650	534

	21	22	23	24	25	26	27	28	29	30
stop	152	118	136	158	163	146	126	147	123	153
walk	289	338	269	407	308	356	404	343	293	259
left	577	588	617	639	647	626	522	549	582	579
right	549	503	626	623	575	483	614	484	473	476

ตารางที่ 6.2 แสดงจำนวนจุดที่มีค่า laplacian มากกว่า 200



รูปที่ 6.2 แสดงกราฟที่ได้จากตารางที่ 6.2

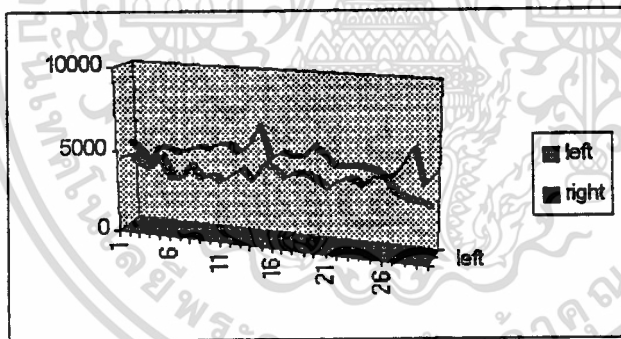
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9	10
left	4391	4657	4199	4001	4746	3436	3563	4405	3690	3839
right	5187	4533	3818	4981	4892	4663	4951	5119	5018	5441

	11	12	13	14	15	16	17	18	19	20
left	3476	3916	4450	3769	4772	4659	4048	4503	4423	4144
right	5456	4954	5623	6781	4692	5246	5039	5050	5765	5329

	21	22	23	24	25	26	27	28	29	30
left	3704	4263	4397	4055	4743	4814	3745	3539	3429	3132
right	4667	4720	4755	4713	4612	4123	5066	6189	4053	4725

ตารางที่ 6.3 แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง -1 และ 101



รูปที่ 6.3 แสดงกราฟที่ได้จากตารางที่ 6.3

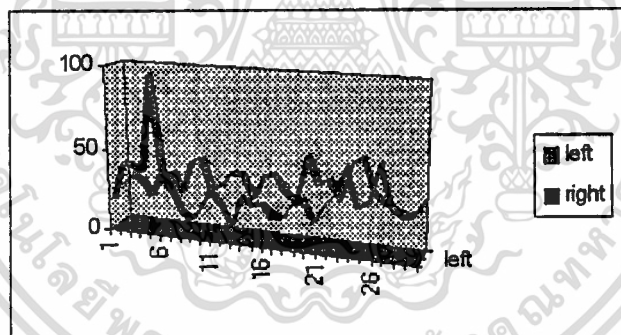
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9	10
left	18	41	40	37	96	35	39	27	46	48
right	24	30	28	18	33	22	8	5	13	25

	11	12	13	14	15	16	17	18	19	20
left	29	33	42	41	27	41	43	33	29	56
right	16	3	23	19	19	13	12	23	24	12

	21	22	23	24	25	26	27	28	29	30
left	39	43	31	53	58	37	32	29	26	29
right	23	30	48	23	27	51	26	20	21	31

ตารางที่ 6.4 แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 100 และ 200



รูปที่ 6.4 แสดงกราฟที่ได้จากตารางที่ 6.4

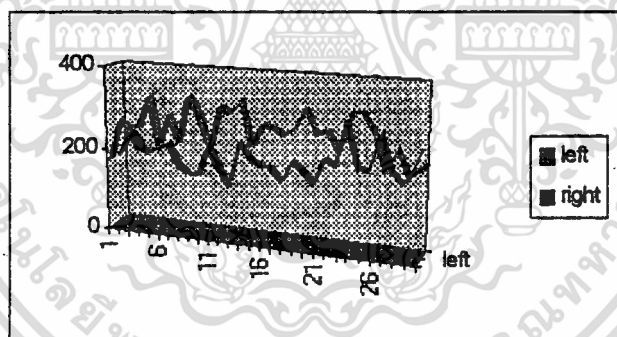
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9	10
left	170	264	253	257	330	217	289	226	338	301
right	175	228	190	178	195	205	152	137	136	204

