

การพัฒนาโปรแกรมแปลภาษามือเป็นตัวอักษรภาษาไทย
โดยใช้ออโตมาตาแบบจำกัด

DEVELOPMENT OF TRANSLATION PROGRAM FROM THAI SIGN
LANGUAGE TO THAI ALPHABETS USING FINITE AUTOMATA

จิวรรธ ตุ่มศรี
JIRAWAT TUMSRI

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2561

KMITL-2018-SC-M-002-037

การพัฒนาโปรแกรมแปลภาษามือเป็นตัวอักษรภาษาไทย
โดยใช้ออโตมาตาแบบจำกัด

DEVELOPMENT OF TRANSLATION PROGRAM FROM THAI SIGN
LANGUAGE TO THAI ALPHABETS USING FINITE AUTOMATA

จิรวรรธ ตุ่มศรี

JIRAWAT TUMSRI

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2561

KMITL-2018-SC-M-002-037

DEVELOPMENT OF TRANSLATION PROGRAM FROM THAI SIGN
LANGUAGE TO THAI ALPHABETS USING FINITE AUTOMATA

JIRAWAT TUMSRI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE
DEGREE OF MASTER OF SCIENCE PROGRAM IN COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2018

KMITL-2018-SC-M-002-037

COPYRIGHT 2018

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อวิทยานิพนธ์	การพัฒนาโปรแกรมแปลภาษามือเป็นตัวอักษรภาษาไทย โดยใช้ซอฟต์แวร์อโตมาตาแบบจำกัด
ชื่อนักศึกษา	จิรวรรธ ตุ่มศรี
รหัสประจำตัว	576705074
ปริญญา	วิทยาศาสตรมหาบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
พ.ศ.	2561
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ.ดร. วรางคณา กิมปาน

บทคัดย่อ

กลุ่มผู้ที่มีลักษณะความบกพร่องทางการได้ยินหรือสื่อความหมายใช้การแสดงท่าทางภาษามือสำหรับการสื่อสารในชีวิตประจำวัน ซึ่งบุคคลทั่วไปไม่มีความรู้ความเข้าใจภาษามือ จึงทำให้เป็นอุปสรรคในการสื่อสารระหว่างผู้ที่มีลักษณะความบกพร่องทางการได้ยินหรือสื่อความหมายกับบุคคลทั่วไป ผู้วิจัยจึงเกิดแนวคิดในการพัฒนาโปรแกรมแปลภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทย โดยใช้อุปกรณ์ลิฟโมชันในการตรวจจับตำแหน่งปลายนิ้วมือทั้ง 5 และกลางฝ่ามือในรูปแบบแกน x และ y จากนั้นนำข้อมูลที่ได้มาทำการปรับให้อยู่ในรูปแบบที่เหมาะสม และสร้างเป็นต้นไม้ตัดสินใจ โดยใช้เทคนิคควอดทรี ผลลัพธ์ที่ได้เป็นรหัสตัวเลขแต่ละนิ้วมือ และนำทฤษฎีอโตมาตาแบบจำกัดมาสร้างต้นแบบภาษามือตัวอักษรภาษาไทยสำหรับเป็นต้นแบบในการค้นหาตัวอักษรภาษาไทย ผลการทดสอบสามารถจำแนกภาษามือตัวอักษรภาษาไทยจำนวน 42 ตัวอักษรได้ ซึ่งมีค่าเฉลี่ยความถูกต้อง 72.27% สามารถนำไปพัฒนาเป็นโปรแกรมที่สามารถนำไปใช้ในการสื่อสารระหว่างผู้ที่มีลักษณะความบกพร่องทางการได้ยินหรือสื่อความหมายกับบุคคลทั่วไปต่อไปได้ พร้อมทั้งยังเปิดโอกาสให้บุคคลทั่วไปสามารถเรียนรู้และเข้าถึงภาษามือได้ง่ายยิ่งขึ้น นอกจากนี้ยังเป็นการพัฒนาการฝึกการเคลื่อนไหวของภาษามือที่ถูกต้องอีกด้วย

คำสำคัญ : การสื่อสาร ออโตมาตาแบบจำกัด ผู้บกพร่องทางการได้ยินหรือสื่อความหมาย ลิฟโมชัน
ควอดทรี ภาษามือตัวอักษรภาษาไทย

Thesis Title	Development of Translation Program from Thai Sign Language to Thai Alphabets Using Finite Automata
Student Name	Jirawat Tumsri
Student ID	57605074
Degree	Master of Science Program in Computer Science
Department	Computer Science
Year	2018
Thesis Advisor	Asst. Prof. Dr. Warangkhana Kimpan

Abstract

People with hearing or speech impairments use hand gestures for everyday communication. The other people do not understand the language. This is a barrier to communication between people with hearing impairment and other people. Researchers have developed the concept of hand-translation in Thai sign language. Leap Motion Controller was applied to detect 5-finger-tip position and palm center in the form of X and Y axis, then the data will be normalized. Next, decision tree was created by using the Quadtree technique. The result is a numeric code for each finger. The finite automata theory is used to create the prototype of the Thai sign language. The results of the tests can be classified into 42 characters of the Thai alphabet, with an accuracy of 72.27%. It can be developed as a prototype program for communicating between people with hearing impairment or to communicate with the public. It also gives people the opportunity to learn and access the hand language more easily. Moreover, it helps people practice the hand language movements in correct formats.

Keywords : Communication, Finite automata, Hearing impairment people, Leap Motion Controller, Quadtree, Thai sign language

กิตติกรรมประกาศ

การทำวิทยานิพนธ์ฉบับนี้ สามารถสำเร็จลงได้ด้วยดีจากการช่วยเหลือ และสนับสนุนของ บุคคลหลายท่าน ผู้จัดทำขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.วรางคณา กัมปาน อาจารย์ที่ปรึกษา วิทยานิพนธ์ที่กรุณาให้คำปรึกษา คำแนะนำด้านการออกแบบ วิเคราะห์ ทดสอบขั้นตอนวิธีและแนวทางการแก้ปัญหา รวมถึงการตรวจสอบ และแก้ไขการเขียนวิทยานิพนธ์ฉบับนี้อย่างละเอียด

ขอขอบพระคุณรองศาสตราจารย์ ดร.จิรพร วีระพันธุ์ ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์ และผู้ช่วย ศาสตราจารย์ ดร.กฤษณะ ชินสาร คณะกรรมการสอบหัวข้อวิทยานิพนธ์ ที่ให้คำแนะนำตลอดข้อ ชี้แนะจนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้

ขอขอบพระคุณอาจารย์วิสันต์ ตั้งวงษ์เจริญที่คอยให้คำปรึกษาทั้งกระบวนการคิดขั้นตอนวิธี ตลอดจนการประสานงานร่วมกับอาจารย์ที่ปรึกษาจนกระทั่งวิทยานิพนธ์นี้สัมฤทธิ์ผลได้ด้วยดีทุก ประการ

ขอขอบคุณพี่ๆ เพื่อนๆ และน้องๆ ทุกคนที่เข้าร่วมทดสอบเก็บข้อมูลในสถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ให้การสนับสนุนช่วยเหลือในการทดสอบขั้นตอนวิธีจนสำเร็จ สมบูรณ์

สุดท้ายนี้ขอกราบขอบพระคุณคุณครอบครัวผู้เป็นแรงกายแรงใจสำคัญตั้งแต่ต้นจนจบ นอกจากนี้ยังมีบุคคลท่านอื่นที่ไม่ได้กล่าวไว้ ณ ที่นี้ จึงใคร่ขอขอบพระคุณทุกท่านที่ให้ความกรุณา มีส่วนร่วมในการให้ความช่วยเหลือ ให้คำปรึกษา ให้คำแนะนำ ตลอดจนให้กำลังใจในการทำ วิทยานิพนธ์ฉบับนี้

จิรวรรธ ตุ่มศรี

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของปัญหาที่ทำการวิจัย.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 สมมติฐานของการวิจัย.....	2
1.4 ขอบเขตของการวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.6 ขั้นตอนของการวิจัย.....	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 ภาษามือ (Sign Language).....	4
2.2 ลีพโมชัน (Leap Motion).....	7
2.2.1 ลักษณะภายนอกและส่วนประกอบภายในอุปกรณ์.....	7
2.2.2 คุณสมบัติ.....	8
2.2.3 ระบบที่รองรับอุปกรณ์.....	9
2.2.4 หน่วยทางกายภาพข้อมูลต่างๆ.....	9
2.2.5 ไลบรารีภาษาจาวา.....	9
2.2.6 ข้อจำกัดการใช้งาน.....	9
2.3 ควอดทรี (Quad Tree).....	11
2.3.1 การสร้างควอดทรี.....	11
2.3.2 การค้นหาตำแหน่งของวัตถุ.....	13
2.3.3 ตัวอย่างงานวิจัยที่นำควอดทรีไปประยุกต์ใช้.....	14
2.4 ออโตมาตาแบบจำกัด (Finite Automata).....	14
2.4.1 องค์ประกอบพื้นฐานของออโตมาตา.....	14
2.4.2 ส่วนประกอบของออโตมาตา.....	17
2.4.3 นิยามออโตมาตาแบบจำกัดเชิงกำหนด.....	18

สารบัญ (ต่อ)

	หน้า
2.5 งานวิจัยที่เกี่ยวข้อง.....	19
บทที่ 3 วิธีการดำเนินงานวิจัย	21
3.1 ขั้นตอนการดำเนินงานวิจัย.....	22
3.2 การจัดแบ่งกลุ่มภาษามือตัวอักษรภาษาไทย	24
3.3 การเก็บข้อมูลต้นแบบภาษามือตัวอักษรภาษาไทย	25
3.4 การปรับข้อมูลให้อยู่ในรูปแบบที่เหมาะสม	26
3.5 อัลกอริทึมในการถอดรหัสภาษามือตัวอักษรภาษาไทย	28
3.6 อัลกอริทึมการหาลำดับและจำนวนของนิ้วมือ.....	30
3.7 อัลกอริทึมการสร้างต้นแบบอโตมาตาแบบจำกัด	31
3.8 อัลกอริทึมการค้นหาวอโตมาตาแบบจำกัดตัวอักษรภาษาไทย	32
บทที่ 4 ผลการวิจัยและการอภิปรายผล	34
4.1 ผลการวิเคราะห์ท่าทางภาษามือตัวอักษรภาษาไทย.....	34
4.2 ผลการถอดรหัสภาษามือตัวอักษรภาษาไทย	35
4.3 ผลการหาลำดับและจำนวนของนิ้วมือ	51
4.4 ผลการสร้างต้นแบบอโตมาตาแบบจำกัดภาษามือตัวอักษรภาษาไทย	51
4.5 ผลการทดสอบการแปลงภาษามือเป็นตัวอักษร	53
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	56
5.1 สรุปผล.....	56
5.2 ข้อเสนอแนะ	57
บรรณานุกรม.....	58
ภาคผนวก.....	59
ภาคผนวก ก.....	60
ประวัติผู้เขียน.....	69

สารบัญตาราง

ตารางที่	หน้า
2.1 ค่าสถานะการรับข้อมูลนำเข้าของ 0 และ 1.....	19
4.1 ความแตกต่างของท่าทางภาษามือตัวอักษรภาษาไทยจำนวน 25 ท่าทาง.....	34
4.2 รหัสการถอดรหัสรูปแบบ THSL การสะกดนิ้วมือ 1 จังหวะ จำนวน 15 ตัวอักษร	48
4.3 รหัสการถอดรหัสรูปแบบ THSL การสะกดนิ้วมือ 2 จังหวะ จำนวน 24 ตัวอักษร	48
4.4 รหัสการถอดรหัสรูปแบบ THSL การสะกดนิ้วมือ 3 จังหวะ จำนวน 3 ตัวอักษร	50
4.5 เพอร์เซ็นต์ความถูกต้องในการแปลงภาษามือเป็นตัวอักษรภาษาไทยการสะกดนิ้วมือ 1 จังหวะ จำนวน 15 ตัวอักษร.....	53
4.6 เพอร์เซ็นต์ความถูกต้องในการแปลงภาษามือเป็นตัวอักษรภาษาไทยการสะกดนิ้วมือ 2 จังหวะ จำนวน 24 ตัวอักษร.....	53
4.7 เพอร์เซ็นต์ความถูกต้องในการแปลงภาษามือเป็นตัวอักษรภาษาไทยการสะกดนิ้วมือ 3 จังหวะ จำนวน 3 ตัวอักษร.....	54

สารบัญญรูป

รูปที่	หน้า
2.1 แผนภาพแสดงท่าทางภาษามือตัวอักษร สระ และวรรณยุกต์ภาษาไทย.....	5
2.2 แผนภาพแสดงท่าทางภาษามือตัวอักษรภาอเมริกัน และภาษามือตัวเลข.....	5
2.3 ด้านซ้ายเป็นภาษามือตัวอักษร “ก” ภาษาไทย ด้านขวาภาษามือตัวอักษร “K” ภาษาอเมริกัน	6
2.4 ด้านซ้ายเป็นภาษามือตัวอักษร “ข” ภาษาไทย ด้านขวาภาษามือตัวอักษร “K” และตัวเลข “1” ภาษาอเมริกัน.....	6
2.5 ด้านซ้ายเป็นภาษามือตัวอักษร “ค” ภาษาไทย ด้านขวาเป็นภาษามือตัวอักษร “K” และตัวเลข “2” ภาษาอเมริกัน.....	7
2.6 ทิศทางการตรวจจับในรูปแบบสามมิติของอุปกรณ์ลิฟโมชั่น.....	7
2.7 ลักษณะทางกายภาพของอุปกรณ์ลิฟโมชั่น	8
2.8 ส่วนประกอบภายในอุปกรณ์ลิฟโมชั่น.....	8
2.9 ตัวอย่างการใช้งานลิฟโมชั่น.....	10
2.10 การแบ่งควอดทรีและการจำแนกเป็นรูปแบบต้นไม้ตัดสินใจ	11
2.11 การแบ่งพื้นที่เพื่อวางวัตถุลงในแต่ละพื้นที่ย่อย	12
2.12 การจัดเก็บวัตถุลงในแต่ละโหนดของต้นไม้ตัดสินใจ	12
2.13 ควอดทรีขนาด 300 x 300 พิกเซล บรรจุเส้นตรง 1,000 เส้น	13
2.14 ประเภทของภาษาและอโตมาตา	15
2.15 องค์ประกอบพื้นฐานของอโตมาตา.....	15
2.16 การจำลองการคำนวณทางคอมพิวเตอร์ $f(x) = (x + 5) / 5$	16
2.17 การจำลองการคำนวณทางคอมพิวเตอร์ $f(x) = (x + 5) / 5$ (ต่อ).....	17
2.18 ส่วนประกอบของอโตมาตา	17
2.19 แผนภาพส่วนประกอบของอโตมาตา	18
3.1 แผนภาพการทำงานของโปรแกรม	21
3.2 ภาพรวมขั้นตอนการดำเนินงานวิจัย.....	22
3.3 ขั้นตอนการสร้างต้นแบบอโตมาตาแบบจำกัด.....	23
3.4 ขั้นตอนการแปลงภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทย.....	23

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.5 แผนภาพตัวอักษรภาษาไทย จำแนกตามจังหวัดการเคลื่อนไหว	24
3.6 ท่าทางภาษามือตัวอักษรภาษาไทยจำแนกตามลักษณะท่าทางภาษามือ	25
3.7 ลักษณะมือก่อนเริ่มต้นการจัดเก็บข้อมูล	26
3.8 ตำแหน่งกลางมือบนตารางควอดทรี	27
3.9 อัลกอริทึมการเข้ารหัสรูปแบบภาษามือ	29
3.10 ตัวอย่างต้นแบบต้นไม้ตัดสินใจตัวอักษร “ก” ของนิ้วมือทั้ง 5 นิ้ว	29
3.11 อัลกอริทึมการหาลำดับและจำนวนของนิ้วมือ	30
3.12 อัลกอริทึมการสร้างออโตมาตาจำกัด	31
3.13 อัลกอริทึมการค้นหออโตมาตาแบบจำกัดตัวอักษรภาษาไทย	32
4.1 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “ก”	35
4.2 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “ต”	36
4.3 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “ส”	36
4.4 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “พ”	37
4.5 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “ห”	37
4.6 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “บ”	38
4.7 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “ร”	38
4.8 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “ว”	39
4.9 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “ด”	39
4.10 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “ฟ”	40
4.11 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “ล”	40
4.12 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “ย”	41
4.13 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “ม”	41
4.14 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “น”	42
4.15 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “อ”	42
4.16 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “ซ” จังหวัดที่ 2	43

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.17 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “จ” จังหวะที่ 1	43
4.18 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “จ” จังหวะที่ 2	44
4.19 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “ง” จังหวะที่ 2	44
4.20 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “ท” จังหวะที่ 2	45
4.21 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “ฉ” จังหวะที่ 1	45
4.22 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวเลข “1”	46
4.23 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวเลข “2”	46
4.24 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวเลข “3”	47
4.25 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวเลข “5”	47
4.26 ตัวอย่างออดิโอมาตาแบบจำกัดตัวอักษรภาษาไทย 1 จังหวะ	51
4.27 ต้นแบบออดิโอมาตาแบบจำกัดตัวอักษรภาษาไทยจำนวน 42 ตัวอักษร	52

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากสำนักสถิติแห่งชาติได้จัดสถิติผู้บกพร่องทางการได้ยินหรือสื่อความหมายอยู่ใน 5 อันดับแรกที่ประชากรมีลักษณะความบกพร่องมากที่สุดของประเทศไทยในปี 2550 แม้ว่าประชากรที่มีลักษณะความบกพร่องทางการได้ยินหรือสื่อความหมายมีแนวโน้มลดลงเล็กน้อยจากร้อยละ 10.3 ในปี 2545 เป็นร้อยละ 9.7 ในปี 2550 แต่การเข้าถึงความช่วยเหลือหรือสวัสดิการของรัฐประชากรที่พิการยังได้รับการศึกษาและส่งเสริมอาชีพไม่ทั่วถึง ซึ่งมีประชากรอายุตั้งแต่ 5 ปีขึ้นไปถึงร้อยละ 24.3 เท่านั้นที่ได้รับการสนับสนุน [1]

จากปัญหาดังกล่าวทางผู้วิจัยจึงมีความสนใจในเรื่องการดำรงชีพในชีวิตประจำวันของกลุ่มผู้ที่มีลักษณะความบกพร่องทางการได้ยินหรือสื่อความหมาย โดยเฉพาะการสื่อสารที่ใช้การแสดงท่าทางภาษามือ ซึ่งบุคคลทั่วไปไม่มีความรู้ความเข้าใจภาษามือ และความแตกต่างของการแสดงท่าทางภาษามือเพียงเล็กน้อยสามารถเปลี่ยนอักขระที่ต้องการได้ ทำให้ผู้ที่มีลักษณะความบกพร่องทางการได้ยินหรือสื่อความหมายเกิดความยากลำบากในการสื่อสารกับบุคคลทั่วไป และถึงแม้ว่าปัจจุบันทางภาครัฐได้เข้ามาช่วยเหลือผู้ที่มีความบกพร่องทางการได้ยินหรือการสื่อความหมายในการสื่อสารกับบุคคลทั่วไปมากขึ้น ตัวอย่างเช่น การพัฒนาตู้ TTRS [2] ตู้สื่อสารสำหรับคนพิการทางการได้ยิน เพื่อลดช่องว่างทางการสื่อสาร แต่ยังคงพบปัญหาในเรื่องการแพร่หลายของการบริการตู้ TTRS ที่มีเพียง 120 จุดตามสถานที่ต่างๆ และปัญหาการเข้าถึงของผู้ที่มีความบกพร่องทางการได้ยินหรือการสื่อความหมาย นอกจากนี้การบริการตู้ TTRS ยังคงต้องมีเจ้าหน้าที่หรือล่ามภาษามือในการให้บริการอีกด้วย

ภาษามือตัวอักษรภาษาไทยนั้น เป็นการแสดงท่าทางโดยใช้การสะกดนิ้วมือแบบมือเดียวเป็นส่วนประกอบในการสะกดตัวอักษรเท่านั้น และจากการศึกษาค้นคว้าเทคโนโลยีเซนเซอร์ที่เหมาะสมสำหรับการตรวจจับท่าทางภาษามือคืออุปกรณ์ลีฟโมชัน (Leap Motion) ที่สามารถตรวจจับตำแหน่งของมือและนิ้วมือในรูปแบบสามมิติได้ และการตรวจจับนั้นมีความละเอียดถึง 0.01 มิลลิเมตร [3] ซึ่งอุปกรณ์มีขนาดเล็ก น้ำหนักเบา และพกพาง่าย