	11	12	13	14	15	16	17	18	19	20
left	214	320	306	337	235	286	280	267	284	332
right	158	115	222	189	170	172	136	188	170	143

	21	22	23	24	25	26	27	28	29	30
left	277	282	235	332	334	290	204	256	194	256
right	204	191	300	188	183	277	186	161	183	210

ตารางที่ 6.5 แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 100 และ 245



รูปที่ 6.5 แสดงกราฟที่ได้จากตารางที่ 6.5

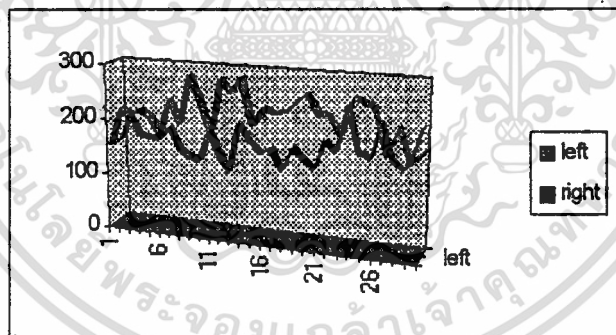
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9	10
left	149	217	208	216	207	179	241	198	288	245
right	149	195	161	159	156	181	140	127	123	178

	11	12	13	14	15	16	17	18	19	20
left	183	285	260	288	207	238	231	234	252	270
right	138	107	197	166	149	158	119	164	146	126

	21	22	23	24	25	26	27	28	29	30
left	235	235	202	267	262	252	171	223	167	227
right	179	160	246	164	156	217	156	140	160	174

ตารางที่ 6.6 แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 200 และ 245



รูปที่ 6.6 แสดงกราฟที่ได้จากตารางที่ 6.6

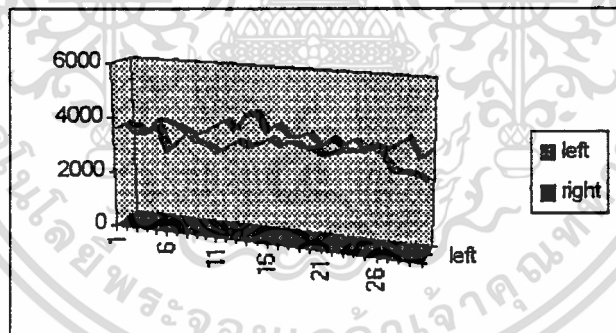
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9	10
left	3601	3827	3431	3494	3903	2931	3250	3618	3346	3227
right	3540	3519	3238	3772	3787	3670	3428	3340	3531	3787

	11	12	13	14	15	16	17	18	19	20
left	2981	3263	3519	3356	3422	3713	3450	3657	3474	3503
right	3987	3657	4284	4324	3709	4021	3586	3604	3787	3306

	21	22	23	24	25	26	27	28	29	30
left	3209	3405	3509	3518	3756	3586	2969	2966	2915	2727
right	3658	3426	3705	3281	3554	3454	3690	4006	3284	3582

ตารางที่ 6.7 แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 0 และ 200



รูปที่ 6.7 แสดงกราฟที่ได้จากตารางที่ 6.7

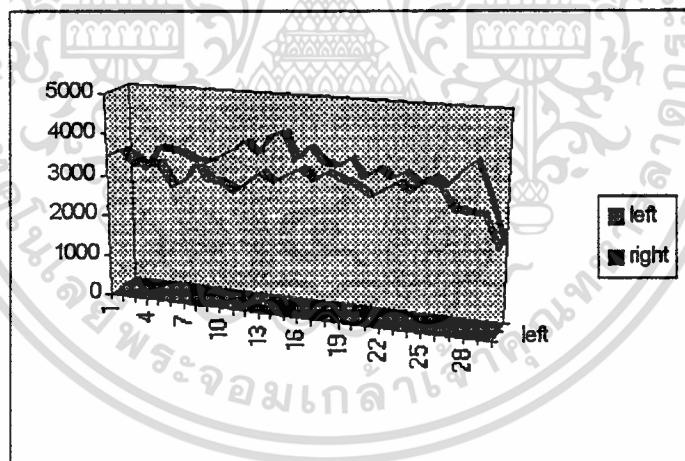
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9	10
left	3466	3600	3252	3275	3363	2788	2980	3454	3037	2994
right	3422	3311	3027	3606	3535	3471	3338	3288	3446	3649