ผู้วิจัยจึงเกิดแนวคิดในการพัฒนาโปรแกรมแปลภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทย โดยใช้อุปกรณ์ลีฟโมชัน ซึ่งข้อมูลนำเข้าเป็นตำแหน่งแนวแกน x และ y ของนิ้วมือต่างๆ และตำแหน่งกลางฝ่ามือไปยังนิ้วชี้ เพื่อหาอัตราส่วนพร้อมทั้งปรับตำแหน่งนิ้วมือให้เหมาะสม ในส่วนการประมวลผลผู้วิจัยได้นำเทคนิคควอทรี (Quad Tree) มาประยุกต์ใช้ เพื่อให้ได้มาซึ่งรหัสตัวเลขแทนตำแหน่งแกน x และ y ของนิ้วมือ ขั้นตอนต่อไปเป็นการนำทฤษฎีออโตมาตาแบบจำกัด (Finite

Automata) มาประยุกต์ใช้ในการสร้างต้นแบบภาษามือตัวอักษรภาษาไทย และวิเคราะห์รหัสตัวเลขในแต่ละนิ้วมือ ซึ่งผลลัพธ์ที่ได้นั้นเป็นตัวอักษรภาษาไทย

ดังนั้นผู้วิจัยจึงคาดหวังว่าโปรแกรมที่พัฒนาขึ้นสามารถนำไปใช้ในการสื่อสารระหว่างผู้ที่มีลักษณะความบกพร่องทางการได้ยินหรือสื่อความหมายกับบุคคลทั่วไปได้ และยังช่วยพัฒนาคุณภาพชีวิตของผู้ที่มีลักษณะความบกพร่องทางการได้ยินหรือสื่อความหมายให้อยู่ในสังคมได้อย่างบุคคลทั่วไป พร้อมทั้งยังเปิดโอกาสให้บุคคลทั่วไปสามารถเรียนรู้และเข้าถึงภาษามือได้ง่ายยิ่งขึ้น นอกจากนี้ยังเป็นการพัฒนาการฝึกการเคลื่อนไหวของภาษามือที่ถูกต้องอีกด้วย

1.2 วัตถุประสงค์ของการวิจัย

- 1) พัฒนารหัสตัวเลขในแต่ละนิ้วมือของภาษามือตัวอักษรภาษาไทย
- 2) พัฒนาต้นแบบอโตมาตาแบบจำกัดภาษามือตัวอักษรภาษาไทย
- 3) พัฒนาอัลกอริทึมในการแปลงรหัสแต่ละนิ้วมือเป็นตัวอักษรภาษาไทย

1.3 สมมติฐานของการวิจัย

- 1) สามารถรับค่าตำแหน่งแกน X และ Y แต่ละนิ้วมือโดยใช้อุปกรณ์สปีโม่ชันได้
- 2) สามารถแปลงตำแหน่งแกน X และ Y แต่ละนิ้วมือเป็นรหัสตัวเลขได้ด้วยเทคนิคควอเทอร์
- 3) สามารถแปลงรหัสแต่ละนิ้วมือเป็นอโตมาตาแบบจำกัดภาษามือตัวอักษรภาษาไทยได้
- 4) รูปแบบอโตมาตาแบบจำกัดสามารถจำแนกรหัสตัวเลขเป็นตัวอักษรภาษาไทยได้

1.4 ขอบเขตของการวิจัย

- 1) กลุ่มตัวอย่างในการทดสอบความถูกต้องของภาษามือ คือ บุคคลทั่วไปที่มีอายุ 18 - 40 ปี และผู้ที่มีลักษณะความบกพร่องทางการได้ยินหรือสื่อความหมาย
- 2) ใช้อุปกรณ์สปีโม่ชันในการตรวจจับการเคลื่อนไหวตำแหน่งแกน x และ y ในแต่ละนิ้วมือ
- 3) ภาษามือตัวอักษรภาษาไทยที่ใช้ในการวิจัย คือ ภาษามือที่มีพยัญชนะภาษาไทยจำนวน 42 ตัวอักษร ซึ่งออกแบบโดยคุณหญิงกมลมา ไกรฤกษ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) โปรแกรมสามารถแปลภาษามือเป็นตัวอักษรภาษาไทยได้ และมีความถูกต้อง
- 2) ผู้ที่มีลักษณะความบกพร่องทางการได้ยินหรือสื่อความหมายสามารถนำโปรแกรมไปใช้ในการสื่อสารกับบุคคลทั่วไปได้
- 3) สามารถนำโปรแกรมไปเป็นแบบฝึกและแบบทดสอบการแสดงท่าทางภาษามือเป็นตัวอักษรภาษาไทยได้

1.6 ขั้นตอนของการวิจัย

- 1) ศึกษาค้นคว้ารวบรวมข้อมูล ทฤษฎี และงานวิจัยที่เกี่ยวข้องในการพัฒนาโปรแกรม
- 2) วิเคราะห์อัลกอริทึมที่มีความเป็นไปได้ในการแปลภาษามือเป็นตัวอักษรภาษาไทยได้ถูกต้อง
- 3) ออกแบบขั้นตอนการแปลภาษามือเป็นตัวอักษรภาษาไทย
- 4) สร้างต้นแบบภาษามือตัวอักษรภาษาไทยที่มีความถูกต้อง และพัฒนาโปรแกรม
- 5) ทดสอบความถูกต้องของโปรแกรมในการแปลภาษามือเป็นตัวอักษรภาษาไทย
- 6) วิเคราะห์ผลการวิจัย และสรุปผลการวิจัย
- 7) จัดทำเอกสารและเสนอรายงานวิทยานิพนธ์ฉบับสมบูรณ์

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ภาษามือ (Sign Language)

ภาษามือเป็นเครื่องมือสำหรับคนหูหนวกหรือบุคคลที่ไม่สามารถยินเสียง จึงเป็นภาษาหนึ่งสำหรับติดต่อหรือสื่อความหมายระหว่างคนหูหนวกกับคนหูหนวกด้วยกัน และคนหูหนวกกับคนหูดี ซึ่งคนหูหนวกใช้สายตาในมองลักษณะท่าทางต่างๆ ที่เคลื่อนไหว ดังนั้นแล้วการสื่อสารใช้การมองเห็นภาพต่างๆ นั้นเป็นสื่อทำให้คนหูหนวกเรียนรู้ความหมาย แม้ว่าสามารถทำความเข้าใจได้ไม่มากหรืออาจเข้าใจไม่ลึกซึ้ง แต่เป็นส่วนหนึ่งที่มีอิทธิพลผลักดันให้คนหูหนวกใช้ส่วนประกอบต่างๆ ของร่างกาย เช่น มือ สีหน้า และกิริยาท่าทางการเคลื่อนไหว เพื่อสื่อความหมายและถ่ายทอดอารมณ์แทนการพูดให้บุคคลอื่นรับรู้และเข้าใจ ท่าทางที่แสดงนั้นสังเกตได้ว่าเป็นท่าทางที่เลียนแบบธรรมชาติมากที่สุด และได้วิวัฒนาการต่อไปเป็นท่าที่ใช้มือเป็นส่วนใหญ่ จึงเกิดเป็นท่าทางที่ใช้แทนความหมายในคำพูด เรียกภาษาท่าทางแบบนี้ว่า “ภาษามือ” [4] และในแต่ละชาติหรือประเทศต่างๆ มีขนบธรรมเนียม ประเพณี วัฒนธรรม ลักษณะภูมิศาสตร์ และภาษาพูดที่แตกต่างกัน ส่งผลให้ภาษามือแต่ละชาติมีความแตกต่างกันไปตามด้วย

สำหรับภาษามือภาษาไทยคุณหญิงกมลลา ไกรฤกษ์ ได้ออกแบบในปี พ.ศ. 2496 แบ่งออกเป็น 3 ส่วน ดังนี้ [5]

ส่วนที่ 1 แบบสะกดตัวอักษร พยัญชนะในภาษาไทยมีจำนวน 44 ตัวอักษร สำหรับแบบสะกดนิ้วมือไทยไม่มีตัวอักษร “ข” และ “ค” เนื่องจากคนหูหนวกพบเห็นตัวอักษร 2 ตัวนี้ในการอ่าน การเขียนน้อยมากจึงไม่มีท่ามือนิ้วดังกล่าว ดังนั้นแบบสะกดตัวอักษรจึงมีจำนวน 42 ท่าทาง แบบสะกดตัวอักษรแสดงดังรูปที่ 2.1

ส่วนที่ 2 แบบสะกดนิ้วมือ สระ วรรณยุกต์ และสัญลักษณ์อื่นๆ เป็นการใช้นิ้วด้านหนึ่งหันฝ่ามือออกไปด้านหน้า กางแต่ละนิ้วออก และใช้นิ้วชี้ของมืออีกด้านหนึ่งชี้ไปยังจุดต่างๆ ของมือ ด้านที่แสดง มีจำนวน 17 ท่าทาง และการใช้การสะกดแบบมือเดียวอีกจำนวน 7 ท่าทาง แบบสะกดนิ้วมือ สระ วรรณยุกต์ และสัญลักษณ์อื่นๆ แสดงดังรูปที่ 2.1 จากรูปไม่มีการแสดงท่าทางของ “ๆ”

ส่วนที่ 3 แบบสะกดตัวเลข ซึ่งมีตั้งแต่ 1 ถึง 10 และตัวเลขหลักสิบถึงล้านตามลำดับ ตัวอย่างแบบสะกดตัวเลขแสดงดังรูปที่ 2.2

≡ แบบสะกดนิ้วมือไทย (Thai Fingerspelling) แผนภาพแสดงการเคลื่อนไหวนิ้วมือเพื่อสื่อสารภาษาในคำศัพท์และสระของไทย

ก (0)	ข (0+1)	ค (0+2)	ฅ (0+3)	ด (1)	ถ (1+1)	ฐ (1+2)	ฌ (1+3)	ท (1+4)	ฏ (1+5)
ส (2)	ศ (2+1)	ษ (2+2)	ซ (2+3)	ฬ (3)	ป (3+1)	ผ (3+2)	ภ (3+3)	พ (3+4)	ธ (3+5)
บ (4)	ร (4+1)	ว (4)	ด (4)	ฎ (4+1)	ฟ (5)	ฝ (5+1)	ล (6)	ฬ (6+1)	จ (6)
ย (7)	ญ (7+1)	ม (8)	น (8)	ณ (8+1)	ง (9+0)				
ช (1+1)	ฌ (1+1+1)	ฉ (1+1+0)	ซ (1+1+1)	ฬ (1+1+1)	ผ (1+1+2)				
แ (1+1)	อ (1+1)	แ (1+0)	แ (1+1)	อ (1)	อ (1)				

ประดิษฐ์โดย คุณหญิงกมล ภิโรฤกษ์
ออกแบบภาพโดย อาจารย์ณัฐนันท์ นวมมาลี ภาควิชาานเทศศิลป์ คณะนิเทศศาสตร์ มหาวิทยาลัยอีสต์สมิธ

รูปที่ 2.1 แผนภาพแสดงท่าทางภาษามือตัวอักษร สระ และวรรณยุกต์ภาษาไทย [6]

≡ American Sign Language (ASL) Fingerspelling
(แบบสะกดอักษรมือแบบอเมริกัน)

A	B	C	D	E	F	G	H	I	J
K	L	M	N	O	P	Q	R	S	T
U	V	W	X	Y	Z				

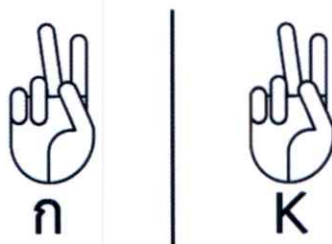
ASL Fingerspelling number 0 - 9 (ตัวสะกดนิ้วมือแบบอเมริกันเลข 0 - 9)

1	2	3	4	5	6	7	8	9	0
---	---	---	---	---	---	---	---	---	---

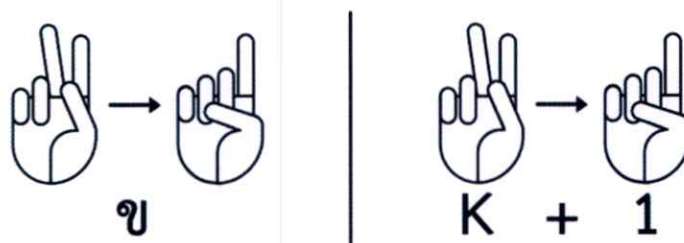
Designed by Nattanan Naewmalee Department of Visual Communication Arts, Albert Laurence School of Communication Arts, Assumption University
ออกแบบภาพโดย อาจารย์ณัฐนันท์ นวมมาลี ภาควิชาานเทศศิลป์ คณะนิเทศศาสตร์ มหาวิทยาลัยอีสต์สมิธ

รูปที่ 2.2 แผนภาพแสดงท่าทางภาษามือตัวอักษรภาษาอเมริกัน และภาษามือตัวเลข [6]

การออกแบบภาษามือตัวอักษรภาษาไทยเกิดจากการผสมผสานหลักการสะกดนิ้วมืออเมริกันกับการสะกดนิ้วมือแบบมือเดียวเข้าด้วยกัน ซึ่งการสะกดนิ้วมือ (Fingerspelling) [6] เป็นรูปแบบการสร้างนิ้วมือให้เป็นตัวอักษร หลักการการออกแบบคือ เสียงพยัญชนะหรือเสียงสระตัวใดในภาษาไทยที่ออกเสียงเหมือนหรือคล้ายคลึงกับเสียงพยัญชนะหรือเสียงสระในภาษาอังกฤษ ให้กำหนดท่าทางการสะกดนิ้วมือเหมือนกัน ตัวอย่างเช่น ภาษามือตัวอักษร “ก” ในภาษาไทยเป็นภาษามือตัวอักษร “K” ในภาษาอังกฤษ เนื่องจากการออกเสียงที่คล้ายคลึงกัน เป็นต้น [7] ลักษณะการสะกดนิ้วมือแสดงดังรูปที่ 2.3



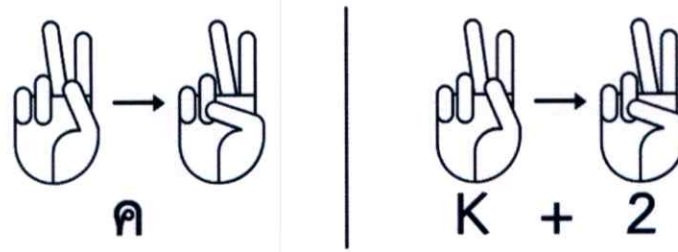
รูปที่ 2.3 ด้านซ้ายเป็นภาษามือตัวอักษร “ก” ภาษาไทย
ด้านขวาภาษามือตัวอักษร “K” ภาษาอังกฤษ



รูปที่ 2.4 ด้านซ้ายเป็นภาษามือตัวอักษร “ข” ภาษาไทย
ด้านขวาภาษามือตัวอักษร “K” และตัวเลข “1” ภาษาอังกฤษ

จากรูปที่ 2.4 จะเห็นว่าภาษามือตัวอักษร “ข” ภาษาไทยมีการเคลื่อนไหว 2 จังหวะ ซึ่งในจังหวะแรกเกิดจากภาษามือตัวอักษร “K” ภาษาอังกฤษ ส่วนจังหวะที่ 2 เกิดจากภาษามือตัวเลข “1” ภาษาอังกฤษ เนื่องจากการออกเสียงตัวอักษร “ข” ในภาษาไทยคล้ายคลึงกับตัวอักษร “K” ในภาษาอังกฤษ แต่เนื่องจากภาษามือตัวอักษร “ก” ในภาษาไทยได้นำภาษามือตัวอักษร “K” ไปใช้แล้วจึงนำภาษามือตัวเลขภาษาอเมริกันมาผสมเข้าด้วยกัน จึงได้เป็นภาษามือตัวอักษร “ข” ที่มีภาษามือตัวอักษร “K” และตัวเลข “1” ภาษาอังกฤษผสมเข้าด้วยกันเป็น 2 จังหวะ

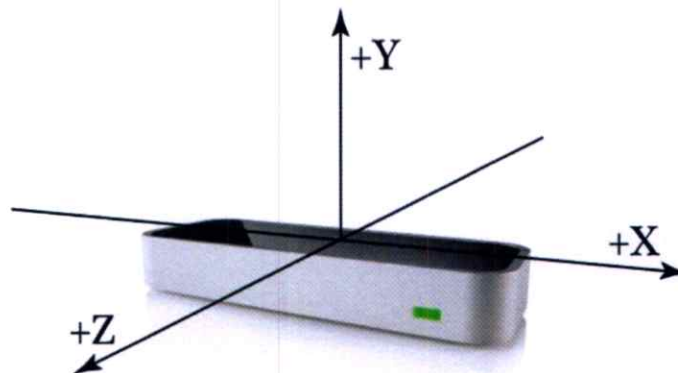
ดังนั้นเมื่อมีการออกเสียงตัวอักษรซ้ำกับตัวอักษรในภาษาอเมริกันที่นำไปใช้แล้ว ในจังหวะที่ 2 ของภาษามือตัวอักษรภาษาไทยจึงเป็นภาษามือตัวเลขภาษาอเมริกันที่เพิ่มตามลำดับตัวเลข ดังเช่นภาษามือตัวอักษร “ค” ภาษาไทยในจังหวะที่ 2 เป็นภาษามือตัวเลข “2” ในภาษาอเมริกัน แสดงดังรูปที่ 2.5



รูปที่ 2.5 ด้านซ้ายเป็นภาษามือตัวอักษร “ค” ภาษาไทย
ด้านขวาเป็นภาษามือตัวอักษร “K” และตัวเลข “2” ภาษาอังกฤษ

2.2 ลีพโมชั่น (Leap Motion)

ลีพโมชั่นผลิตโดยบริษัท Leap motion inc. เป็นอุปกรณ์เซ็นเซอร์สำหรับตรวจจับตำแหน่ง การเคลื่อนไหวของมือ นิ้วมือ และอุปกรณ์ทรงกระบอก เช่น ปากกา ดินสอ พู่กัน ตะเกียบ เป็นต้น ซึ่งเป็นการจับการเคลื่อนไหวในรูปแบบสามมิติ คือ แกน X แทนแกนขนานกับแนวยาว แกน Y แทนแกน ความสูง และแกน Z เป็นแกนแนวขวางที่พุ่งออกจากหน้าจอ [8] แสดงดังรูปที่ 2.6

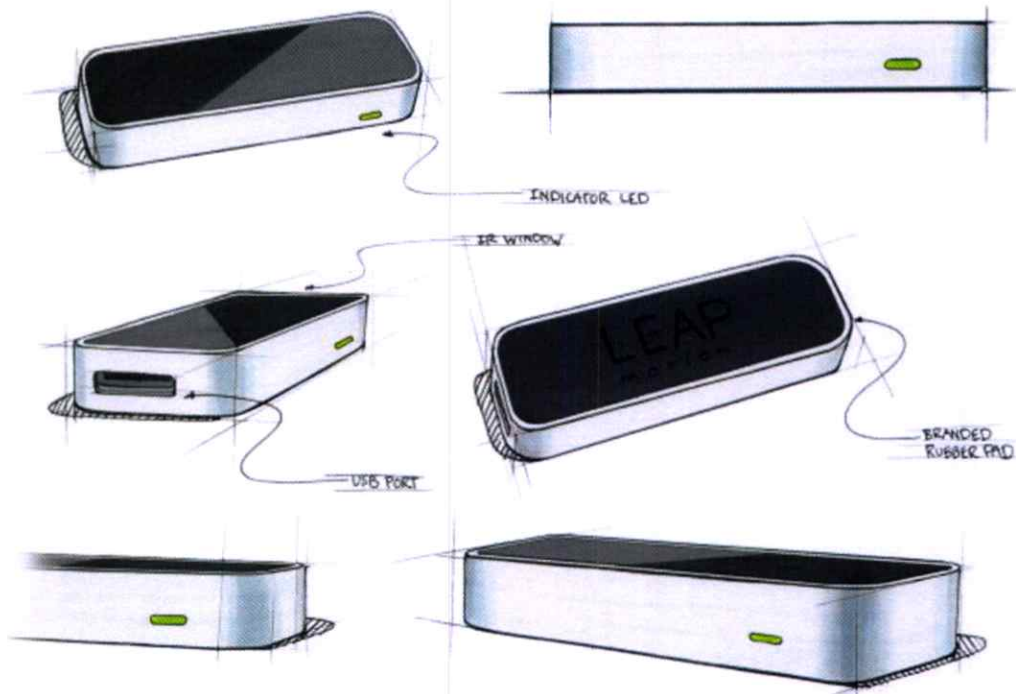


รูปที่ 2.6 ทิศทางการตรวจจับในรูปแบบสามมิติของอุปกรณ์ลีพโมชั่น [9]