	11	12	13	14	15	16	17	18	19	20
left	2818	3072	3326	3110	3314	3487	3246	3474	3303	3254
right	3883	3603	4081	4190	3535	3887	3516	3477	3660	3230

	21	22	23	24	25	26	27	28	29	30
left	2984	3171	3366	3195	3460	3369	2867	2830	2825	2576
right	3499	3271	3445	3176	3395	3197	3574	3878	3153	3371

ตารางที่ 6.8 แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 0 และ 30



รูปที่ 6.8 แสดงกราฟที่ได้จากตารางที่ 6.8

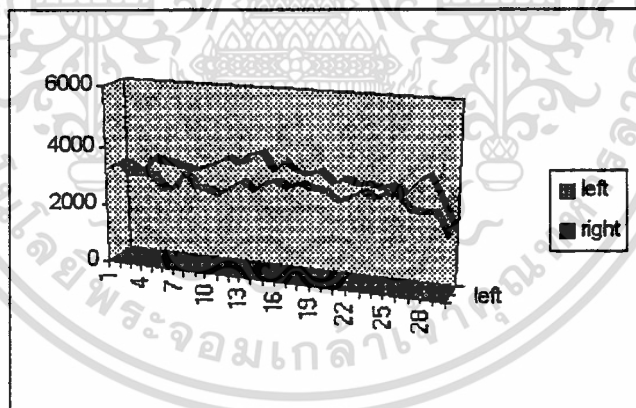
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9	10.
left	3280	3379	3044	3064	3048	2631	2711	3274	2798	2758
right	3244	3140	2869	3455	3369	3298	3208	3169	3318	3440

	11	12	13	14	15	16	17	18	19	20
left	2629	2919	3118	2854	3165	3283	3056	3279	3122	3090
right	3702	3487	3851	4010	3323	3700	3388	3299	3519	3138

	21	22	23	24	25	26	27	28	29	30
left	2769	2946	3179	2975	3221	3162	2711	2663	2698	2408
right	3301	3113	3194	3070	3250	2955	3428	3703	2996	3162

ตารางที่ 6.9 แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 0 และ 20



รูปที่ 6.9 แสดงกราฟที่ได้จากตารางที่ 6.9

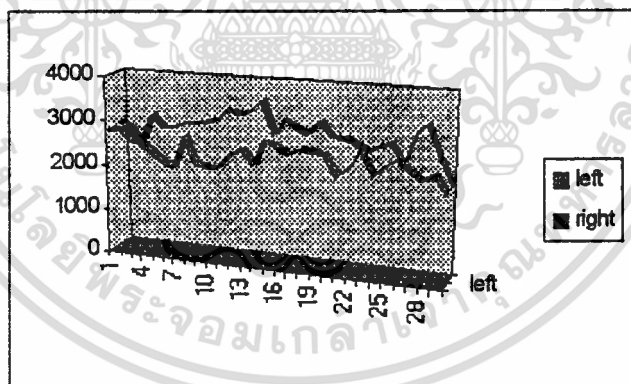
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9	10
left	2790	2779	2521	2496	2322	2094	2018	2736	2115	2112
right	2864	2683	2373	3044	2773	2795	2905	2917	2977	3032

	11	12	13	14	15	16	17	18	19	20
left	2135	2429	2524	2246	2780	2679	2520	2685	2629	2589
right	3293	3201	3264	3554	2810	3171	3010	2882	3187	2857

	21	22	23	24	25	26	27	28	29	30
left	2152	2368	2770	2235	2487	2622	2390	2260	2342	2023
right	2847	2755	2616	2767	2827	2403	3103	3296	2624	2652

ตารางที่ 6.10 แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 0 และ 10



รูปที่ 6.10 แสดงกราฟที่ได้จากตารางที่ 6.10

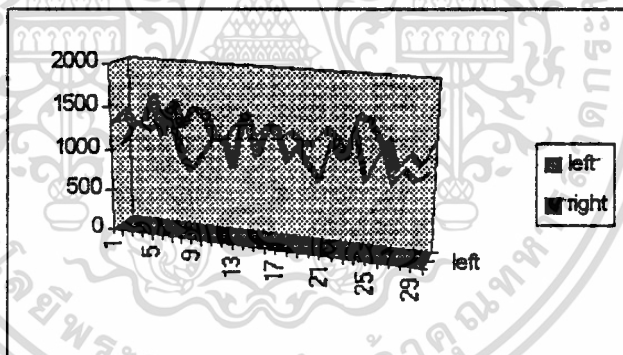
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9	10
left	1290	1438	1280	1341	1647	1186	1604	1317	1546	1488
right	977	1244	1206	1178	1423	1247	849	724	917	1121