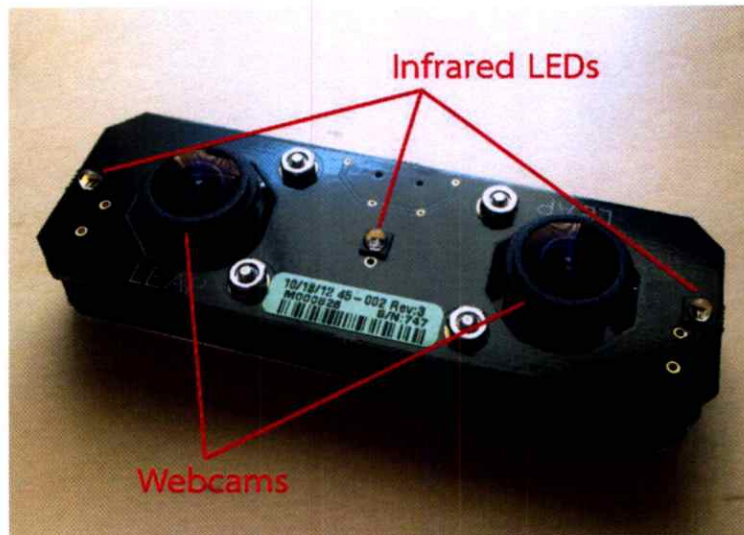
2.2.1 ลักษณะภายนอกและส่วนประกอบภายในอุปกรณ์ [8]

อุปกรณ์ลีพโมชั่นมีลักษณะเป็นกล่องสี่เหลี่ยมผืนผ้า ขนาด $1.2 \times 3 \times 0.5$ นิ้ว น้ำหนัก 45 กรัม ด้านบนเป็นพลาสติกสีดำ ด้านล่างเป็นพื้นผิวแบบยางที่กั้นการเคลื่อนที่ของอุปกรณ์ ด้านข้างทำจากอลูมิเนียม มีช่องเสียบสาย USB 3.0 และช่องไฟ LED สีเขียวเพื่อแสดงสถานะ แสดงดังรูปที่ 2.7

ส่วนประกอบภายในอุปกรณ์ลีพโมชั่นที่สำคัญประกอบด้วย หลอด LED แสงอินฟราเรด จำนวน 3 หลอด และกล้องวิดีโอจำนวน 2 ตัว แสดงดังรูปที่ 2.8



รูปที่ 2.7 ลักษณะทางกายภาพของอุปกรณ์ลีพโมชัน [10]



รูปที่ 2.8 ส่วนประกอบภายในอุปกรณ์ลีพโมชัน [11]

2.2.2 คุณสมบัติ

- 1) สามารถตรวจจับได้ทุกข้อมือและกลางฝ่ามือ หรืออุปกรณ์ทรงกระบอก
- 2) มีความแม่นยำในระดับ 0.01 มิลลิเมตร
- 3) อัตราการส่งข้อมูล 120 เฟรมต่อวินาที
- 4) รองรับการเขียนโปรแกรม 5 ภาษา ได้แก่ C++ C# (และ Unity) Java JavaScript

และ Python

2.2.3 ระบบที่รองรับอุปกรณ์

- 1) ระบบปฏิบัติการ Windows 7 ขึ้นไป หรือ Mac OS X 10.7 Lion ขึ้นไป
- 2) หน่วยประมวลผล AMD Phenom™ II หรือ Intel® Core™ i3 ขึ้นไป
- 3) หน่วยความจำ 2 GB ขึ้นไป

2.2.4 หน่วยทางกายภาพข้อมูลต่างๆ มีดังนี้

- 1) ระยะทาง (Distance) มีหน่วยเป็น มิลลิเมตร
- 2) เวลา (Time) มีหน่วยเป็น ไมโครวินาที
- 3) ความเร็ว (Speed) มีหน่วยเป็น มิลลิเมตรต่อวินาที
- 4) มุม (Angle) มีหน่วยเป็น เรเดียน

2.2.5 ไลบรารีภาษาจาวา [3]

ไลบรารีของสปีโมชันมี Native Code ในการแปลงภาษาจาวาให้สามารถเรียกใช้ API พื้นฐานของสปีโมชันได้ ไลบรารีโดยรวมมีดังนี้

- 1) เฟรม (Frame) : ข้อมูลทุกอย่างที่อุปกรณ์อ่าน 1 ครั้ง ซึ่งปรับปรุงตลอดเวลา โดยแต่ละเฟรมจะมีข้อมูลของมือ ตัวชี้ และท่าทาง
- 2) มือ (Hand) : เก็บตำแหน่ง มุม ความเร็ว และตัวชี้ที่อยู่ที่มีอยู่นั้นๆ
- 3) ตัวชี้ (Pointer) : แบ่งเป็น 2 ประเภทคือ นิ้วมือ (Finger) และอุปกรณ์ (Tool) ทั้ง 2 ประเภทต่างกันที่ความกว้างของตัวชี้ โดยเก็บตำแหน่ง มุม ความเร็ว ขนาดทั้งความกว้างและความยาว และมือของผู้ใช้ตัวชี้ขึ้นๆ
- 4) ท่าทาง (Gesture) : มี 3 รูปแบบ คือ วงกลม การปิด และการแตะ
 - วงกลม (Circle) : เก็บค่าจุดศูนย์กลาง ทิศทาง ความเร็ว และจำนวนรอบ
 - การปิด (Swipe) : เก็บค่าตำแหน่งจุดเริ่มต้น ตำแหน่งปัจจุบัน ทิศทาง และความเร็วการปิด
 - การแตะ (Screen Tap, Key Tap) : เก็บค่าตำแหน่ง และทิศทางที่แตะ

2.2.6 ข้อจำกัดการใช้งาน [3]

- 1) แสง : แสงที่มีมากเกินไปหรือน้อยเกินไปมีผลในการตรวจจับวัตถุของแสงอินฟราเรดภายในอุปกรณ์ เช่น แสงแดดหรือแสงไฟบางชนิด ซึ่งทำให้ความแม่นยำของตำแหน่งในการตรวจจับได้ลดลงหรืออาจตรวจจับผิดพลาดได้
- 2) ระยะที่จำกัด : ตัวอุปกรณ์นั้นมีระยะที่จะตรวจจับที่จำกัด โดยระยะตามที่สปีโมชันระบุคือ 600 มิลลิเมตร มีลักษณะเป็นรูปทรงกรวยชี้ลง ทำมุมกับพื้น 45 องศาจากพื้น ซึ่งเมื่อผู้ใช้ทำ

การทดลองจริงพบว่าระยะไกลที่สุดที่ทำได้ในที่แสงน้อยนั้นประมาณ 500 มิลลิเมตร ซึ่งแสงมีผลกระทบต่อการทำงานของจอสมควร์ และระยะของวัตถุที่อยู่ใกล้ตัวอุปกรณ์มากเกินไปกว่าระยะ 10 มิลลิเมตรมีผลให้ไม่สามารถตรวจจับได้เช่นเดียวกัน

3) **ลักษณะของวัตถุ** : เนื่องจากอุปกรณ์สีโมชันใช้แสงอินฟราเรดในการตรวจจับวัตถุ จึงไม่สามารถตรวจจับวัตถุที่อยู่ซ้อนกันได้ ซึ่งจะเป็นวัตถุชิ้นเดียวที่ใหญ่ที่สุด นอกจากนั้นวัตถุที่อยู่ใกล้กันมากเกินไปอาจส่งผลให้ถูกตีความเป็นวัตถุเพียงชิ้นเดียวเช่นกัน



รูปที่ 2.9 ตัวอย่างการใช้งานสีโมชัน [12]

ตัวอย่างการรับค่าตำแหน่งนิ้วมือทั้ง 5 จากสีโมชันภาษาจาวาแสดงดังนี้

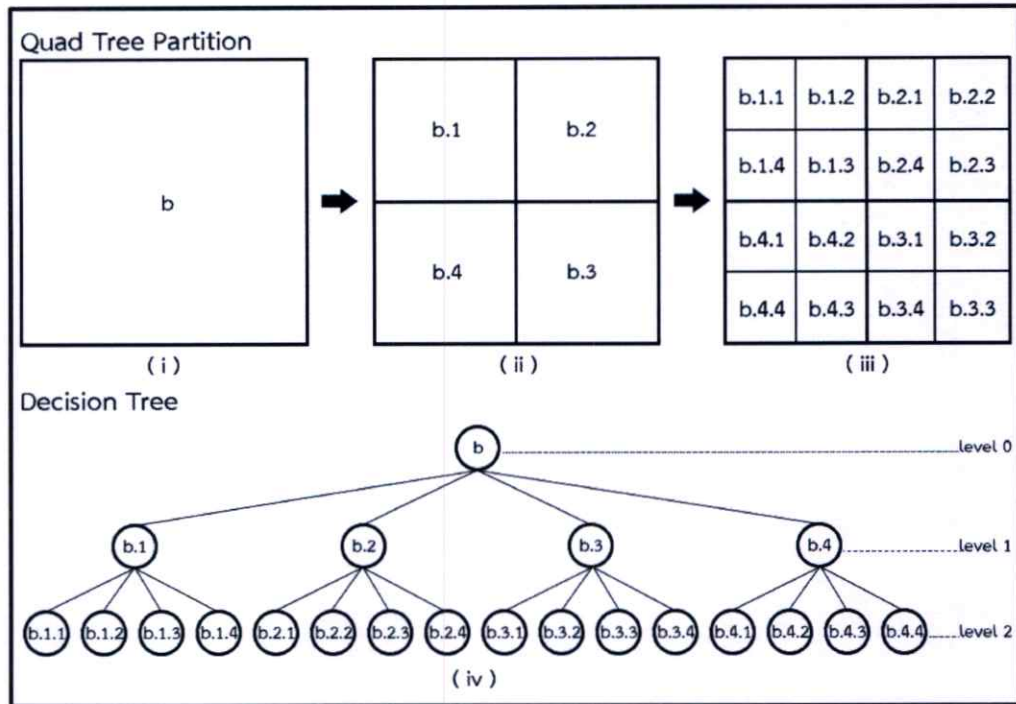
```

Frame frame = controller.frame();
if (!frame.hands().isEmpty()) {
    // Get the first hand
    Hand hand = frame.hands().get(0);
    // Check if the hand has any fingers
    FingerList fingers = hand.fingers();
    if (!fingers.isEmpty()) {
        // Set vector is zero
        Vector pos = Vector.zero();
        for (Finger finger : getFingers())
            System.out.println("X: " + pos.getX() + " Y: " + pos.getY());
    }
}

```

2.3 ควอดทรี (Quad Tree)

ควอดทรีมีลักษณะคล้ายกับ Q-tree เป็นการจำแนกข้อมูลภาพเชิงวัตถุ เพื่ออธิบายหรือนำไปวิเคราะห์ข้อมูลขั้นต่อไป ซึ่งเป็นรูปแบบหนึ่งของ Spatial Data Structures ในการจำแนกข้อมูลภาพเชิงวัตถุ โดยควอดทรีเป็นการแบ่งพื้นที่รูปสี่เหลี่ยมจัตุรัสขนาด 2 มิติ ออกเป็น 4 ส่วนเท่าๆ กันเป็นรูปเครื่องหมายบวกไปเรื่อยๆ ซึ่งจะได้เป็นรูปสี่เหลี่ยมจัตุรัส และนำมาสร้างเป็นต้นไม้ตัดสินใจ ตัวอย่างการแบ่งรูปสี่เหลี่ยมและการนำมาสร้างเป็นต้นไม้ตัดสินใจแสดงดังรูปที่ 2.10



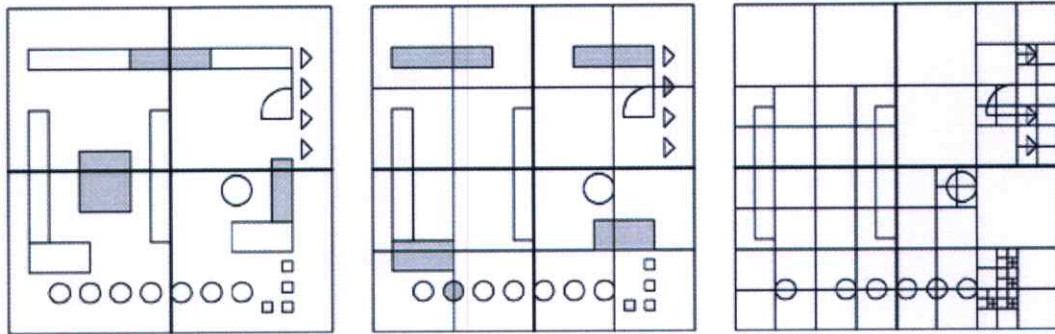
รูปที่ 2.10 การแบ่งควอดทรีและการจำแนกเป็นรูปแบบต้นไม้ตัดสินใจ

จากรูปที่ 2.10 จุดที่ i รูปสี่เหลี่ยมจัตุรัสเริ่มต้น และเป็นโหนดเริ่มต้น (Root Node) ของต้นไม้ตัดสินใจใน level 0 เมื่อทำการแบ่งออกเป็น 4 ส่วนย่อยหรือ 4 โหนด จะได้เป็นดังจุดที่ ii เมื่อเปรียบเทียบกับจุดที่ iv รูปแบบต้นไม้ตัดสินใจ จะเห็นว่าโหนดพ่อแม่ (Parent Node) ประกอบด้วยโหนดลูก (Child Node) 4 โหนด ตามจำนวนการแบ่งรูปสี่เหลี่ยมจัตุรัส จากรูปโหนดลูกคือ Level 1 และ level 2 ซึ่งโหนดใน Level 2 เป็นการแบ่งขั้นสุดท้ายโหนดลูกที่ได้นั้นเรียกว่า Leaf Node ตัวอย่างเช่น ในส่วนรูปแบบต้นไม้ตัดสินใจจะเห็นว่า b.1 มีโหนด b เป็นโหนดพ่อแม่ และมีโหนด b.1.1, b.1.2, b.1.3 และ b.1.4 เป็นโหนดลูก ซึ่งโหนดลูกดังกล่าวเรียกว่า Leaf Node เป็นต้น

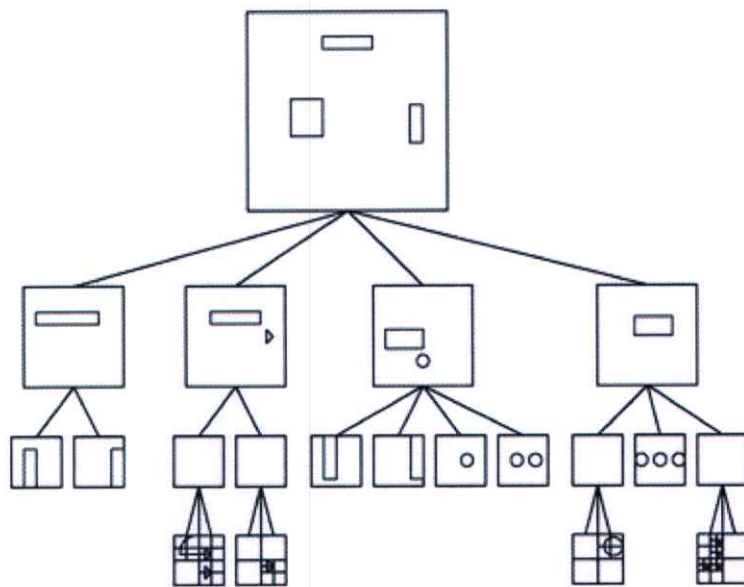
2.3.1 การสร้างควอดทรี

การสร้างควอดทรี [13] เป็นการแบ่งพื้นที่แล้วพิจารณาวัตถุที่ซ้อนทับกับขอบของรูปสี่เหลี่ยมจัตุรัสหรือไม่แสดงดังรูปที่ 2.11 ถ้าหากซ้อนทับให้ทำการแบ่งพื้นที่ย่อยลงไปเรื่อยๆ

จนกระทั่งไม่เหลือวัตถุใดๆ โดยพื้นที่รูปสี่เหลี่ยมนั้นจะเก็บวัตถุนั้นหรือถ้าเป็นต้นไม้ตัดสินใจคือเก็บวัตถุลงในโหนดนั้นๆ แสดงดังรูปที่ 2.12



รูปที่ 2.11 การแบ่งพื้นที่เพื่อวางวัตถุลงในแต่ละพื้นที่ย่อย [14]



รูปที่ 2.12 การจัดเก็บวัตถุลงในแต่ละโหนดของต้นไม้ตัดสินใจ [14]

ตัวอย่าง Pseudo Code การสร้างควอดทรีแสดงดังนี้ [13]

```
void insert (Element e) {
    if (e inside root)
        recursive_insert (root, e);
}

void recursive_insert (Node N, Element e) {
    if (e intersect with horizontal bisector || e intersect with vertical bisector) {
        N->list.push(e);
    } else {
```

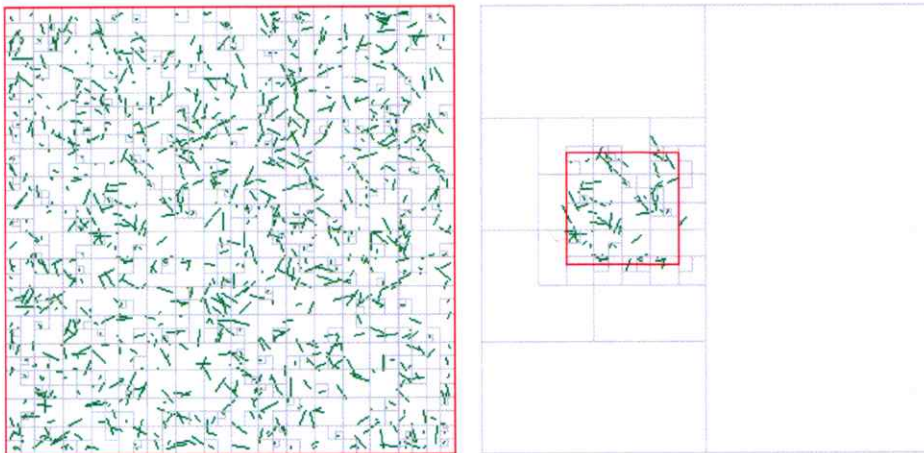
```

for (int i=0; i<4; i++) {
    if (e inside N->child[i]) {
        if (N->child[i] == NULL)
            N->child[i] = new Node();
        recursive_insert (N->child[i], e);
        break;
    }
}
}
}

```

2.3.2 การค้นหาตำแหน่งของวัตถุ

การนำไปควอดทรีไปประยุกต์ใช้สามารถปรับปรุงได้ตามความต้องการและความเหมาะสมของแต่ละงาน ดังที่กล่าวมาข้างต้นเป็นตัวอย่งการสร้างควอดทรีที่แต่ละพื้นที่มีการแบ่งย่อยหรือโหนดมีความยาวไม่เท่ากันตามตำแหน่งการวางของวัตถุ ส่วนการค้นหาเป็นการนำวัตถุที่ต้องการมาค้นหาภายในควอดทรี [13] แสดงดังรูปที่ 2.13



รูปที่ 2.13 ควอดทรีขนาด 300 x 300 พิกเซล บรรจุเส้นตรง 1,000 เส้น
 ด้านซ้าย แสดงเส้น 1,000 เส้นที่อยู่ในกรอบขนาด 300 พิกเซล
 ด้านขวา แสดงเส้น 62 เส้นที่อยู่ในกรอบขนาด 75 พิกเซล [13]

จากรูปที่ 2.13 เมื่อทำการแบ่งพื้นที่แล้วพิจารณาวัตถุนั้นซ้อนทับกับพื้นที่ตำแหน่งนั้นหรือไม่ ถ้าหากพบให้ทำการส่งตำแหน่งค่าของวัตถุในโหนดนั้นๆ ออกมา แต่หากไม่พบวัตถุในโหนดนั้นให้ทำการค้นไปยังโหนดลูกของโหนดนั้นต่อไปจนถึงโหนดสุดท้าย

ตัวอย่าง Pseudo Code การค้นหาตำแหน่งของวัตถุแสดงดังนี้ [13]

```
List query (Rectangle R) {
    recursive_draw (root, R);
}

List recursive_query (Node N, Rectangle R) {
    if (N == NULL) return;
    draw all Elements in N;
    for (int i=0; i<4; i++)
        if (N->child[i] != NULL)
            if(N->child[i] inside R) recursive_draw(N->child[i], R);
}
```