	11	12	13	14	15	16	17	18	19	20
left	1217	1235	1407	1506	1042	1443	1289	1371	1245	1252
right	1128	802	1464	1187	1307	1343	928	1129	911	758

	21	22	23	24	25	26	27	28	29	30
left	1426	1354	1119	1610	1568	1330	888	1053	925	1040
right	1222	1023	1397	784	1055	1311	956	1175	1039	1299

ตารางที่ 6.11 แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 5 และ 30



รูปที่ 6.11 แสดงกราฟที่ได้จากตารางที่ 6.11

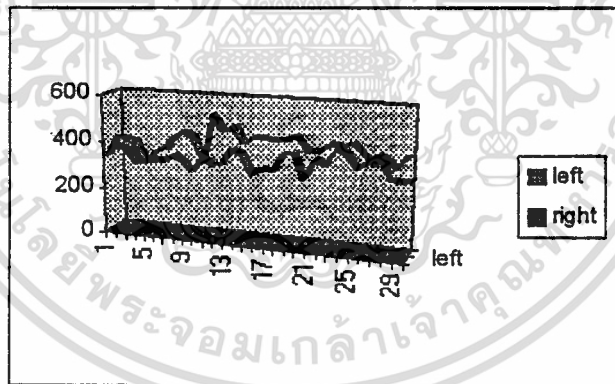
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9	10
left	337	414	412	401	322	362	379	423	464	438
right	354	342	288	296	307	303	336	312	271	332

	11	12	13	14	15	16	17	18	19	20
left	339	540	474	492	430	459	456	451	455	482
right	304	315	382	353	278	305	299	374	382	279

	21	22	23	24	25	26	27	28	29	30
left	418	416	454	443	463	426	380	413	376	423
right	363	347	419	395	339	365	402	306	298	305

ตารางที่ 6.12 แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 200 และ 253



รูปที่ 6.12 แสดงกราฟที่ได้จากตารางที่ 6.12

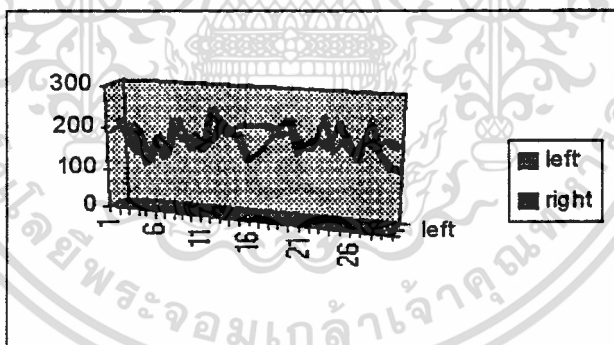
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9	10
left	188	197	204	185	115	183	138	225	176	193
right	205	147	127	137	151	122	196	185	148	154

	11	12	13	14	15	16	17	18	19	20
left	156	255	214	204	223	221	225	217	203	212
right	166	208	185	187	129	147	180	210	236	153

	21	22	23	24	25	26	27	28	29	30
left	183	181	252	176	201	174	209	190	209	196
right	184	187	173	231	183	148	246	166	138	131

ตารางที่ 6.13 แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 244 และ 253



รูปที่ 6.13 แสดงกราฟที่ได้จากตารางที่ 6.13

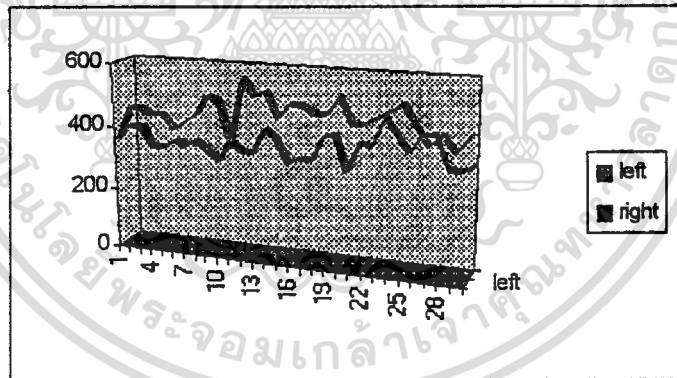
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9	10
left	358	461	457	442	445	400	427	451	514	494
right	380	375	317	315	346	327	348	322	284	358