2.3.3 ตัวอย่างงานวิจัยที่นำควอดทรีไปประยุกต์ใช้

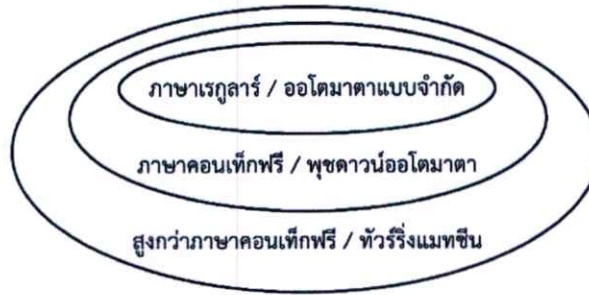
ในปี 2010 Detlev Marpe และคณะ [15] ได้นำเสนองานวิจัยการนำโครงสร้าง Nested Quadtree, Leaf Merging และเทคนิคการปรับปรุงข้อมูล ในการบีบอัดข้อมูลภาพที่มีการเคลื่อนไหว หรือวิดีโอ ซึ่งผลการวิจัยพบว่า Quadtree มีความยืดหยุ่นสูงในการสร้างแบบจำลองการเคลื่อนไหวที่แบ่งพาร์ทิชันตามพื้นที่ รวมทั้งการประมวลผลการตัดกรองของการรวมรูปภายในที่มีประสิทธิภาพ

ในปี 2011 Amitava Chakraborty Sudip Kumar De และ Ranjan Dasgupta [16] นำเสนองานวิจัย Balancing of Quad Tree using Point Pattern Analysis เป็นรูปแบบหนึ่งในการจัดเก็บข้อมูลเชิงพื้นที่ เพื่อเพิ่มประสิทธิภาพในการค้นหาข้อมูล และปัจจัยหนึ่งที่มีผลต่อประสิทธิภาพการค้นหาข้อมูล คือการเพิ่มข้อมูลที่ทำให้ความสูงของต้นไม้ไม่สมดุล ส่งผลให้เวลาในการค้นหาข้อมูลที่สูงขึ้นตามไปด้วย งานวิจัยนี้ได้ออกแบบอัลกอริทึมสร้างความสูงต้นไม้ที่มีความสมดุล

ส่วนใหญ่แล้วควอดทรีนำไปใช้ในการจำแนกข้อมูลภาพเชิงวัตถุ เช่น การหาตำแหน่งต่างๆ บนแผนที่ ความสามารถในการให้รายละเอียดเชิงพื้นที่ของเกม เป็นต้น เพื่อการจัดเก็บข้อมูลอย่างเป็นระบบ และยังช่วยลดจำนวนในการตรวจสอบในพื้นที่อื่นๆ ที่ไม่เกี่ยวข้อง

2.4 ออโตมาตาแบบจำกัด (Finite Automata)

ออโตมาตาหรือแบบจำลองการทำงานของคอมพิวเตอร์ เป็นโครงสร้างการทำงานเชิงลำดับ ที่ถูกนำมาใช้ในการแปลภาษาคอมพิวเตอร์ แบ่งออกเป็น 3 ประเภท ได้แก่ ออโตมาตาแบบจำกัด (Finite Automata) พูชดาวน์ออโตมาตา (Pushdown Automata) และทัวริงแมชชีน (Turning Machine)

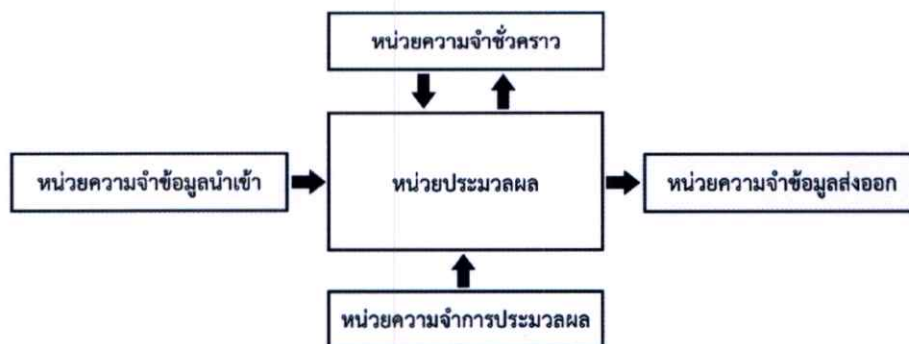


รูปที่ 2.14 ประเภทของภาษาและออโตมาตา

ภาษาพื้นฐานที่สุดในทางคอมพิวเตอร์ คือ ภาษาเรกูลาร์ (Regular Language) เป็นภาษาที่ไม่ใช้หน่วยความจำในการทำงานมากนัก ดังนั้นออโตมาตาแบบจำกัดจึงเหมาะสำหรับการรองรับภาษานี้ ภาษาที่มีความซับซ้อนสูงสุดอีกระดับคือภาษาคอนเท็กซ์ฟรี (Context-free Language) เป็นภาษาที่จำนวนสายอักขระภายในสามารถมีความสัมพันธ์กันได้ เช่น จำนวนตัวอักษร a และ b ภายในข้อมูลมีจำนวนเท่ากัน ซึ่งออโตมาตาที่สามารถรองรับภาษานี้คือ พุชตาวน้อโตมาตา สำหรับภาษาที่มีความซับซ้อนสูงกว่าภาษาภาษาคอนเท็กซ์ฟรี จะใช้ทัวริงแมชชีนรองรับภาษานี้ [17]

จากรูปที่ 2.14 จะเห็นได้ว่าภาษาคอนเท็กซ์ฟรีหรือพุชตาวน้อโตมาตาครอบคลุมภาษาเรกูลาร์หรือออโตมาตาแบบจำกัดแสดงว่าภาษาคอนเท็กซ์ฟรีหรือพุชตาวน้อโตมาต่ายังสามารถรองรับภาษาเรกูลาร์หรือออโตมาตาแบบจำกัดได้ และทัวริงแมชชีนสามารถรองรับภาษาคอนเท็กซ์ฟรีหรือพุชตาวน้อโตมาตา รวมทั้งภาษาเรกูลาร์หรือออโตมาตาแบบจำกัดได้เช่นกัน ดังนั้นจะเห็นว่าทัวริงแมชชีนเป็นภาษาที่มีความสามารถในการคำนวณสูงที่สุด และออโตมาตาแบบจำกัดมีความสามารถในการคำนวณน้อยที่สุด

2.4.1 องค์ประกอบพื้นฐานของออโตมาตา



รูปที่ 2.15 องค์ประกอบพื้นฐานของออโตมาตา

จากรูปที่ 2.15 แสดงความสัมพันธ์ในแต่ละองค์ประกอบพื้นฐานโดยรวม ซึ่งแบ่งออกเป็น 2 ส่วน ดังนี้ [17]

1. หน่วยประมวลผล (CPU) : ทำหน้าที่คำนวณกระบวนการพื้นฐานทางคณิตศาสตร์ เช่น บวก ลบ แอนด์ (AND) ออร์ (OR) เอ็กคลูซีฟ-ออร์ (Exclusive-OR) เป็นต้น ซึ่งทางด้านอโตมาตาเป็นหน่วยการควบคุมสถานะภายในที่เปลี่ยนแปลงไปตามข้อมูลนำเข้า

2. หน่วยความจำ (Memory) : แบ่งออกเป็น 4 ส่วน

1) หน่วยความจำข้อมูลนำเข้า (Input Data Memory) : เป็นข้อมูลสายอักขระที่ผู้ใช้ป้อนเข้าไปยังโปรแกรมและส่งไปยังหน่วยประมวลผล

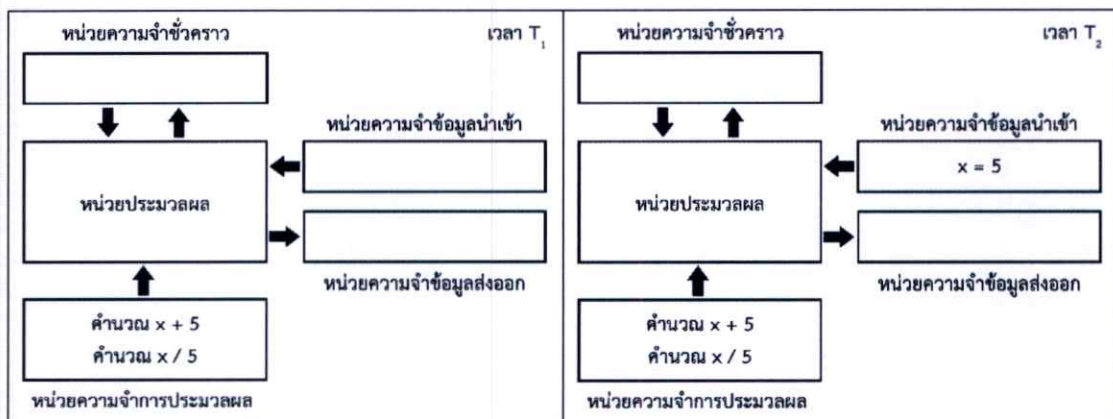
2) หน่วยความจำการประมวลผล (Program Memory) : เป็นข้อมูลการทำงานของส่วนการจัดเก็บซอร์สของโปรแกรม

3) หน่วยความจำชั่วคราว (Temporary Memory) : ทำหน้าที่เก็บผลลัพธ์ที่ได้ในระหว่างการประมวลผล ซึ่งแต่ละประเภทของอโตมาตามีหน่วยความจำชั่วคราวที่แตกต่างกัน โดยอโตมาตาแบบจำกัดนั้นเป็นอโตมาตาที่ไม่มีหน่วยความจำชั่วคราว

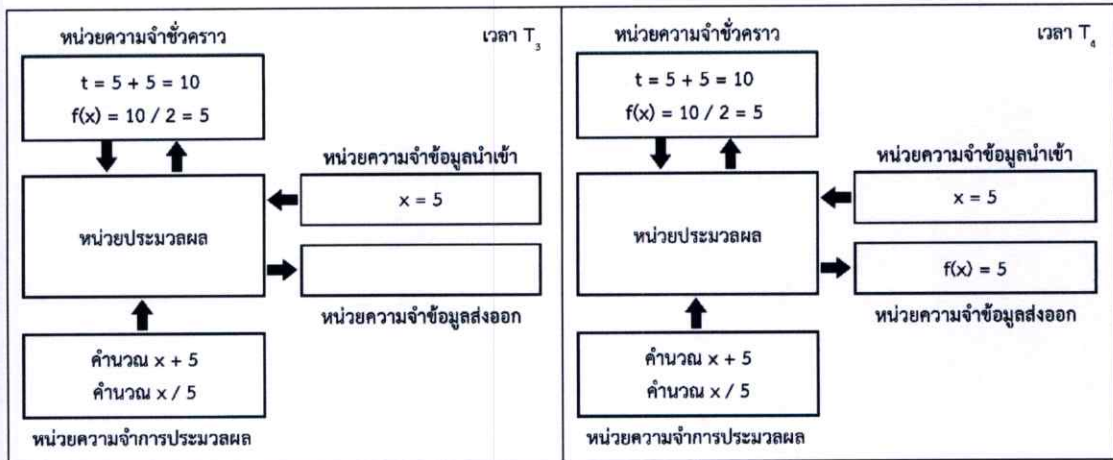
4) หน่วยความจำข้อมูลส่งออก (Output Data Memory) : เป็นผลลัพธ์ที่ได้หลังจากหน่วยประมวลผลทำงานเสร็จสิ้น

ตัวอย่างการทำงานภายในของคอมพิวเตอร์ สามารถอธิบายได้ดังนี้

สมมติให้หน่วยความจำข้อมูลนำเข้าเป็น 5 ซึ่งมาจากผู้ใช้ป้อนเข้ามา และหน่วยการประมวลผลหรือซอร์สโปรแกรมมีการคำนวณผลลัพธ์ของ $f(x) = (x + 5) / 5$ จากนั้นหน่วยประมวลผลจะนำโปรแกรมที่เก็บไว้ในหน่วยความจำการประมวลผล เข้าสู่หน่วยความจำชั่วคราวพร้อมทั้งแทนค่า x ด้วย 5 และคำนวณผลลัพธ์ซึ่งมีค่าเท่ากับ 2 หลังจากนั้นผลลัพธ์ดังกล่าวจะถูกนำมาเก็บลงหน่วยความจำข้อมูลส่งออก เพื่อให้หน่วยแสดงผลนำค่าที่ได้ไปใช้งานต่อไป ดังรูปที่ 2.16 และรูปที่ 2.17



รูปที่ 2.16 การจำลองการคำนวณทางคอมพิวเตอร์ $f(x) = (x + 5) / 5$

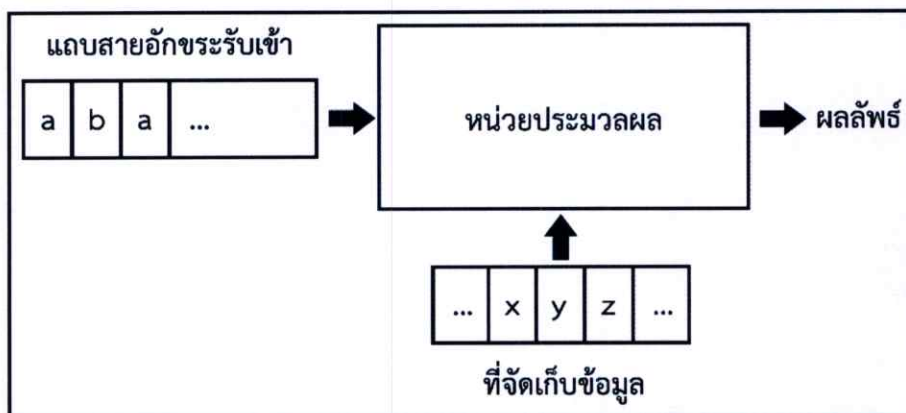


รูปที่ 2.17 การจำลองการคำนวณทางคอมพิวเตอร์ $f(x) = (x + 5) / 5$ (ต่อ)

2.4.2 ส่วนประกอบของออโตมาตา

แสดงส่วนประกอบของออโตมาตาโดยรวมแสดงดังรูปที่ 2.18 แบ่งออกเป็น 3 ส่วน ดังนี้ [17]

1. แถบสายอักขระรับเข้า (Input Tape) : ทำหน้าที่เก็บอักขระ เพื่อส่งไปยังหน่วยประมวลผล
2. หน่วยประมวลผล (Processor) : ทำหน้าที่ควบคุมสถานะภายในที่เปลี่ยนแปลงไปตามข้อมูลนำเข้า
3. ที่เก็บข้อมูล (Storage) : ทำหน้าที่จัดเก็บผลลัพธ์ที่ได้ในระหว่างการคำนวณหรือการประมวลผล



รูปที่ 2.18 ส่วนประกอบของออโตมาตา

ออโตมาตาสามารถนำมาใช้เป็นเครื่องมือตรวจสอบความถูกต้องตามหลักไวยากรณ์ของโปรแกรม ข้อมูลนำเข้าคือซอร์สโปรแกรม ผลลัพธ์ที่ได้นั้นคือใช่หรือไม่ใช่ นอกจากนี้สามารถนำออโตมาตาไปประยุกต์ในการคำนวณทางคณิตศาสตร์ เช่น การหาผลบวก เป็นต้น

2.4.3 นิยามออโตมาตาแบบจำกัดเชิงกำหนด

ออโตมาตาแบบจำกัด [18] แบ่งออกเป็น 2 รูปแบบ คือ ออโตมาตาแบบจำกัดเชิงกำหนด (Deterministic Finite Automata) และออโตมาตาแบบจำกัดเชิงไม่กำหนด (Non-deterministic Finite Automata) ซึ่งในที่นี้จะอธิบายเพียงออโตมาตาแบบจำกัดเชิงกำหนด

นิยามองค์ประกอบออโตมาตาแบบจำกัดเชิงกำหนดแสดงดังสมการที่ (2.1)

$$M = (Q, \Sigma, \delta, q_0, F) \quad (2.1)$$

โดยที่ Q เป็นเซตสถานะต่างๆ (State) ที่เป็นเซตจำกัด

Σ เป็นเซตของอักขระ (Alphabet) ที่เป็นเซตจำกัด

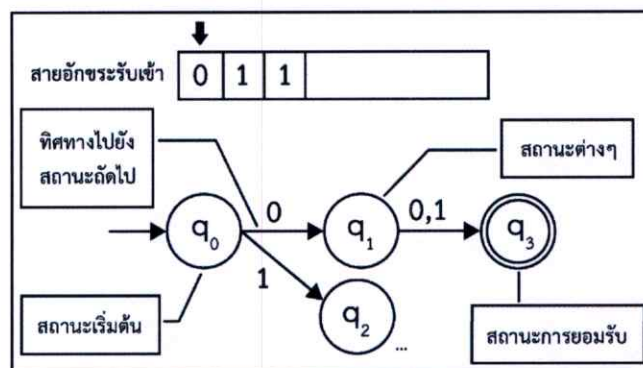
$\delta: Q \times \Sigma \rightarrow Q$ เป็นทศการเคลื่อนไปยังอีกสถานะ (Transition Function)

q_0 เป็นสถานะเริ่มต้น (Initial State) โดยที่ $q_0 \in Q$

F เป็นเซตของสถานะการยอมรับ (Final State) โดยที่ $F \subseteq Q$ และ F เป็นสถานะที่ถือว่าออโตมาตาแบบจำกัดยอมรับสายอักขระที่เป็นข้อมูลนำเข้า

นิยามแผนภาพการเดินทางของออโตมาตาแบบจำกัดเชิงกำหนด (Transitive Diagram) แสดงดังรูปที่ 2.19 โดยแต่ละแผนภาพคือกราฟระบุทิศทางที่ประกอบด้วย

1. มี Q เป็นเซตของจุดทั้งหมด และมีลูกศรชี้เข้าที่จุดเริ่มต้น
2. มีวงกลม 2 วงล้อมรอบทุกจุดใน F
3. มีเส้นเชื่อมจากจุด p ไปยังจุด q โดยมีสัญลักษณ์รับเข้า a เขียนได้เป็น $\delta(p, a) = q$



รูปที่ 2.19 แผนภาพส่วนประกอบของออโตมาตา

จากรูปที่ 2.19 เป็นตัวอย่างการเขียนออโตมาตาจำกัด โดยในการเคลื่อนที่หัวอ่านในสายอักขระแต่ละครั้งออโตมาตาจะอยู่ในเซตของสถานะใดสถานะหนึ่งเสมอ เมื่อเริ่มต้นอ่านข้อมูลนำเข้าเป็น 0 ดังนั้นทิศทางจากสถานะเริ่มต้น (q_0) จึงไปยังสถานะ q_1 จากนั้นเคลื่อนหัวอ่านไป

ทางขวา 1 ช่อง ทำซ้ำจนกระทั่งอ่านข้อมูลนำเข้าเสร็จสิ้น และถ้าหากสถานะปัจจุบันอยู่ในเซตของสถานะการยอมรับ (q_3) แสดงว่าผลลัพธ์ที่ได้คือ ยอมรับข้อมูลนำเข้านี้ และจากสมการที่ (2.1) สามารถเขียนแทนได้เป็น $M = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_0, q_3)$ ซึ่งสามารถเขียนเป็นตารางแสดงค่าสถานะการรับข้อมูลนำเข้าของ 0 และ 1 ได้ดังตารางที่ 2.1

ตารางที่ 2.1 ค่าสถานะการรับข้อมูลนำเข้าของ 0 และ 1

สถานะ / สัญลักษณ์	0	1
q_0	q_1	q_2
q_1	q_3	q_3
q_2	-	-
q_3	-	-

2.5 งานวิจัยที่เกี่ยวข้อง

ในปี 2007 Qutaishat Munib และคณะ [19] ได้นำเสนองานวิจัยเพื่อพัฒนาระบบสำหรับการแปลท่าทางของภาษามือตัวอักษรอเมริกัน (ASL) อัตโนมัติ โดยใช้การประมวลผลภาพแปลงข้อมูลภาพเป็นเวกเตอร์ และนำมาเปรียบเทียบกับเวกเตอร์ที่ใช้เทคนิคเครือข่ายโครงประสาทเทียม (Neural Network) ในการเรียนรู้ชุดท่าทางภาษามือ โดยการหมุนท่าทางภายในภาพนั้นไม่มีผลต่อการสกัดข้อมูล ระบบได้ทดสอบโดยใช้ชุดข้อมูลภาพสัญญาณมือ 300 ตัวอย่าง ผลการทดสอบระบบสามารถรับรู้ท่าทางภาษามือและมีความถูกต้อง 92.3 เปอร์เซ็นต์