	11	12	13	14	15	16	17	18	19	20
left	370	575	520	541	458	507	505	484	487	544
right	324	323	407	376	299	319	316	398	406	296

	21	22	23	24	25	26	27	28	29	30
left	460	463	487	508	535	464	413	446	403	452
right	388	378	473	419	366	425	432	327	321	341

ตารางที่ 6.14 แสดงจำนวนจุดที่มีค่า laplacian ระหว่าง 100 และ 253



รูปที่ 6.14 แสดงกราฟที่ได้จากตารางที่ 6.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

1. ค่าทางสถิติที่สามารถใช้แยกเสียงคำว่า "หยุด" คือจำนวนแถวของข้อมูล เนื่องจากเมื่อดูจากกราฟ ที่ 6.1 จะเห็นว่าเส้นของกราฟ แยก ออกจากกลุ่มอย่างเห็นได้ชัด
2. ค่าทางสถิติที่สามารถใช้แยกเสียงคำว่า "เดิน" คือ จำนวนจุดที่มีค่า laplacian มากกว่า 200 ขึ้นไป
3. ค่าทางสถิติที่สามารถใช้แยกแยะเสียงคำว่า "ซ้าย" และ "ขวา" ได้ คือ
 - จำนวนจุดที่มีค่า laplacian ระหว่าง -1 และ 100 โดยถ้ามีจำนวนมากกว่า 4660 จะเป็นคำว่า "ขวา" นอกนั้นเป็นคำว่า ซ้าย
 - จำนวนจุดที่มีค่า laplacian ระหว่าง 100 และ 245 โดยถ้าจำนวนมากกว่าหรือเท่ากับ 214 จะเป็นคำว่า "ซ้าย"
 - จำนวนจุดที่มีค่า laplacian ระหว่าง 200 และ 245 โดยถ้าจำนวนมากกว่าหรือเท่ากับ 182 จะเป็นคำว่า "ซ้าย"
 - จำนวนจุดที่มีค่า laplacian ระหว่าง 0 และ 10 โดยถ้าจำนวนมากกว่าช่วง 2600-2700 จะเป็นคำว่า "ขวา"
 - จำนวนจุดที่มีค่า laplacian ระหว่าง 100 และ 253 โดยถ้าจำนวนมากกว่าหรือเท่ากับ 420 จะเป็นคำว่า "ซ้าย"

บทที่ 7

บทวิจารณ์และสรุป

จากผลการทดลองที่ผ่านมาทั้งหมดอาจสรุปได้ดังนี้ คือ

1. ไม่สามารถใช้ค่าทางสถิติเพียงแบบใดแบบหนึ่งในการจำแนก input ทั้งหมดได้ เพราะเหตุว่าบางครั้งลักษณะที่ดึงออกมาได้นั้นมีลักษณะกำกวมยากที่จะตัดสินใจได้ว่า จะจัดให้อินพุทนั้นอยู่ในคลาสใด

ข้อสนับสนุนนี้ดูได้จาก ลักษณะของเส้นกราฟที่ได้จากข้อมูลของ pattern ต่าง ๆ เป็นจำนวนมาก ของแต่ละคลาส (คือของแต่ละคำ นั้นเอง) ตัวอย่างเช่นเมื่อใช้ข้อมูลของ จำนวนจุดที่มีค่า laplacian ระหว่าง 100 และ 245 โดยให้จำนวนจุดค่าดังกล่าวมากกว่าหรือเท่ากับ 214 จุดเป็นคำว่า "ชาย" ที่เหลือให้เป็นคำว่า "ขวา"