ในปี 2008 Suwanee Phitakwinai และคณะ [20] ได้นำเสนองานวิจัย Thai Sign Language Translation Using Fuzzy C-Means and Scale Invariant Feature Transform เป็นการแปลงภาษามือเป็นตัวอักษรจำนวน 15 ตัวและคำจำนวน 10 คำ โดยใช้กล้องวิดีโอเป็นข้อมูลนำเข้าในการประมวลผล ซึ่งอัลกอริทึมที่นำมาใช้คือ FCM และ SIFT ผลการทดลองพบว่าระบบสามารถแปลงเป็นตัวอักษรได้ถูกต้อง 82.19 เปอร์เซ็นต์ และแปลเป็นคำได้ถูกต้อง 55.08 เปอร์เซ็นต์ โดยที่เพื่อนบ้านใกล้เป็น 3 และ SIFT threshold เป็น 0.7

ในปี 2014 Veronica Yenquenida Flamenco Cordova [21] ได้นำเสนอการวิเคราะห์ตำแหน่งและทิศทางแต่ละนิ้วมือของผู้ใช้ว่ามีความถูกต้องหรือไม่ จำนวน 12 ตัวอักษร โดยข้อมูลนำเข้าเป็นภาพนิ่งภาษามือที่สวมลูกโป่งสีที่แตกต่างกันแต่ละปลายนิ้วมือ จากนั้นนำภาพนิ่งมาแยกประเภทนิ้วมือจากสีของลูกโป่งแต่ละนิ้ว และหาจุดกึ่งกลางแต่ละปลายนิ้วมือ เพื่อวิเคราะห์ความสัมพันธ์ระหว่างนิ้วมือ ได้แก่ ระยะห่างระหว่างนิ้ว มุมระหว่างนิ้ว และจุดศูนย์กลาง จากนั้นทำการหาค่าความผิดพลาดกับภาพต้นแบบที่เป็นมาตรฐาน ถ้าผลลัพธ์ที่ได้ไม่ถูกต้อง ทำการตรวจสอบตำแหน่งการวางนิ้วควรเลื่อนไปยังทิศทางใดโดยใช้วิธีการ Eight Nearest Neighbor และหาระยะ

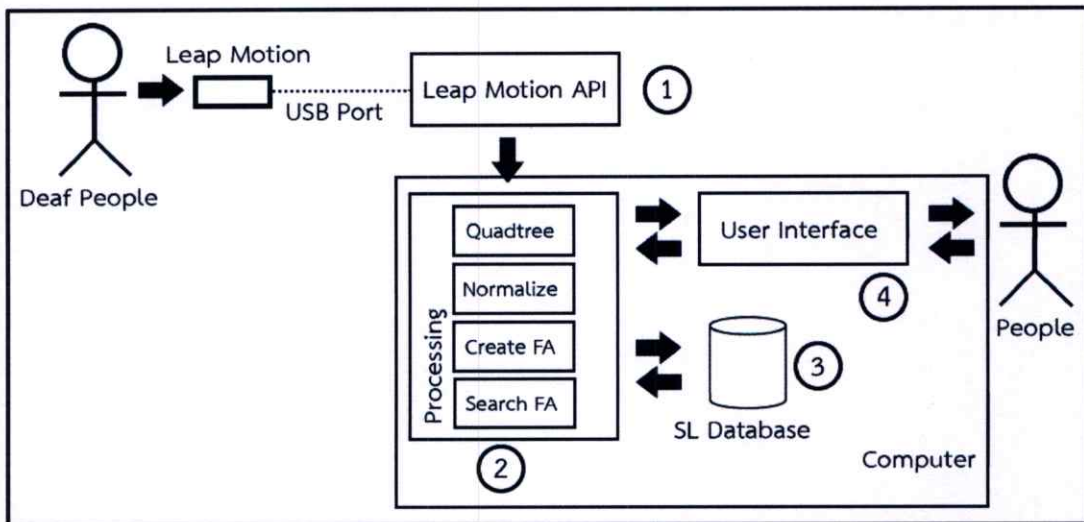
การเคลื่อนนิ้วไปยังตำแหน่งที่ถูกต้อง นอกจากนี้ยังมีการตรวจสอบความถูกต้องการทำงานของโปรแกรม โดยข้อมูลในการทดสอบเป็นภาพภาษามือที่ถูกต้องและไม่ถูกต้อง และผลการทดสอบการทำงานของโปรแกรมนั้นมีความถูกต้อง

ในปี 2014 วิสันต์ ตั้งวงษ์เจริญ และจิรวรรธ คุ้มศรี [22] ได้นำเสนองานวิจัยการแปลงภาษามือเป็นตัวอักษรสำหรับผู้มีความบกพร่องทางการได้ยินหรือการสื่อความหมายสามารถนำไปใช้ในการสื่อสารกับบุคคลทั่วไปได้สะดวก โดยการรับตำแหน่งปลายนิ้วมือทั้ง 5 ผ่านอุปกรณ์สปีโมเซน และนำตำแหน่งปลายนิ้วมือมาวิเคราะห์ด้วยเทคนิคควอดตริ ผลลัพธ์ที่ได้เป็นรหัสตัวเลขรูปแบบภาษามือหนึ่งตัวอักษร 3 หลัก จำนวน 42 ตัวอักษร จำแนกรูปแบบการสะกดนิ้วมือแทนหนึ่งตัวอักษรออกเป็น 3 รูปแบบ คือ แบบที่ 1 เป็นการสะกดนิ้วมือ 1 จังหวะ มีจำนวน 18 ตัวอักษร แบบที่ 2 เป็นการสะกดนิ้วมือ 2 จังหวะ จำนวน 23 ตัวอักษร และแบบที่ 3 เป็นการสะกดนิ้วมือ 3 จังหวะ จำนวน 3 ตัวอักษร

บทที่ 3

วิธีการดำเนินงานวิจัย

งานวิจัยนี้เป็นการนำเสนอการแปลงภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทยจำนวน 44 ตัวอักษร โดยใช้อุปกรณ์สปีโมชันเป็นเซ็นเซอร์ในการตรวจจับตำแหน่งนิ้วมือ และตำแหน่งกลางฝ่ามือ เพื่อปรับค่าต่างๆ จากอุปกรณ์สปีโมชันให้เหมาะสมก่อนนำไปประมวลผลต่อไป จากการศึกษาผู้วิจัยได้นำเทคนิคควอดทรีมาประยุกต์ใช้ในการแปลงท่าทางภาษามือแต่ละนิ้วเป็นรหัสตัวเลขจำนวน 3 หลัก และนำมาสร้างเป็นออโตมาตาแบบจำกัดต้นแบบ เพื่อใช้ในการวิเคราะห์ท่าทางภาษามือเป็นตัวอักษรภาษาไทย ซึ่งภาพรวมการทำงานของโปรแกรมแสดงดังรูปที่ 3.1



รูปที่ 3.1 แผนภาพการทำงานของโปรแกรม

จากรูปที่ 3.1 เป็นโครงสร้างการทำงานของโปรแกรม ซึ่งแบ่งออกเป็น 4 ส่วนดังนี้

1. **การทำงานส่วน Leap Motion API** เป็นไลบรารีภาษาจาวาของอุปกรณ์สปีโมชันทำหน้าที่ที่ติดต่อบริหารเหตุการณ์ต่างๆ ที่เกิดขึ้นกับอุปกรณ์ตรวจจับการเคลื่อนไหวของมือและนิ้ว และส่งข้อมูลนั้นไปยังการทำงานส่วน Processing ผ่าน USB Port เพื่อนำข้อมูลไปเป็นส่วนหนึ่งภายในโปรแกรม และแสดงผลบนหน้าจอของผู้ใช้งาน

2. **การทำงานส่วน Processing** เป็นส่วนที่ประมวลผลข้อมูลที่ได้รับจากอุปกรณ์สปีโมชันผ่าน API พร้อมทั้งดึงค่าข้อมูลต้นแบบออโตมาตาแบบจำกัดภาษามือตัวอักษรภาษาไทยจากส่วนของ SL Database หลังจากได้ข้อมูลแล้วจึงประมวลผล โดยจำแนกแต่ละส่วนดังนี้

- การประมวลผลข้อมูลตำแหน่งการเคลื่อนไหวของนิ้วมือและตำแหน่งกลางฝ่ามือ จากอุปกรณ์สปีโมชันที่ติดต่อกับผู้ใช้งานผ่าน API มาปรับค่าให้เหมาะสำหรับการวิเคราะห์

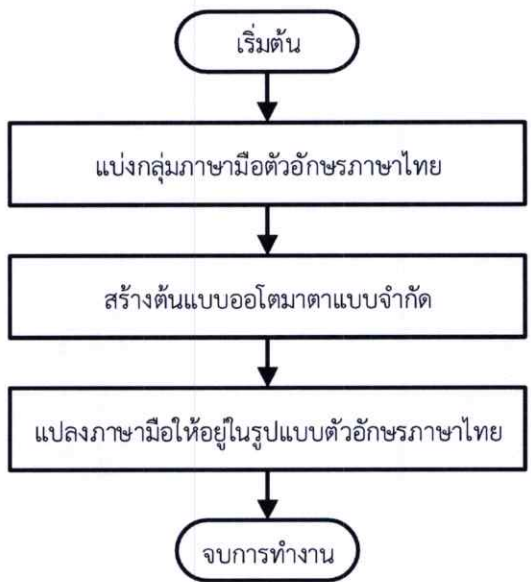
- การประมวลผลตำแหน่งของนิ้วมือที่แปลงเป็นรหัสตัวเลขแต่นิ้วมือโดยใช้เทคนิคควอดทรี จากนั้นทำการปรับค่าให้เหมาะสม และค้นหาผลลัพธ์ตัวอักษรภาษาไทยจากต้นแบบอัตโนมัติแบบจำกัดภาษามือตัวอักษรภาษาไทย

3. การทำงานส่วน SL Database มีหน้าที่ในการจัดเก็บข้อมูลต้นแบบตำแหน่งอักขระภาษามือ ซึ่งประกอบด้วย ตัวอักษร และข้อมูลรหัสจำนวน 3 หลักของแต่ละนิ้วมือ

4. การทำงานส่วน User Interface เป็นส่วนที่ทำหน้าที่ติดต่อกับผู้ใช้งานโดยตรง โดยแสดงตำแหน่งการเคลื่อนไหวจากการเคลื่อนไหวมือและนิ้วมือผู้ใช้จากอุปกรณ์สัมผัส และแสดงผลการวิเคราะห์จากส่วน Processing ออกมาเป็นตัวอักษร

3.1 ขั้นตอนการดำเนินงานวิจัย

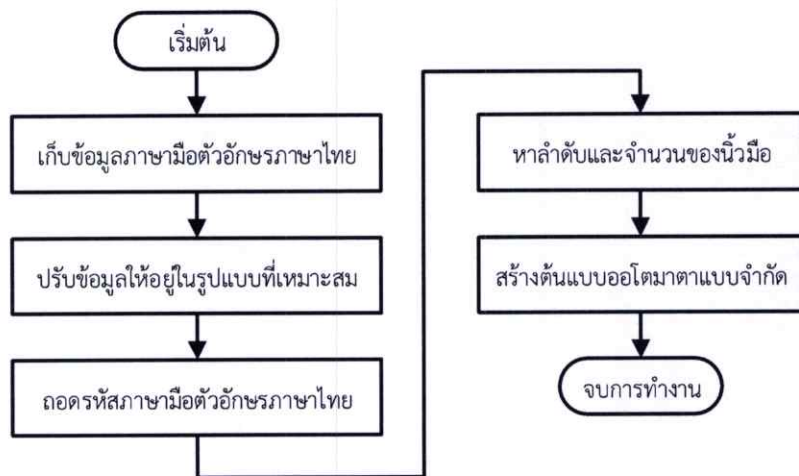
ผู้วิจัยได้จัดแบ่งกลุ่มภาษามือตัวอักษรภาษาไทยจำนวน 3 รูปแบบ จากนั้นแบ่งการทำงานออกเป็น 2 ส่วน คือ ส่วนการสร้างต้นแบบอัตโนมัติแบบจำกัดภาษามือตัวอักษรภาษาไทย และส่วนการแปลงภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทยแสดงดังรูปที่ 3.2



รูปที่ 3.2 ภาพรวมขั้นตอนการดำเนินงานวิจัย

ส่วนของการสร้างต้นแบบอัตโนมัติแบบจำกัดภาษามือตัวอักษรภาษาไทยนั้น ผู้วิจัยได้เก็บตัวอย่างท่าทางภาษามือตัวอักษรภาษาไทยจากผู้เชี่ยวชาญทางด้านภาษามือตัวอักษรภาษาไทย เมื่อเก็บรวบรวมข้อมูลในรูปแบบตำแหน่งแกน X และ Y เรียบร้อย จึงนำข้อมูลต่างๆ ในรูปแบบตำแหน่งแกน X และ Y มาปรับค่าข้อมูลให้อยู่ในรูปแบบที่เหมาะสม เข้ารหัสภาษามือตัวอักษรภาษาไทย ซึ่งผลลัพธ์ที่ได้ในแต่ละนิ้วมือเป็นตัวเลขจำนวน 3 หลัก หลังจากนั้นจึงนำรหัสทั้ง 3 หลักมา

หาลำดับและจำนวนของนิ้วมือสำหรับการสร้างเป็นต้นแบบอโตมาตาแบบจำกัดภาษามือตัวอักษรภาษาไทยแสดงดังรูปที่ 3.3



รูปที่ 3.3 ขั้นตอนการสร้างต้นแบบอโตมาตาแบบจำกัด

ส่วนการแปลงภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทย เมื่อผู้ใช้แสดงท่าทางภาษามือผ่านอุปกรณ์ลิฟโมชันส่งมายังโปรแกรมแล้วจึงนำข้อมูลต่างๆ ในรูปแบบตำแหน่งแกน X และ Y มาปรับค่าข้อมูลให้อยู่ในรูปแบบที่เหมาะสม จากนั้นจึงนำตำแหน่งแกน X และ Y ที่ปรับค่าแล้วมาเข้ารหัสภาษามือตัวอักษรภาษาไทย และทำการค้นหาผลลัพธ์ 1 ตัวอักษรจากต้นแบบอโตมาตาแบบจำกัดภาษามือตัวอักษรภาษาไทยจากฐานข้อมูลแสดงดังรูปที่ 3.4



รูปที่ 3.4 ขั้นตอนการแปลงภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทย

3.2 การจัดแบ่งกลุ่มภาษามือตัวอักษรภาษาไทย

ผู้วิจัยได้จำแนกรูปแบบการสะกดนิ้วมือแทนหนึ่งตัวอักษรจำนวน 42 ตัวอักษร แสดงดังรูปที่ 2.1 เป็น 3 รูปแบบตามจังหวะการเคลื่อนไหว ได้แก่ แบบที่ 1 เป็นการสะกดนิ้วมือ 1 จังหวะ มีจำนวน 15 ตัวอักษร แบบที่ 2 เป็นการสะกดนิ้วมือ 2 จังหวะ จำนวน 24 ตัวอักษร และแบบที่ 3 เป็นการสะกดนิ้วมือ 3 จังหวะ จำนวน 3 ตัวอักษร [22] แสดงดังรูปที่ 3.5

แบบสะกดนิ้วมือไทย (Thai Fingerspelling)

แบ่งตามจังหวะการเคลื่อนไหว

1. การเคลื่อนไหว 1 จังหวะ

ก	ต	ส	พ	ห	บ	ร	ว	ด	ฟ	ล	ย	ม	น	อ

2. การเคลื่อนไหว 2 จังหวะ

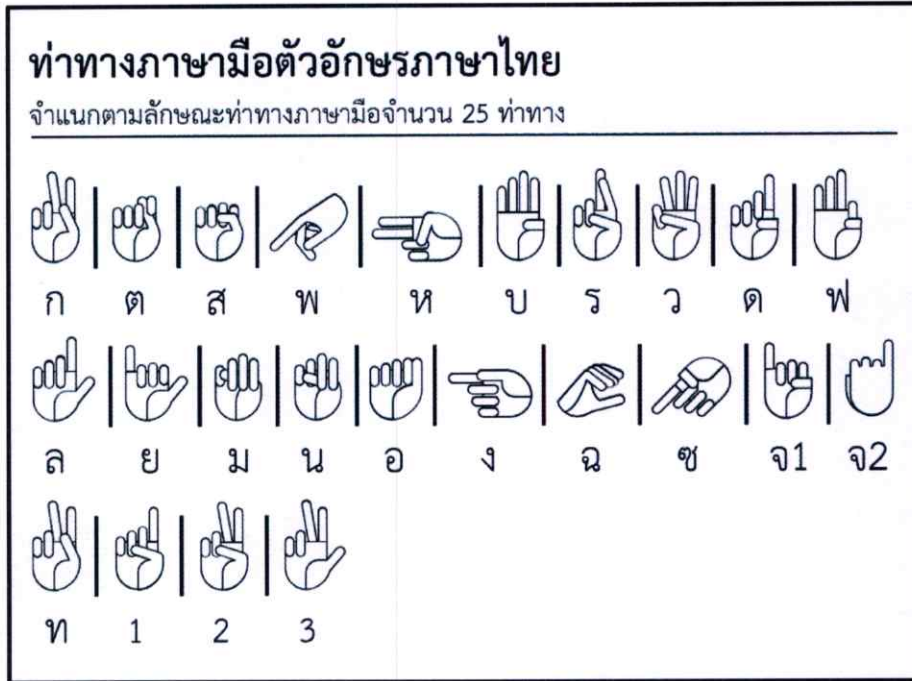
ข	ค	ฃ	ฅ	ง	ฉ	ซ	ฌ	ฌ	ฌ
ช	ซ	ป	ผ	ภ	ย	ฎ	ฬ		
ช	ซ	ป	ผ	ภ	ย	ฎ	ฬ		
ฬ	จ	ญ	ณ	ง	ท	ฉ			
ฬ	จ	ญ	ณ	ง	ท	ฉ			

3. การเคลื่อนไหว 3 จังหวะ

ฐ	ช	ณ			

รูปที่ 3.5 แผนภาพตัวอักษรภาษาไทย จำแนกตามจังหวะการเคลื่อนไหว [22]

เนื่องจากตัวอักษรในจังหวะที่ 2 และจังหวะที่ 3 เกิดจากการผสมตัวอักษรหรือตัวเลขเข้าด้วยตัวอย่างดังรูปที่ 2.4 ตัวอักษร “ข” ที่เกิดจากการผสมตัวอักษร “ค” หรือ “ก” เข้ากับตัวเลข “1” ดังนั้นจึงได้ลักษณะท่าทางการแสดงภาษามือตัวอักษรภาษาไทยจำนวน 24 ท่าทางดังรูปที่ 3.6



รูปที่ 3.6 ท่าทางภาษามือตัวอักษรภาษาไทยจำแนกตามลักษณะท่าทางภาษามือ

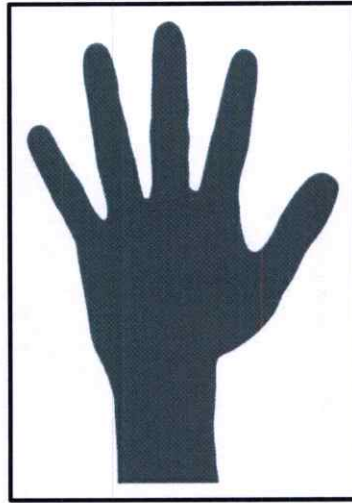
3.3 การเก็บข้อมูลต้นแบบภาษามือตัวอักษรภาษาไทย

การจัดเก็บข้อมูลในงานวิจัยนี้ใช้อุปกรณ์สปีพโมชันเชื่อมต่อกับคอมพิวเตอร์ผ่าน USB 3.0 ดังรูปที่ 3.1 โดยข้อมูลที่รับค่าผ่านอุปกรณ์สปีพโมชันที่นำมาใช้ในการวิจัยมีดังนี้

ในส่วนของการสร้างต้นแบบอโตมาตาแบบจำกัดภาษามือตัวอักษรภาษาไทย ได้แก่ ตำแหน่งแกน X และ Y ของปลายนิ้วมือทั้ง 5 นิ้ว และตำแหน่งแกน X และ Y ที่กลางฝ่ามือ โดยจัดเก็บท่าทางภาษามือตัวอักษรภาษาไทยจำนวน 24 ท่าทางดังรูปที่ 3.6

การแปลภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทยการเก็บข้อมูลแต่ละนิ้วมือนั้นขึ้นอยู่กับผลลัพธ์ของการหาลำดับ และจำนวนของนิ้วมือในขั้นตอนการสร้างต้นแบบอโตมาตาแบบจำกัดภาษามือตัวอักษรภาษาไทย ตัวอย่างเช่น ผลลัพธ์การหาลำดับและจำนวนของนิ้วมือ ได้แก่ นิ้วกลาง นิ้วโป้ง และนิ้วชี้ จำนวน 3 นิ้ว เป็นต้น และตำแหน่งแกน X และ Y ที่กลางฝ่ามือ เพื่อใช้ในการปรับข้อมูลให้อยู่ในรูปแบบที่เหมาะสมต่อไป

การจัดเก็บข้อมูลต้นแบบหรือการแปลงภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทยนั้น เริ่มต้นจะต้องแสดงท่าทางมือแสดงดังรูปที่ 3.7 โดยจัดเก็บตำแหน่งแกน X และ Y ของปลายนิ้วชี้ และกลางฝ่ามือ เป็นเวลา 3 วินาที เพื่อทำการจัดเก็บและปรับขนาดของมือทั้งต้นแบบและมือของผู้ทำการทดสอบให้มีขนาดเท่ากัน หลังจากแสดงท่าทางมือเริ่มต้นแล้วจึงจัดเก็บการแสดงแต่ละท่าทางภาษามือ โดยจับตำแหน่งแกน X และ Y ของปลายนิ้วมือที่ได้กล่าวข้างต้นเป็นเวลา 5 วินาที



รูปที่ 3.7 ลักษณะมือก่อนเริ่มต้นการจัดเก็บข้อมูล

3.4 การปรับข้อมูลให้อยู่ในรูปแบบที่เหมาะสม

การปรับค่าข้อมูลให้อยู่ในรูปแบบที่เหมาะสมนั้น เป็นการปรับขนาดของมือทั้งต้นแบบและมือของผู้ทำการทดสอบให้มีขนาดเท่ากัน และกำหนดตำแหน่งกลางฝ่ามือเดียวกัน เพื่อความถูกต้องในการแปลงภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทย โดยสมการที่ผู้วิจัยได้นำมาใช้แสดงดังนี้

สมการการหาระยะห่างระหว่างจุดสองจุดในแนวเส้นตรง (Euclidean Distance) [23] แสดงดังสมการที่ (3.1)

$$distance = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} \quad (3.1)$$

โดยที่ *distance* คือ ระยะห่างระหว่างปลายนิ้วกลางกับกลางฝ่ามือ

X_1, Y_1 คือ แกน X และ Y ปลายนิ้วกลาง

X_2, Y_2 คือ แกน X และ Y กลางฝ่ามือ

เมื่อได้ค่าระยะห่างระหว่างปลายนิ้วกลางกับกลางฝ่ามือแล้ว จากนั้นนำตำแหน่งแกน X และ Y ที่ได้มาหาอัตราส่วนดังสมการที่ (3.2) เพื่อนำผลลัพธ์จากสมการที่ (3.2) มาปรับตำแหน่งแกน X ดังสมการที่ (3.3) และแกน Y ดังสมการที่ (3.4) ของแต่ละนิ้วมือ เพื่อแก้ปัญหาขนาดของมือผู้ใช้ที่ต่างกัน และลดความผิดพลาดในการแปลงภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทย

$$ratio = \frac{distance_1}{distance_2} \quad (3.2)$$

โดยที่ $ratio$ คือ อัตราส่วนระหว่างภาษามือต้นแบบกับการทดสอบ

$distance_1$ คือ ระยะห่างระหว่างปลายนิ้วกลางกับกลางฝ่ามือของภาษามือต้นแบบ

$distance_2$ คือ ระยะห่างระหว่างปลายนิ้วกลางกับกลางฝ่ามือของภาษามือการทดสอบ

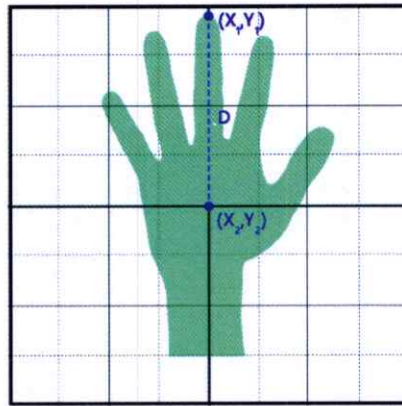
$$X_{post} = X_{pre} * ratio \quad (3.3)$$

$$Y_{post} = Y_{pre} * ratio \quad (3.4)$$

โดยที่ X_{post}, Y_{post} คือ ตำแหน่งแกน X และ Y ที่หลังการปรับอัตราส่วน

X_{pre}, Y_{pre} คือ ตำแหน่งแกน X และ Y ที่ก่อนการปรับอัตราส่วน

$ratio$ คือ อัตราส่วนระหว่างภาษามือต้นแบบกับการทดสอบ



รูปที่ 3.8 ตำแหน่งกลางมือบนตารางควอดทรี

นอกจากการปรับตำแหน่งของนิ้วมือแล้วยังมีการเลื่อนตำแหน่งของนิ้วมือเพื่อการแสดงบนตารางควอดทรีที่เหมาะสม ซึ่งผู้วิจัยได้กำหนดตำแหน่งกลางฝ่ามือเป็นจุดคงที่เสมอเพื่ออ้างอิงไปยังนิ้วมืออื่นๆ โดยกำหนดให้ตำแหน่งกลางฝ่ามืออยู่ในจุดกึ่งกลางแนวแกน X และ Y ของตารางควอดทรีดังรูปที่ 3.8 ทำให้ตำแหน่งนิ้วมืออื่นๆ เปลี่ยนตามไปด้วย ซึ่งสามารถครอบคลุมทั้งมือแม้ว่ามือมีการเคลื่อนย้ายตำแหน่ง

การเลื่อนตำแหน่งของนิ้วมือนั้นทำโดยการหารระยะห่างระหว่างตำแหน่งกลางฝ่ามือเดิม (X_1, Y_1) และตำแหน่งกลางมือหลังจากเลื่อนไปยังจุดกึ่งกลางแนวแกน X และ Y ของตารางควอดทรีแล้ว (X_2, Y_2) ดังสมการที่ (3.1) เมื่อหารระยะการเลื่อนตำแหน่งกลางฝ่ามือแล้ว ทำการปรับตำแหน่งแกน X และ Y แต่ละนิ้วมื่อดังสมการที่ (3.5) และสมการที่ (3.6)

$$X_{post} = X_{pre} + distance \quad (3.5)$$

$$Y_{post} = Y_{pre} + distance \quad (3.6)$$

โดยที่ X_{post}, Y_{post} คือ ตำแหน่งแกน X และ Y ที่หลังการเลื่อนตำแหน่งนิ้วมือ

X_{pre}, Y_{pre} คือ ตำแหน่งแกน X และ Y ที่ก่อนการเลื่อนตำแหน่งนิ้วมือ

$distance$ คือ ระยะการเลื่อนตำแหน่งกลางฝ่ามือไปยังจุดกึ่งกลางตารางคอร์ดทรี

การสร้างต้นแบบอโตมาตาแบบจำกัดภาษามือตัวอักษรภาษาไทย เริ่มต้นจากการหาค่าระยะห่างระหว่างปลายนิ้วกลางกับกลางฝ่ามือเริ่มต้นดังสมการที่ (3.1) จากนั้นทำการเลื่อนตำแหน่งแกน X ดังสมการที่ (3.5) และแกน Y ดังสมการที่ (3.6) เพื่อการแปลงภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทยที่ถูกต้อง

การปรับค่าข้อมูลในอยู่ในรูปแบบที่เหมาะสมในขั้นตอนการแปลงภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทยนั้น นำค่าตำแหน่งแกน X และ Y ของปลายนิ้วกลางกับตำแหน่งแกน X และ Y กลางฝ่ามือมาหารระยะห่างระหว่างจุด 2 จุดดังสมการที่ (3.1) จากนั้นหาอัตราส่วนระยะห่างระหว่างปลายนิ้วกลางกับกลางฝ่ามือของต้นแบบกับการทดสอบดังสมการที่ (3.2) เมื่อได้อัตราส่วนแล้วทำการปรับค่าตำแหน่งแกน X ดังสมการที่ (3.3) และแกน Y ดังสมการที่ (3.4) ขั้นตอนต่อไปทำการเลื่อนตำแหน่งแกน X ดังสมการที่ (3.5) และแกน Y ดังสมการที่ (3.6)

3.5 อัลกอริทึมในการเข้ารหัสภาษามือตัวอักษรภาษาไทย

อัลกอริทึมในการเข้ารหัสมือตัวอักษรภาษาไทยโดยใช้เทคนิคคอร์ดทรี เป็นการเข้ารหัสท่าทางของนิ้วมือ 1 นิ้ว โดยมีข้อมูลนำเข้าเป็นตำแหน่งแกน X และ Y ของแต่ละนิ้วมือ จำนวนรหัสตัวอักษร และขนาดตารางสี่เหลี่ยมจัตุรัสของคอร์ดทรี ส่วนผลลัพธ์ที่ได้เป็นรหัสตัวเลขจำนวน 3 หลักของแต่ละนิ้วมือ ซึ่งอัลกอริทึมในการเข้ารหัสมือตัวอักษรภาษาไทยแสดงดังรูปที่ 3.9

อัลกอริทึมในการเข้ารหัสรูปแบบภาษามือแสดงดังรูปที่ 3.9 เริ่มต้นทำการแบ่งขนาดแกน X และ Y ของตารางคอร์ดทรี เป็น 2 ส่วนเท่าๆ กัน จากนั้นพิจารณาแกน X ถ้าตำแหน่งแกน X ของนิ้วมือนั้นอยู่ทางด้านขวามือ ให้ ex เก็บค่าเป็น 2 และ 3 ถ้าตำแหน่งแกน X ของนิ้วมือนั้นอยู่ทางด้านซ้ายมือ ให้ ex เก็บค่าเป็น 1 และ 4 พร้อมกับเก็บขนาดแกน X และ Y ที่ทำการแบ่งตารางคอร์ดทรี เป็น 2 ส่วน เพื่อพิจารณาการแบ่งขนาดแกน X และ Y ย่อยต่อไป จากนั้นพิจารณาในแกน Y ถ้าตำแหน่ง Y ของนิ้วมือนั้นอยู่ทางด้านล่าง ให้ ex เก็บค่าเป็น 3 และ 4 ถ้าตำแหน่งแกน Y ของนิ้วมือนั้นอยู่ทางด้านบน ให้ ex เก็บค่าเป็น 1 และ 2 เมื่อได้ค่า ex และ ey จากนั้นทำการตรวจสอบค่าภายใน ex และ ey โดยเก็บค่าภายใน ex และ ey ที่เหมือนกันลงตัวแปร e ทำซ้ำการแบ่งตารางคอร์ดทรีย่อยต่อไปตามกระบวนการข้างต้นจำนวน 2 ครั้ง และเวลาของอัลกอริทึมในการประมวลเป็น $O(n)$

Algorithm: Encode position THSL

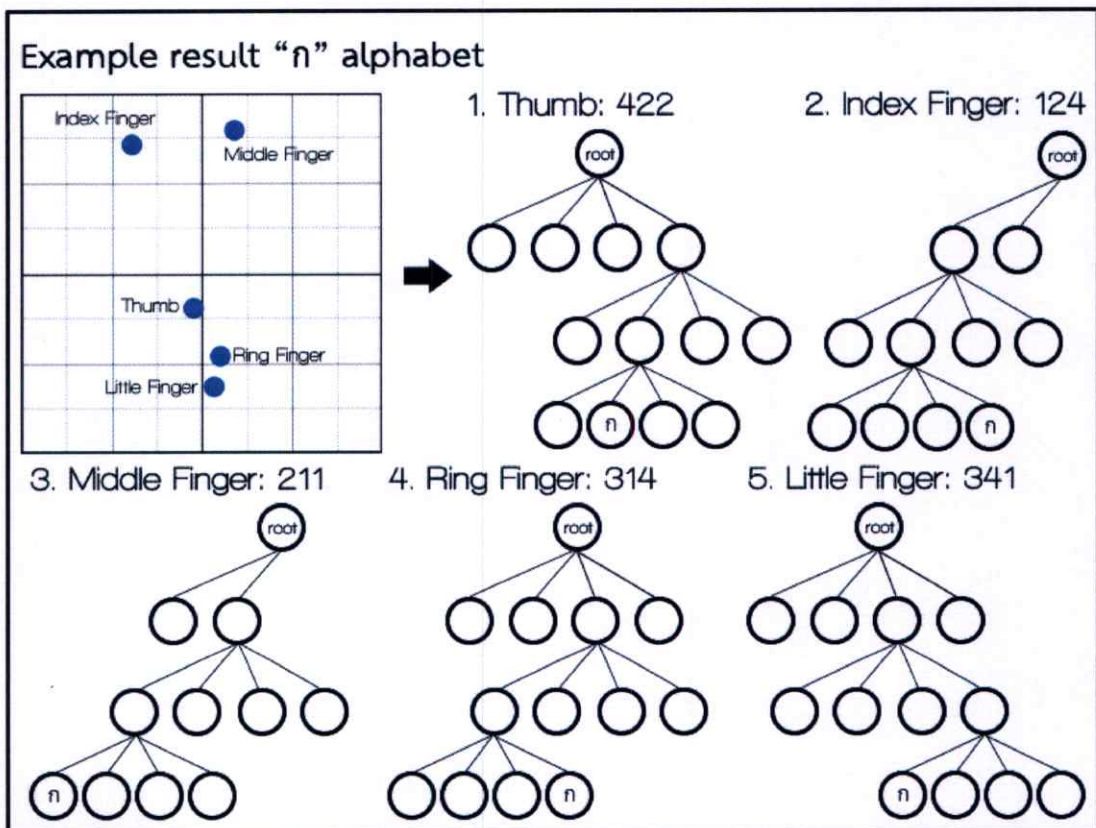
```

1.  ex = { 0, 0 }, ey = { 0, 0 }
2.  For level = 0 to N and e == null
3.    size table = size table/2
4.    If px > size table, Then
5.      ex = { 2, 3 }
6.    Else
7.      ex = { 1, 4 }
8.    End if else
9.    If py > size table, Then
10.   ey = { 2, 3 }
11.  Else
12.   ey = { 1, 4 }
13.  End if else
14.  e [level] = duplicate value in array ex and ey
15. End for
16. Return e [level]

```

รูปที่ 3.9 อัลกอริทึมการเข้ารหัสรูปแบบภาษามือ

เมื่อทำการประมวลผลการแปลงตำแหน่งนิ้วมือทั้ง 3 จะได้รูปแบบต้นไม้ตัดสินใจ และรหัสตัวเลขภาษามือ 3 หลัก แสดงตัวอย่างดังรูปที่ 3.10



รูปที่ 3.10 ตัวอย่างต้นไม้ตัดสินใจตัวอักษร "ก" ของนิ้วมือทั้ง 5 นิ้ว [22]

จากรูปที่ 3.10 เป็นตัวอย่างแสดงตำแหน่งนิ้วมือจากการแสดงท่าทางภาษามือตัวอักษร “ก” และต้นแบบต้นไม้ตัดสินใจ จากการแบ่งพื้นที่ตารางควอดรีออกเป็น 4 ส่วนย่อยจนกระทั่งครบจำนวน 3 ครั้ง โดยผู้วิจัยได้พิจารณาตำแหน่งนิ้วมือบนตารางควอดรี 5 ตำแหน่งตามประเภตนิ้วมือ ซึ่งแต่ละนิ้วมือเก็บข้อมูลเป็นรหัสตัวเลขจำนวน 3 หลัก

3.6 อัลกอริทึมการหาลำดับและจำนวนของนิ้วมือ

การหาลำดับและจำนวนของนิ้วมือ เป็นขั้นตอนหนึ่งของการสร้างต้นแบบอโตมาตาแบบจำกัดภาษามือตัวอักษรภาษาไทย เพื่อลดการจัดเก็บข้อมูลและเพิ่มประสิทธิภาพการประมวลผลในการแปลงภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทย โดยข้อมูลนำเข้าประกอบด้วยตำแหน่งแกน x และ y ของนิ้วมือทั้ง 5 นิ้ว และผลลัพธ์ที่ได้เป็นลำดับและจำนวนของนิ้วมือ ซึ่งอัลกอริทึมการหาลำดับและจำนวนของนิ้วมือแสดงดังรูปที่ 3.11

Algorithm: Search of Order Fingers

1. found duplicate = true, number of finger = 5, finger number $i = 1$
 2. **While** found duplicate
 3. **Set** temp finger number = 1, minimum code = 44
 4. **While** temp finger number \leq number of finger
 5. number duplicate finger array = find code of finger duplicate
 6. **If** minimum code $>$ number duplicate finger array **Then**
 7. minimum code = number duplicate finger array
 8. finger number $i =$ type of finger
 9. **End if**
 10. **Next** temp finger number
 11. **End while**
 12. **If** found duplicate code of finger array **Then**
 13. Remove code of finger in temp finger number
 14. number of finger = number of finger - 1
 15. **Next** i
 16. **Else**
 17. found duplicate = false
 18. **End if else**
 19. **End while**
 20. **Return** finger number
-

รูปที่ 3.11 อัลกอริทึมการหาลำดับและจำนวนของนิ้วมือ

จากรูปที่ 3.11 เป็นขั้นตอนการหาลำดับและจำนวนของนิ้วมือ เริ่มต้นค้นหาลำดับของนิ้วแรก โดยเลือกเก็บรหัสตัวอักษรที่ซ้ำกันของแต่ละนิ้ว (find code of finger duplicate) ตัวอย่างเช่น นิ้วโป้ง ตัวอักษรที่ 1 มีรหัสเป็น 244 ตัวอักษรที่ 2 มีรหัสเป็น 123 ตัวอักษรที่ 3 มีรหัสเป็น 244 ดังนั้นการเก็บรหัสตัวอักษรของนิ้วโป้งจึงเป็น 244 เป็นต้น เมื่อค้นหาครบทุกตัวอักษรแล้วทำการตรวจสอบว่านิ้วมือก่อนหน้ามีขนาดของการจัดเก็บรหัสตัวอักษร (minimum code) มากกว่านิ้วมือ

ปัจจุบัน (size finger array) หรือไม่ ถ้ามากกว่าให้จัดเก็บขนาดของการจัดเก็บรหัสตัวอักษร และ กำหนดให้นิ้วมือปัจจุบันเป็นลำดับของนิ้วแรก จากนั้นตรวจสอบรหัสของนิ้วมือที่จัดเก็บมีรหัสซ้ำหรือไม่ ถ้าซ้ำให้เริ่มค้นหานิ้วถัดไปคือนิ้วที่สอง การค้นหานิ้วที่สอง (find code of finger duplicate) จะนำเฉพาะรหัสนิ้วมือนิ้วแรกที่ซ้ำที่จัดเก็บ (finger array) มาพิจารณาเท่านั้น และจะไม่นำผลลัพธ์ของนิ้วแรกที่ค้นหาก่อนหน้ามาพิจารณา ตัวอย่างเช่น ผลลัพธ์ของนิ้วแรกคือนิ้วกลาง ในการพิจารณา รอบที่สอง จึงเป็นรหัสนิ้วโป้ง รหัสนิ้วชี้ รหัสนิ้วนาง และรหัสนิ้วก้อย เป็นต้น ทำจนกระทั่งไม่มีรหัสนิ้วมือที่ซ้ำกันหรือครบ 5 นิ้ว ตัวอย่างผลลัพธ์เช่น จำนวนของนิ้วมือเท่ากับ 3 นิ้ว ได้แก่ นิ้วแรกคือนิ้วกลาง นิ้วที่ 2 คือนิ้วโป้ง และนิ้วที่ 3 คือนิ้วชี้ เป็นต้น

3.7 อัลกอริทึมการสร้างต้นแบบออโตมาตาแบบจำกัด

อัลกอริทึมการสร้างต้นแบบออโตมาตาแบบจำกัด มีข้อมูลนำเข้า คือ ต้นแบบรหัสการเข้ารหัสรูปแบบ THSL จำนวน 42 ตัวอักษร ผลลัพธ์ที่ได้ คือ โครงสร้างออโตมาตาแบบจำกัดรหัสภาษาไทย ซึ่งอัลกอริทึมการสร้างต้นแบบออโตมาตาแบบจำกัดแสดงดังรูปที่ 3.12