2. ผลจากข้อ 1 ต้องมีการใช้ข้อมูลทางสถิติหลาย ๆ ค่าจึงจะแยกแยะข้อมูลได้ ใกล้เคียงความเป็นจริงมากยิ่งขึ้น ดังจะเห็นได้จากการทดลองที่มีการเก็บค่าข้อมูลระหว่าง 2 คำคู่กัน คือ "ชาย" และ "ขวา" เป็นจำนวนมากทั้งนี้เนื่องจากข้อมูลที่เก็บได้จากคำทั้งสอง บางครั้ง กำกวมกัน จึงต้องทำการทดลองเก็บค่า หลายค่าเพื่อแยกความแตกต่างของคำทั้งสอง ให้ได้

3. การเลือกค่าข้อมูลที่จะใช้เป็นสถิตินั้นค่อนข้างยาก และไม่มีกฎเกณฑ์แน่นอนในการเลือก โครงการนี้จึงต้องอาศัยการสังเกตจากลักษณะของข้อมูลที่ได้ประมวลผลแล้วว่า มีลักษณะข้อมูลใดที่โดดเด่นเป็นพิเศษสำหรับแต่ละ pattern นั้นก่อนที่จะทำการทดลองจริงเพื่อดึงค่าข้อมูลที่มีลักษณะเด่นพิเศษนั้นออกมา

4. การจำแนกคำที่ใช้ในโครงการงานทำในลักษณะลำดับดังนี้คือ (โดยดูข้อมูลสถิติที่ใช้ตรวจในส่วนถัดไป)

4.1 ทำการตรวจคำว่า "หยุด" ก่อน

4.2 ตรวจคำว่า "เดิน" เป็นลำดับถัดมา

4.3 ในกรณีที่ไม่ใช่ 2 คำแรก คือเป็นคำว่า "ซ้าย" และ "ขวา" ต้องตรวจ 2 คำนี้คู่กัน โดยใช้ข้อมูลสถิติสำหรับ 2 คำนี้ในการแยก

5. ลักษณะของข้อมูลเสียงบางคำในกลุ่มคำเดียว มีลักษณะแตกต่างจากในกลุ่มมากจนกระทั่งไม่สามารถจัดเข้ากับกลุ่มได้เลยไม่ว่าจะเก็บค่าสถิติเช่นใด เช่น คำว่า "ขวา"

คำที่ 3 ซึ่งจะมีค่าตัวเลขน้อยกว่ากลุ่มมาก แม้จะผ่านการ normalize แล้วก็ตาม ทั้งนี้ อาจเนื่องมาจากความผิดพลาดในขั้นตอนของการเก็บข้อมูล ทำให้ข้อมูลผิดเพี้ยนไป

สรุปลักษณะข้อมูลของสถิติที่ใช้แยกความแตกต่างของเสียงแต่ละคำ

- ได้
1. คำว่า "หยุด" สามารถใช้ค่าสถิติของจำนวนแถวที่มีข้อมูลอยู่ในการจำแนก
 2. คำว่า "เดิน" ใช้จำนวนจุดที่มีค่า laplacian มากกว่า 200 ขึ้นไป
 3. คำว่า "ซ้าย" และ "ขวา" ใช้ข้อมูลในการจำแนกคือ
 - จำนวนจุดที่มีค่า laplacian ระหว่าง -1 และ 101
 - จำนวนจุดที่มีค่า laplacian ระหว่าง 100 และ 245
 - จำนวนจุดที่มีค่า laplacian ระหว่าง 0 และ 10
 - จำนวนจุดที่มีค่า laplacian ระหว่าง 100 และ 253

ข้อสังเกตเกี่ยวกับลักษณะของเสียงบางคำ

- คำว่า "ซ้าย" จะมีจำนวนจุดที่มีค่า laplacian สูง ๆ จำนวนมาก และค่า laplacian ต่ำ ๆ มีจำนวนน้อย

- คำว่า "ขวา" จะมีจำนวนจุดที่มีค่า laplacian สูง ๆ จำนวนน้อย และค่า

laplacian ต่ำ ๆ มีจำนวนมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลโครงการ

โครงการนี้เป็น การนำเสนอแนวทางหนึ่งที่ใช้ในการสอนให้คอมพิวเตอร์รู้จักเสียงพูด โดยใช้หลักการทั่วไปที่ค่อนข้างจะฉีกแนวไปจากวิธีที่มักมีการใช้กันมาก ๆ เช่น การใช้ neural network ซึ่งแต่ละวิธีนี้ต่างก็มีข้อดีและข้อเสียในตัวเอง และวิธีการที่ใช้ในโครงการนี้ก็เช่นกัน ข้อดี คือมีการประมวลผลที่ค่อนข้างเร็ว โดยเฉพาะเมื่อตัดขั้นตอนการดึงข้อมูลจากเสียงที่บันทึกไว้ในไฟล์ออก แล้วใช้วิธีการรับข้อมูลจากเสียงภายนอกมาเก็บในหน่วยความจำแล้วประมวลผลทันที ซึ่งจะทำให้การทำงานรวดเร็วขึ้น สำหรับประสิทธิภาพของวิธีการนี้เท่าที่ทดลองในโครงการแม้จะไม่ดีเลิศแต่ ก็ให้ผลลัพธ์การทำงานที่อยู่ในขั้นน่าพอใจ เนื่องจากเปอร์เซ็นต์ความถูกต้องยังคงค่อนข้างสูง แต่ข้อเสียของวิธีนี้ที่เห็นได้อย่างชัดเจนก็คือ การเลือกค่าข้อมูลที่จะนำมาใช้เป็นสถิติ ค่อนข้างยากและไม่มีหลักเกณฑ์แน่นอน และยังขึ้นอยู่กับประสบการณ์ของผู้ออกแบบด้วย ข้อมูลส่วนใหญ่ที่ได้ในโครงการนี้จึงมาจากการทดลองเป็นส่วนใหญ่ ซึ่งอาจมีการเลือกค่าที่ไม่ใช่ลักษณะของ pattern นั้นอย่างแท้จริงก็ได้

อย่างไรก็ตามดังที่ได้กล่าวไปแล้วว่าโครงการนี้เป็นเพียงการนำเสนอทางเลือกหนึ่งที่ใช้ในเทคโนโลยีด้านเสียง ซึ่งมีแนวโน้มว่าในอนาคตจะมีที่ใช้กันแพร่หลายสำหรับการสั่งงานคอมพิวเตอร์ด้วยเสียง การจะเลือกวิธีการใดในการนำไปใช้งานจริงจึงขึ้นอยู่กับวิสัยทัศน์ของผู้ออกแบบเป็นสำคัญ

ภาคผนวก

ฟอร์แมตของไฟล์ .WAV

ลักษณะของไฟล์ชนิดนี้จะมีโครงสร้างเป็นแบบ RIFF (Resource Interchange File Format)

ซึ่งโครงสร้างแบบนี้ใช้สำหรับ ริชอาร์ทไฟล์ของมัลติมีเดีย

RIFF File Format

บล็อกพื้นฐานของ RIFF File เรียกว่า 'chunk' ซึ่งมีฟอร์แมตดังนี้ คือ

<rID><rLen><rData(rLen)>

ซึ่ง

- <rID> "RIFF" ซึ่งใช้เป็นตัวแทนของข้อมูล ใน chunk (4 ไบต์)
- <rLen> เป็นความยาวของข้อมูลใน chunk ที่ตามมาหลังจากนี้ (4 ไบต์)
- <rData> คือส่วนที่เป็นข้อมูลของ chunk (RIFF Data Chunk) (ยาว rlen ไบต์)

WAV Format Definition

ฟอร์แมตของ WAV ในส่วนข้อมูลของ chunk ยังสามารถแบ่งออกได้อีกเป็น chunk ย่อย ๆ โดยจะต้องประกอบไปด้วย Format chunk ตามด้วย Data chunk

ซึ่ง

- <wID> "WAVE" เป็นตัวกำหนดว่าข้อมูลมีฟอร์แมตเป็นแบบ WAV format (4 ไบต์)

WAV Format chunk

ส่วนฟอร์แมตของ chunk ประกอบด้วยข้อมูลซึ่งกำหนดฟอร์แมตของของดาต้าใน chunk

โดยมีลักษณะส่วนประกอบดังนี้ คือ

<Format Chunk> = <ChunkID> <flen> <wFormatTag> <nChannels>

<nSamplesPerSec> <nAvgBytesPerSec>

<nBlockAlign> <FormatSpecific>

ซึ่ง

- <fID> "fmt" เป็นตัวกำหนดบล็อกของ Format Chunk (4 ไบต์)
- <fLen> เป็นความยาวของข้อมูลใน Format Chunk ซึ่งตามมา (4 ไบต์)
- <wFormatTag> เป็นตัวบอกประเภทของ Wave format (2 ไบต์)