จากนิยามที่ (2.1) องค์ประกอบของออโตมาตาแบบจำกัด กำหนดให้ Σ เป็นตัวเลขรหัสภาษาไทยแต่ละหลักในแต่ละนิ้วมือ และ F เป็นเซตของตัวอักษร 1 ตัวอักษร ดังนั้นอักขระนำเข้าเป็นรหัสภาษามือ อ่านค่าแต่ละหลัก โดยเริ่มอ่านตามผลลัพธ์การค้นลำดับและจำนวนของนิ้วมือ ซึ่งแต่ละนิ้วมีรหัสจำนวน 3 หลัก

Algorithm: Create Finite Automata

1. alphabet = 1, number of alphabet = 42
 2. **While** alphabet <= number of alphabet
 3. finger number = 1, current state = initial state
 4. **While** finger number <= size finger number
 5. **Read** code of finger number 3 integer
 6. **For** i = 1 to 3
 7. **Get** code of finger number integer i
 8. Find state code finger at current state
 9. **If** not found state code finger **Then**
 10. Create state finite automata
 11. **End if**
 12. **Next** i
 13. **End for**
 14. **Next** finger number
 15. **End while**
 16. state automata = final state
 17. **Next** alphabet
 18. **End while**
 19. **Return** finite automata
-

รูปที่ 3.12 อัลกอริทึมการสร้างออโตมาตาจำกัด

จากรูปที่ 3.12 เป็นอัลกอริทึมการสร้างออโตมาตาแบบจำกัด เริ่มต้นกำหนดสถานะปัจจุบัน ซึ่งไปยังสถานะเริ่มต้น รับค่ารหัสตัวอักษร 1 ตัว จากนั้นอ่านค่ารหัสนิ้วที่ 1 จำนวน 3 หลัก แล้วทำการอ่านค่ารหัสหลักที่ 1 ค้นหาสถานะปัจจุบันมีการสร้างทิศทางเคลื่อนไปยังเป็นอีกสถานะเป็นรหัสหลักที่ 1 แล้วหรือไม่ ถ้าไม่พบให้สร้างสถานะที่มีทิศทางเคลื่อนไปยังเป็นอีกสถานะเป็นรหัสหลักที่ 1 และให้สถานะปัจจุบันซึ่งไปยังสถานะที่สร้างขึ้น ทำจนกระทั่งครบทั้ง 3 หลัก เมื่อสร้างสถานะครบทั้ง 3 หลักแล้วให้อ่านค่ารหัสนิ้วถัดไป ทำเช่นเดียวกับขั้นตอนการสร้างออโตมาตาของรหัสนิ้วที่ 1 จนกระทั่งครบตามลำดับ และจำนวนนิ้วมือจากขั้นตอนการหาลำดับและจำนวนของนิ้วมือข้างต้น จึงได้โครงสร้างออโตมาตาแบบจำกัด 1 ตัวอักษร จากนั้นให้ทำสร้างตัวอักษรถัดไปจนครบ 42 ตัวอักษร

3.8 อัลกอริทึมการค้นหาออโตมาตาแบบจำกัดตัวอักษรภาษาไทย

อัลกอริทึมการค้นหาออโตมาตาแบบจำกัดตัวอักษรภาษาไทย มีข้อมูลนำเข้าเป็นรหัสแต่ละนิ้วมือของผู้ทดสอบ และต้นแบบรหัสการเข้ารหัสรูปแบบ THSL ผลลัพธ์ที่ได้ คือ ตัวอักษรภาษาไทย ซึ่งอัลกอริทึมการค้นหาออโตมาตาแบบจำกัดตัวอักษรภาษาไทยแสดงดังรูปที่ 3.13

Algorithm: Search Finite Automata

1. current state = first state of THSL finite automata
 2. **While** EOF of input tape
 3. found = false
 4. **Read** value at current pointer of input tape
 5. **While** consider all transition or not found
 6. **If** value on transition == value of input tape
 7. **Then**
 8. Next state of current transition
 9. found = true
 10. **Else**
 11. Next transition of current state
 12. **End if**
 13. **Next i**
 14. **End if else**
 15. **End while**
 16. Next pointer of input tape
 17. **End while**
 18. **If** state of current state == final state **Then**
 19. Get alphabet of current state
 20. **Return** alphabet
-

รูปที่ 3.13 อัลกอริทึมการค้นหาออโตมาตาแบบจำกัดตัวอักษรภาษาไทย

เริ่มต้นการทำงานกำหนดให้สถานะปัจจุบัน (current state) ซึ่งไปยังสถานะเริ่มต้น จากนั้นอ่านค่าจากหัวอ่านที่ซึ่งไปยังแถบสายอักขระ (input tape) พิจารณาทุกทิศทางสถานะถัดไปของสถานะปัจจุบัน (transition) ถ้าหากค่าบนทิศทางสถานะถัดไป (value on transition) เท่ากับค่าที่

อ่านจากแถบสายอักขระให้สถานะปัจจุบันชี้ไปยังสถานะถัดไปของทิศทางนั้น จากนั้นเลื่อนหัวอ่านของแถบสายอักขระไปทางด้านขวา และพิจารณาทิศทางสถานะถัดไปของสถานะปัจจุบันเช่นเดียวกัน ทำจนกระทั่งสิ้นสุดหัวอ่านของแถบสายอักขระ (EOF of input tape) เมื่อสิ้นสุดหัวอ่านของแถบสายอักขระแล้วให้พิจารณาผลลัพธ์ของอัลกอริทึมนี้จากสถานะการยอมรับของสถานะปัจจุบัน ถ้าหากเป็นสถานะการยอมรับให้ดึงตัวอักษรภาษาไทยของสถานะปัจจุบันออกมา ถ้าหากไม่เป็นสถานะการยอมรับแสดงว่าไม่พบตัวอักษรภาษาไทยใดๆ แสดงดังรูปที่ 3.13

บทที่ 4

ผลการวิจัยและการอภิปรายผล

ในบทนี้เป็นการอธิบายผลการจัดเก็บข้อมูลท่าทางภาษามือในรูปแบบตัวอักษรภาษาไทย 42 ตัวอักษร อุปกรณ์สปีทโมชันเป็นเซ็นเซอร์ในการตรวจจับตำแหน่งนิ้วมือจากผู้ที่มีลักษณะความบกพร่องทางการได้ยินหรือสื่อความหมายสามารถและบุคคลทั่วไป โดยจัดเก็บข้อมูลจากผู้ทดสอบบุคคลทั่วไป จำนวน 30 คน ซึ่งผลการทดสอบในแต่ละขั้นตอนมีรายละเอียดดังต่อไปนี้

4.1 ผลการวิเคราะห์ท่าทางภาษามือตัวอักษรภาษาไทย

พยัญชนะตัวอักษรในภาษาไทยมีจำนวน 44 ตัวอักษร ส่วนท่าทางภาษามือตัวอักษรภาษาไทย มีจำนวน 42 ท่าทาง ตามที่ได้กล่าวในหัวข้อที่ 2.1 ผู้วิจัยได้วิเคราะห์แต่ละท่าทางภาษามือตัวอักษรภาษาไทยพบว่ามิต่างที่แตกต่างจำนวน 25 ท่าทาง แสดงดังตารางที่ 4.1

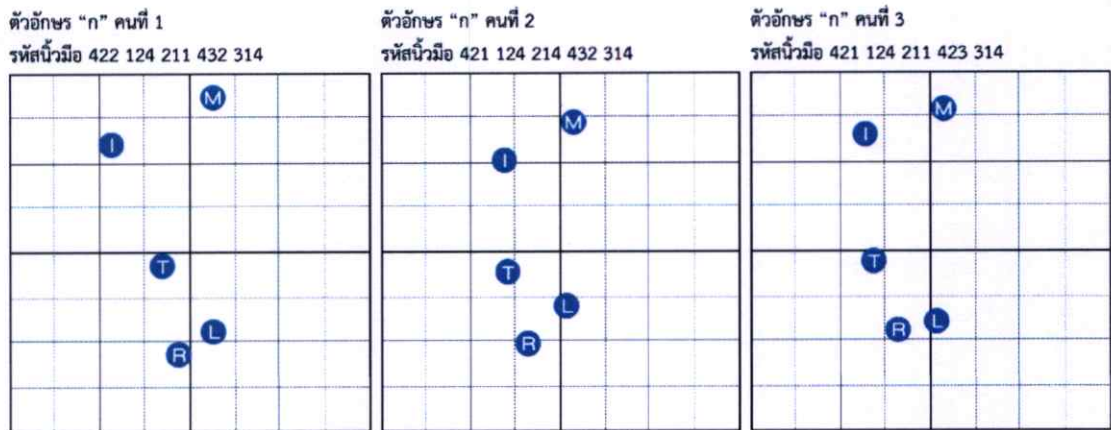
ตารางที่ 4.1 ความแตกต่างของท่าทางภาษามือตัวอักษรภาษาไทยจำนวน 25 ท่าทาง

ลำดับที่	ตัวอักษร	ท่าทาง	ลำดับที่	ตัวอักษร	ท่าทาง	ลำดับที่	ตัวอักษร	ท่าทาง
1	ก		11	ล		21	ฉ1	
2	ค		12	ย		22	1	
3	ส		13	ม		23	2	
4	พ		14	น		24	3	
5	ท		15	อ		25	5	
6	บ		16	ช2				
7	ร		17	จ1				
8	ว		18	จ2				
9	ด		19	ง2				
10	ฟ		20	ท2				

จากตารางที่ 4.1 ความแตกต่างของท่าทางภาษามือตัวอักษรภาษาไทยจำนวน 25 ท่าทาง ประกอบด้วย ท่าทางภาษามือตัวอักษรภาษาไทย 1 จังหวะ จำนวน 15 ท่าทาง ได้แก่ ตัวอักษร “ก” “ต” “ส” “พ” “ห” “บ” “ร” “ว” “ด” “ฟ” “ล” “ย” “ม” “น” และตัวอักษร “อ” ท่าทางภาษามือตัวอักษรภาษาไทย 2 จังหวะ ได้แก่ ตัวอักษร “ซ” ในจังหวะที่ 2 ตัวอักษร “จ” ในจังหวะที่ 1 และจังหวะที่ 2 ตัวอักษร “ง” ในจังหวะที่ 2 ตัวอักษร “ท” ในจังหวะที่ 2 และตัวอักษร “ฉ” ในจังหวะที่ 1 และตัวเลขจำนวน 4 ท่าทาง ได้แก่ ตัวเลข “1” “2” “3” และตัวเลข “5”

4.2 ผลการเข้ารหัสภาษามือตัวอักษรภาษาไทย

ผลการเข้ารหัสภาษามือตัวอักษรภาษาไทยนั้นเกิดจากการจัดเก็บข้อมูลตำแหน่งแกน X และ Y ของปลายนิ้วมือทั้ง 5 นิ้ว และตำแหน่งแกน X และ Y ที่กลางฝ่ามือตามหัวข้อที่ 3.3 จากนั้นทำการปรับค่าข้อมูลให้อยู่ในรูปแบบที่เหมาะสมดังหัวข้อที่ 3.4 และขั้นตอนที่สาม เป็นการเข้ารหัสภาษามือตัวอักษรภาษาไทยโดยใช้เทคนิคควอดทรีในหัวข้อที่ 3.5 ซึ่งแสดงรหัสและตำแหน่งนิ้วมือ 5 นิ้วในแต่ละตัวอักษรดังรูปที่ 4.1 ถึงรูปที่ 4.25 ตามลำดับ

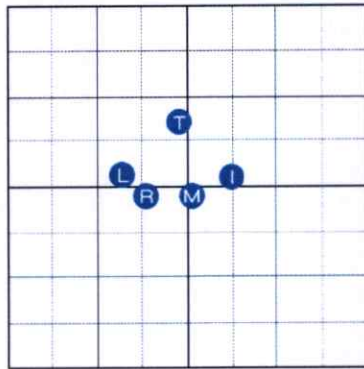


รูปที่ 4.1 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร “ก”

จากรูปที่ 4.1 จะเห็นว่าโครงสร้างตำแหน่งการสะกดนิ้วมือมีความคล้ายคลึงกัน โดยรหัสนิ้วโป่ง คนที่ 2 และ 3 มีรหัสเดียวกัน ส่วนนิ้วชี้ นิ้วนาง และนิ้วก้อยทั้ง 3 คน มีรหัสเดียวกัน นิ้วกลางคนที่ 1 และ 3 มีรหัสเดียวกัน ตำแหน่งของนิ้วโป่งอยู่บริเวณกลางตารางไปทางด้านซ้าย ตำแหน่งของนิ้วชี้ และนิ้วกลางจะอยู่บริเวณด้านบนบนของตาราง ส่วนนิ้วนาง และนิ้วก้อยจะอยู่บริเวณด้านล่างของตาราง

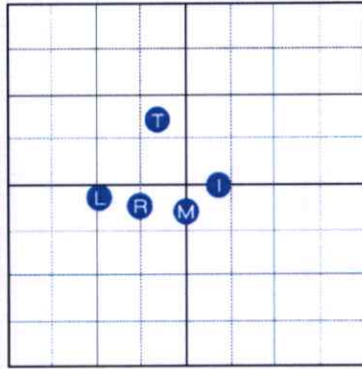
ตัวอักษร "ต" คนที่ 1

รหัสนิ้วมือ 132 243 311 422 134



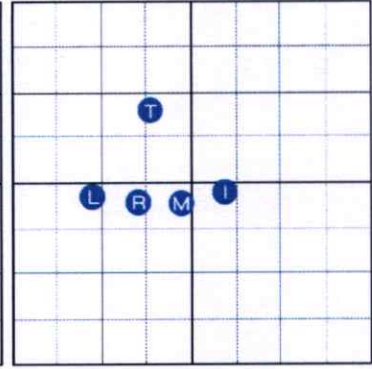
ตัวอักษร "ต" คนที่ 2

รหัสนิ้วมือ 132 331 311 422 421



ตัวอักษร "ต" คนที่ 3

รหัสนิ้วมือ 132 311 422 421 412

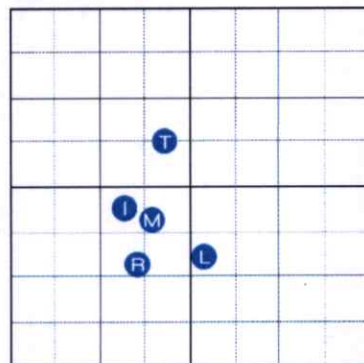


รูปที่ 4.2 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "ต"

จากรูปที่ 4.2 แสดงโครงสร้างตำแหน่งการสะกดนิ้วมือมีความคล้ายคลึงกัน โดยรหัสนิ้วโป้งทั้ง 3 คน มีรหัสเดียวกัน ส่วนนิ้วชี้ทั้ง 3 คน มีรหัสที่แตกต่างกัน นิ้วกลางคนที่ 1 และ 2 มีรหัสเดียวกัน ตำแหน่งของนิ้วโป้งอยู่บริเวณกึ่งกลางด้านบนของตาราง ตำแหน่งของนิ้วชี้ นิ้วกลาง นิ้วนาง และนิ้วก้อย จะอยู่บริเวณด้านล่างของตาราง โดยแต่ละนิ้วจะกระจายกันออกไปตามลำดับของนิ้วมือ

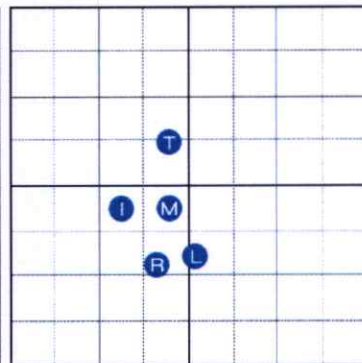
ตัวอักษร "ส" คนที่ 1

รหัสนิ้วมือ 134 421 422 424 314



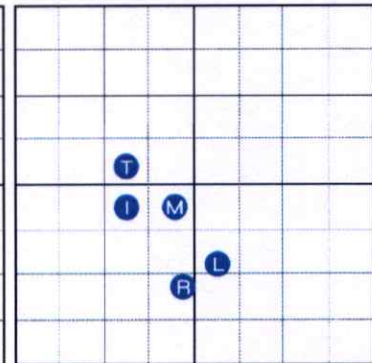
ตัวอักษร "ส" คนที่ 2

รหัสนิ้วมือ 133 421 422 423 314



ตัวอักษร "ส" คนที่ 3

รหัสนิ้วมือ 134 421 422 311 423



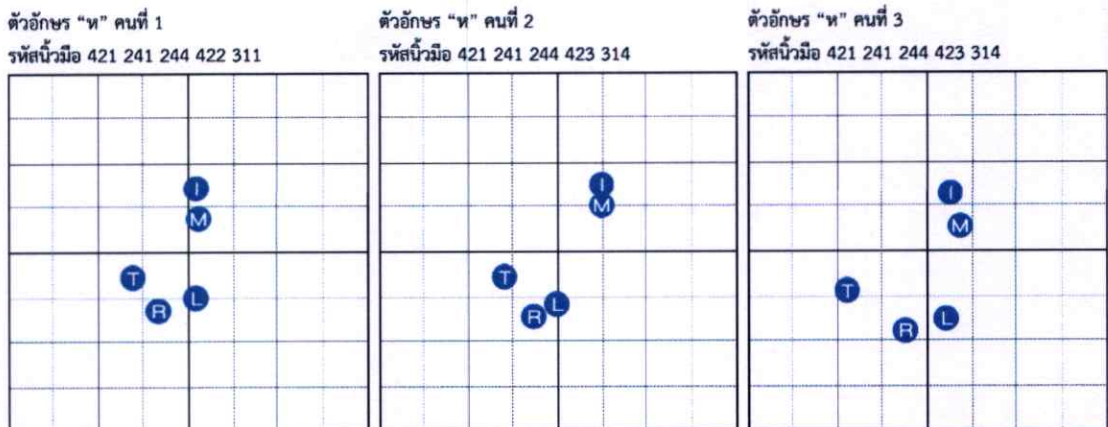
รูปที่ 4.3 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "ส"

จากรูปที่ 4.3 แสดงรหัสนิ้วโป้งคนที่ 1 และคนที่ 3 มีรหัสเดียวกัน ส่วนนิ้วชี้ และนิ้วกลางทั้ง 3 คน มีรหัสที่เหมือนกัน นิ้วนางของแต่ละคนมีรหัสที่ต่างกันไป นิ้วก้อยของคนี่ 1 และคนที่ 2 มีรหัสเดียวกัน ตำแหน่งแต่ละนิ้วมือจะกระจุกตัวบริเวณกึ่งกลางของตาราง โดยตำแหน่งนิ้วโป้งจะอยู่ด้านบนของตาราง ส่วนตำแหน่งอื่นๆ ของนิ้วมือจะอยู่บริเวณด้านล่างของตารางเสมอ



รูปที่ 4.4 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "พ"

จากรูปที่ 4.4 แสดงรหัสนิ้วโป้ง นิ้วชี้ นิ้วกลาง และนิ้วนางของคน ที่ 1 และคนที่ 2 มีรหัสเดียวกัน ส่วนนิ้วโป้ง นิ้วชี้ นิ้วกลาง และนิ้วนางคนที่ 3 จะมียุทธศาสตร์แตกต่างกับคนที่ 1 และคนที่ 2 เพียงหลักที่ 3 เท่านั้น ส่วนนิ้วก้อยทั้ง 3 คน มีรหัสเหมือนกัน ตำแหน่งนิ้วชี้จะอยู่บริเวณด้านบนบนซ้าย ตำแหน่งนิ้วโป้ง นิ้วกลาง นิ้วนาง และนิ้วก้อยจะกระจุกตัวบริเวณกึ่งกลางด้านล่างของตาราง



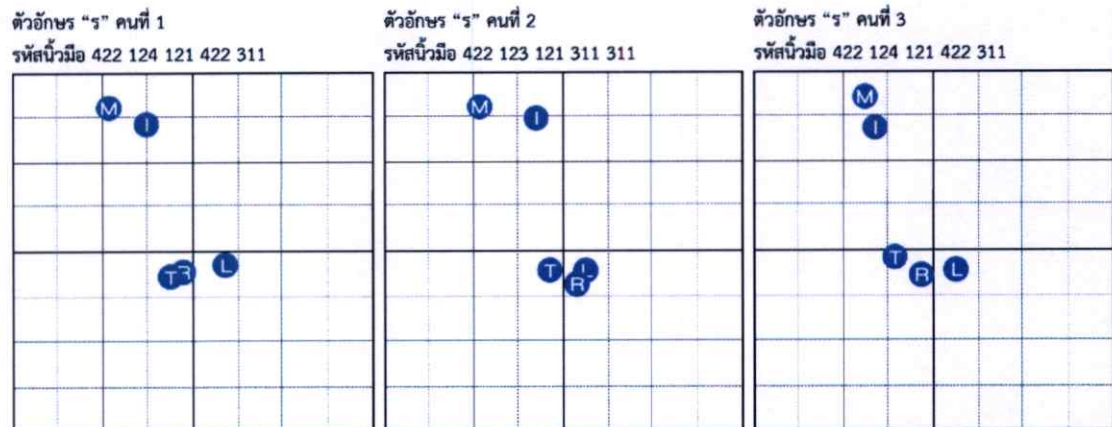
รูปที่ 4.5 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "ห"