ตัวอย่างเช่น

Pulse Code Modulation (PCM) format = 01

- <nChannels> เป็นตัวกำหนดจำนวนช่อง (channels) ของเอาต์พุต (2 ไบต์)

ตัวอย่างเช่น

1 = mono, 2 = stereo

- <nSamplesPerSec> เป็นตัวบอกอัตราการแซมปลิง (sampling rate) (จำนวนการแซมเปิลต่อวินาที) ซึ่งแต่ละช่องจะใช้ในการเล่น (play back) (2 ไบต์)

- <nAvgBytesPerSec> เป็นตัวบอกจำนวนไบต์โดยเฉลี่ยต่อวินาทีที่ใช้ในการเคลื่อนย้าย (transfer) ข้อมูล (2 ไบต์) ซึ่ง

$$\text{<nAvgBytesPerSec>} = \text{nChannels} * \text{nSamplesPerSec} * (\text{nBitsPerSample} / 8)$$

- <nBlockAlign> เป็นตัวกำหนด block alignment (เป็นไบต์) ของข้อมูลใน Data chunk

ซอฟต์แวร์ที่ใช้ในการ playback จะต้องมีการประมวลผล (process) ข้อมูลหลาย

<nBlockAlign> ไบต์

ณ เวลานั้น ๆ ดังนั้นค่าของ <nBlockAlign> สามารถใช้กำหนด buffer alignment (2 ไบต์)

$$\text{<nBlockAlign>} = \text{nChannels} * (\text{nBitsPerSample} / 8)$$

- <FormatSpecific> 필드นี้เป็นที่เก็บพารามิเตอร์ (2 ไบต์)

WAV Data Chunk

Data chunk นี้เป็นส่วนที่เก็บข้อมูลจริงของ WAV ไฟล์นี้ โดยฟอร์แมตของข้อมูลนี้ขึ้นอยู่กับค่าของ <wFormatTag> ที่เก็บใน Format chunk

$$\text{<Data Chunk>} = \text{<dId><dLen><dData(dLen)>}$$

ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- <dlid> "data" เป็นตัวกำหนดว่าบล็อกนั้นเป็น Data Chunk (4 ไบต์)
- <dLen> เป็นตัวกำหนดความยาวของข้อมูลใน Data Chunk ที่ตามมา (4 ไบต์)
- <dData> คือส่วนที่เป็นข้อมูลจริง (ยาว dLen ไบต์)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

หนังสือ

- [1] Don Pearson, " Image Processing ", Mc Graw-Hill Book Company, 1991
- [2] Rafael C.Gonzalez,Paul Wintz, " Digital Image Processing ", Second Edition,Addison-Wesley Publishing Company,1987.
- [3] Peter M. Ridge, David M. Golden, Ivan Luke, and Scott sindorf, "Sound Blaster: The Official Book, Second Edition, Mc Graw-Hill Book Company, 1994
- [4] Jim Conger, "Windows API New Testament", The Waite Group Press, 1993
- [5] Paul Perry ,"Multimedia Developer 's Guide", Sams Publishing,1994



กิติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จลงได้ ด้วยความช่วยเหลืออย่างดีของบุคคลหลายท่าน
ดังนั้นคณะผู้จัดทำจึงใคร่ขอขอบคุณทุกท่าน ที่มีส่วนช่วยเหลือ สนับสนุน
และแนะนำในหลาย ๆ ด้าน โดยเฉพาะอย่างยิ่งบุคคลท่านต่อไปนี้

- รศ.ดร. ครรชิต ไมตรี อาจารย์ที่ปรึกษาที่ให้โอกาสในการทำงานนี้
- พี่ทิพ ที่ช่วยจัดเรียงหน้าและให้ยืมปริ้นเตอร์ใช้พิมพ์งาน

สุดท้ายนี้ ขอกราบขอบพระคุณ บิดา มารดา ที่เข้าใจและสนับสนุนในการศึกษา
ตลอดจนกระทั่งการทำโครงการชิ้นนี้สำเร็จลงด้วยดี

คณะผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้