จากรูปที่ 4.5 แสดงรหัสนิ้วโป้ง นิ้วชี้ นิ้วกลางทั้ง 3 คน มีรหัสเดียวกัน ส่วนนิ้วนางคนที่ 1 มีรหัสหลักที่ 3 แตกต่างจากคนที่ 1 และคนที่ 2 ส่วนนิ้วก้อยคนที่ 2 และคนที่ 3 มีรหัสเดียวกัน ตำแหน่งนิ้วชี้และนิ้วกลางจะกระจุกตัวอยู่บริเวณด้านบนบนขวา ตำแหน่งนิ้วโป้ง นิ้วนาง และนิ้วก้อยจะกระจุกตัวบริเวณด้านล่างซ้ายของตาราง



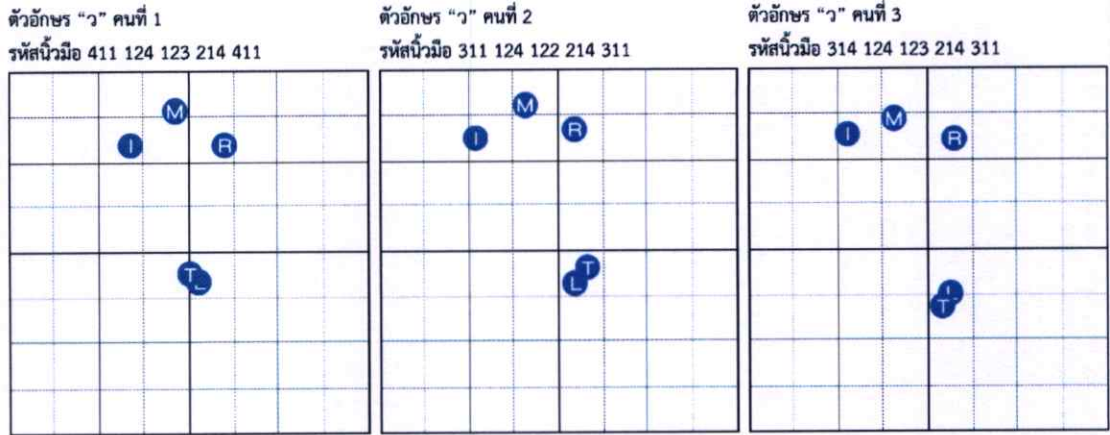
รูปที่ 4.6 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "บ"

จากรูปที่ 4.6 แสดงรหัสนิ้วโป่งคนที่ 1 และคนที่ 3 มีรหัสเดียวกัน นิ้วชี้ของทั้ง 3 มีความแตกต่างกันไปในหลักที่ 3 นิ้วกลางทั้ง 3 คน มีรหัสที่เหมือนกัน คือ "122" ส่วนนิ้วก้อย และนิ้วนางของทั้ง 3 คนไม่เหมือนกันในหลักที่ 2 และหลักที่ 3 ตำแหน่งนิ้วโป่งจะอยู่บริเวณด้านล่างซ้ายของตาราง ส่วนตำแหน่งนิ้วชี้ นิ้วกลาง นิ้วนาง และนิ้วก้อยจะแผ่กระจายตัวอยู่บริเวณด้านบน



รูปที่ 4.7 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "ร"

จากรูปที่ 4.7 พบว่ารหัสนิ้วโป่ง นิ้วกลาง และนิ้วก้อยทั้ง 3 คน มีรหัสเดียวกัน ส่วนนิ้วชี้มีเพียงคนที่ 2 ที่รหัสในหลักที่ 3 แตกต่างกัน พิจารณานิ้วนางคนที่ 1 และคนที่ 3 มีรหัสเดียวกัน ตำแหน่งนิ้วโป่ง นิ้วนาง และนิ้วก้อยจะอยู่บริเวณกึ่งกลางด้านล่างของตาราง ส่วนตำแหน่งนิ้วชี้และนิ้วกลางจะอยู่บริเวณด้านบนซ้าย



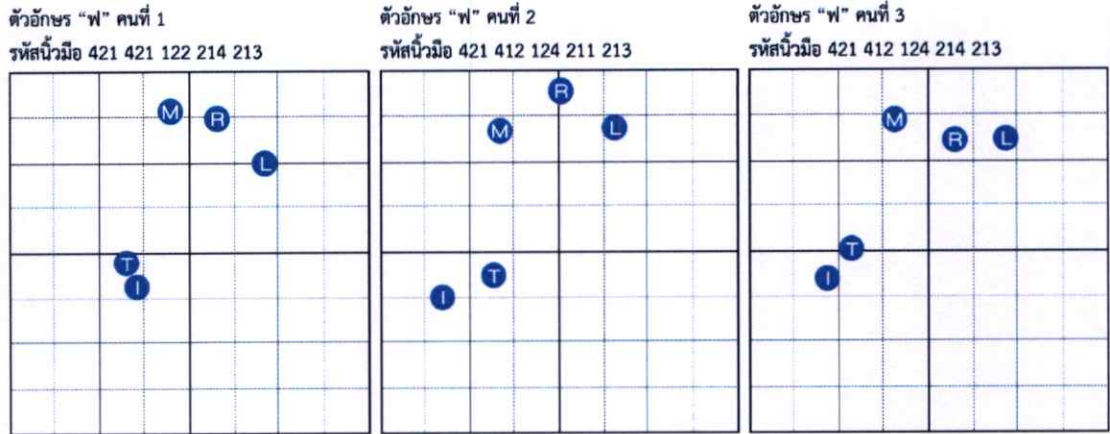
รูปที่ 4.8 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "ว"

จากรูปที่ 4.8 พบว่ารหัสนิ้วโป้งแต่ละคนมีรหัสที่แตกต่างกัน ส่วนนิ้วกลางและนิ้วนางทั้ง 3 นิ้ว มีรหัสเดียวกัน นิ้วกลางคนที่ 1 และคนที่ 3 คน มีรหัสเดียวกัน นิ้วก้อยมีเพียงคนที่ 1 ที่รหัสในหลักที่ 1 แตกต่างกัน ตำแหน่งนิ้วโป้ง และนิ้วก้อยจะอยู่บริเวณกึ่งกลางด้านล่างของตาราง ส่วนตำแหน่งนิ้วชี้ นิ้วกลาง และนิ้วนางจะอยู่บริเวณด้านบน



รูปที่ 4.9 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "ด"

จากรูปที่ 4.9 จะเห็นว่ารหัสนิ้วโป้งคนที่ 1 และคนที่ 3 คน มีรหัสเดียวกัน นิ้วชี้ นิ้วกลาง นิ้วนาง และนิ้วก้อยของแต่ละคนมีรหัสที่เหมือนกัน ตำแหน่งนิ้วโป้ง นิ้วกลาง นิ้วนาง และนิ้วก้อยจะอยู่บริเวณกึ่งกลางด้านล่างของตาราง ส่วนตำแหน่งนิ้วชี้จะอยู่บริเวณด้านบนซ้ายของตาราง



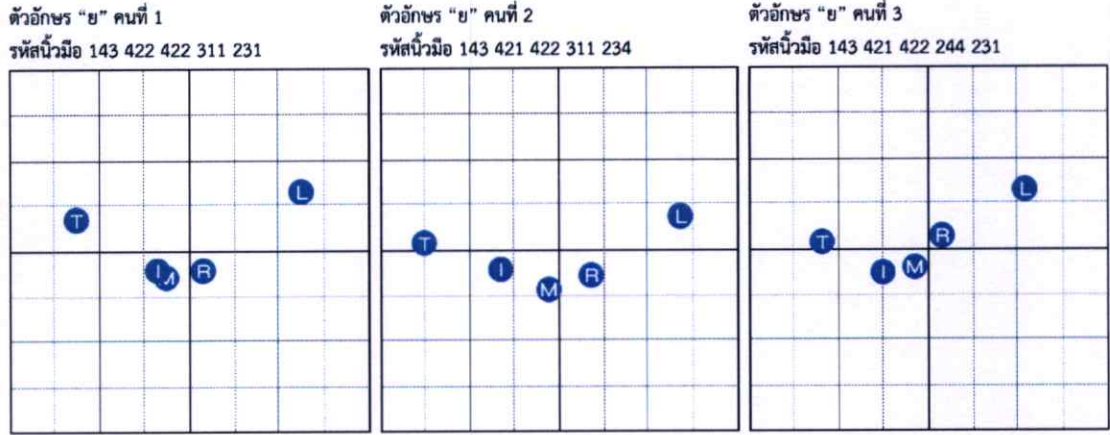
รูปที่ 4.10 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "ฟ"

จากรูปที่ 4.10 จะเห็นว่ารหัสนิ้วโป้งและนิ้วก้อยทั้ง 3 คนมีรหัสเดียวกัน นิ้วชี้และนิ้วกลางคนที่ 2 และคนที่ 3 มีรหัสเดียวกันคือ "412" "124" ตามลำดับ นิ้วนางคนที่ 1 และคนที่ 3 มีรหัสที่เหมือนกัน ตำแหน่งนิ้วโป้ง และนิ้วชี้จะอยู่บริเวณกึ่งกลางด้านล่างของตาราง ส่วนตำแหน่งนิ้วกลาง นิ้วนาง และนิ้วก้อยจะอยู่บริเวณด้านบนบนของตาราง



รูปที่ 4.11 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "ล"

จากรูปที่ 4.11 แสดงรหัสนิ้วโป้งคนที่ 1 และคนที่ 3 มีรหัสเดียวกัน นิ้วชี้ของทั้ง 3 มีความแตกต่างกันไปในหลักที่ 3 นิ้วกลางทั้ง 3 คน มีรหัสที่เหมือนกัน คือ "122" ส่วนนิ้วก้อย และนิ้วนางของทั้งสามคนไม่เหมือนกันในหลักที่ 2 และหลักที่ 3 ตำแหน่งนิ้วโป้งจะอยู่บริเวณด้านล่างซ้ายของตาราง ส่วนตำแหน่งนิ้วชี้ นิ้วกลาง นิ้วนาง และนิ้วก้อยจะแผ่กระจายตัวอยู่บริเวณด้านบน



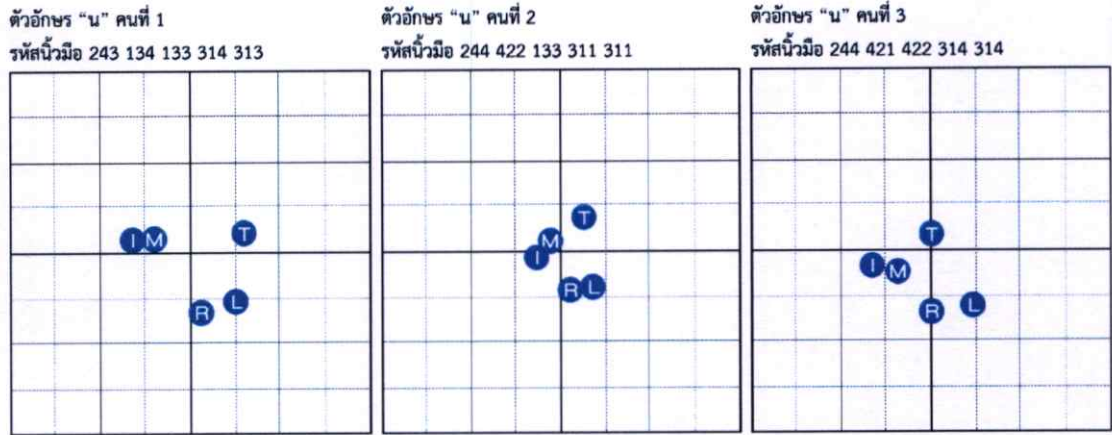
รูปที่ 4.12 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "ย"

จากรูปที่ 4.12 จะเห็นได้ว่ารหัสนิ้วโป้งและนิ้วกลางทั้ง 3 มีรหัสเดียวกัน นิ้วชี้คนที่ 2 และคนที่ 3 มีรหัสเดียวกัน สำหรับนิ้วชี้คนที่ 1 มีเพียงรหัสหลักที่ 3 เท่านั้นที่แตกต่างกับนิ้วชี้คนที่ 2 และคนที่ 3 นิ้วนางคนที่ 1 และคนที่ 2 มีรหัสเดียวกัน นิ้วก้อยมีเพียงคนที่ 1 และคนที่ 3 ที่มีรหัสเดียวกัน ตำแหน่งนิ้วมือทั้ง 5 จะกระจายตัวอยู่บริเวณกึ่งกลางแนวแกน X ของตาราง โดยนิ้วชี้จะอยู่ห่างจากกลุ่มนิ้วมืออื่นๆ ไปทางด้านขวาค่อนข้างมาก



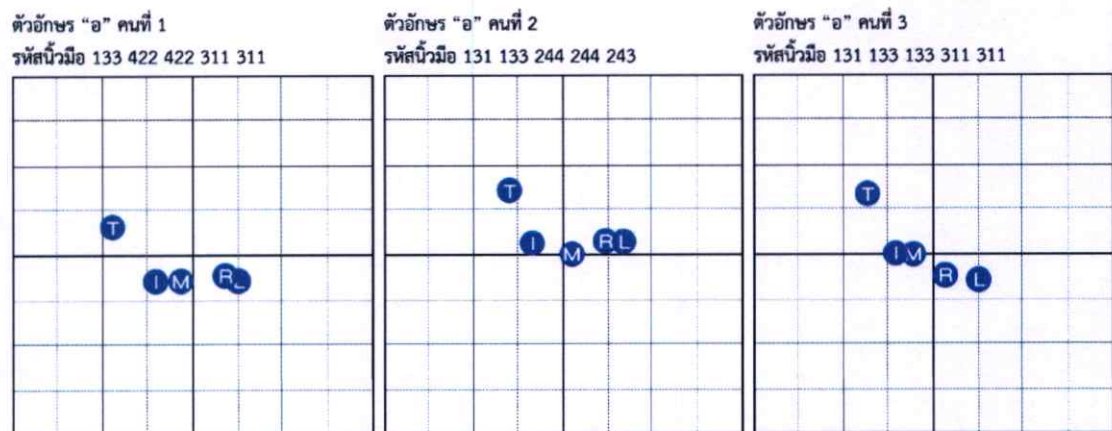
รูปที่ 4.13 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "ม"

จากรูปที่ 4.13 แสดงรหัสนิ้วโป้ง นิ้วกลาง และนิ้วนางคนที่ 1 และคนที่ 3 มีรหัสเดียวกัน นิ้วชี้ของทั้ง 3 มีความแตกต่างกันไปในหลักที่ 3 เท่านั้น ส่วนนิ้วก้อยของทั้ง 3 คนไม่เหมือนกันในหลักที่ 2 และหลักที่ 3 ตำแหน่งนิ้วมือทั้ง 5 นิ้วจะอยู่บริเวณกึ่งกลางของตาราง



รูปที่ 4.14 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "น"

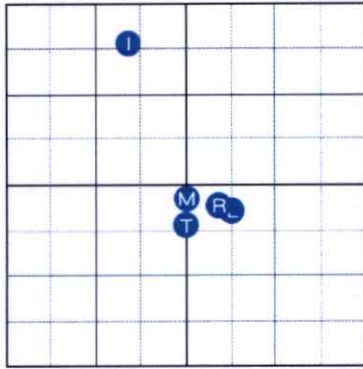
จากรูปที่ 4.14 แสดงรหัสนิ้วโป่งคนที่ 2 และคนที่ 3 มีรหัสเดียวกัน นิ้วชี้ของทั้ง 3 มีความแตกต่างกัน ส่วนนิ้วกลางคนที่ 1 และคนที่ 2 มีรหัสเดียวกัน นิ้วนางคนที่ 1 และคนที่ 3 มีรหัสเดียวกัน และนิ้วก้อยของทั้ง 3 คน มีรหัสที่ต่างกันในรหัสหลักที่ 3 ตำแหน่งนิ้วมือทั้ง 5 นิ้วจะอยู่บริเวณกึ่งกลางของตาราง พิจารณาตำแหน่งแต่ละนิ้วมือของตัวอักษร "ม" กับตัวอักษร "น" มีลักษณะคล้ายคลึงกันเป็นอย่างมาก



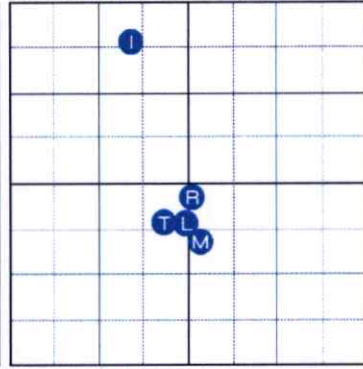
รูปที่ 4.15 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "อ"

จากรูปที่ 4.15 แสดงรหัสนิ้วโป่งและนิ้วชี้คนที่ 2 และคนที่ 3 มีรหัสเดียวกัน ส่วนนิ้วกลางของทั้ง 3 มีความแตกต่างกัน พิจารณานิ้วนางและนิ้วก้อยคนที่ 1 และคนที่ 3 มีรหัสเดียวกัน ตำแหน่งนิ้วชี้ นิ้วกลาง นิ้วนาง และนิ้วก้อยจะกระจุกตัวอยู่บริเวณกึ่งกลางของตาราง ส่วนตำแหน่งนิ้วโป่งจะอยู่บริเวณด้านบนซ้าย ห่างออกไปเล็กน้อยจากกลุ่มนิ้วมืออื่นๆ

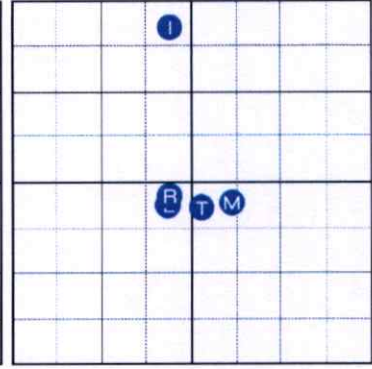
ตัวอักษร "ซ" จังหวะที่ 2 คนที่ 1
รหัสนิ้วมือ 311 121 311 311 311



ตัวอักษร "ซ" จังหวะที่ 2 คนที่ 2
รหัสนิ้วมือ 422 121 314 311 422



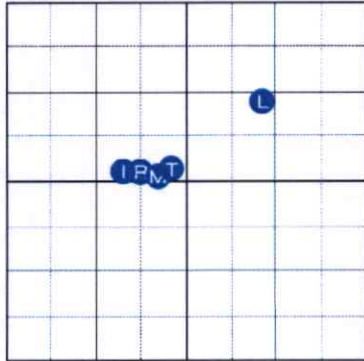
ตัวอักษร "ซ" จังหวะที่ 2 คนที่ 3
รหัสนิ้วมือ 311 122 311 422 422



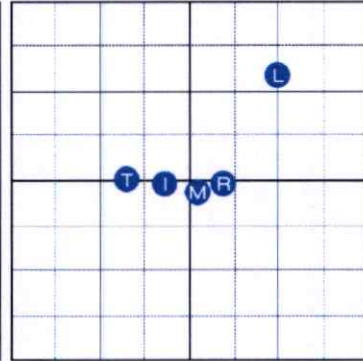
รูปที่ 4.16 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "ซ" จังหวะที่ 2

จากรูปที่ 4.16 แสดงรหัสนิ้วโป้งและนิ้วกลางคนที่ 1 และคนที่ 3 มีรหัสเดียวกัน ส่วนนิ้วกลางและนิ้วนางของทั้ง 3 มีความแตกต่างกัน พิจารณานิ้วชี้และนิ้วนางคนที่ 1 และคนที่ 2 มีรหัสเดียวกัน นิ้วชี้คนที่ 3 มีเพียงรหัสหลักที่ 3 เท่านั้นที่แตกต่างกับคนที่ 1 และคนที่ 2 นิ้วก้อยคนที่ 2 และคนที่ 3 มีรหัสเดียวกัน ตำแหน่งนิ้วโป้ง นิ้วกลาง นิ้วนาง และนิ้วก้อยจะกระจุกตัวอยู่บริเวณกึ่งกลางของตาราง ส่วนตำแหน่งนิ้วชี้จะอยู่บริเวณด้านบนซ้าย

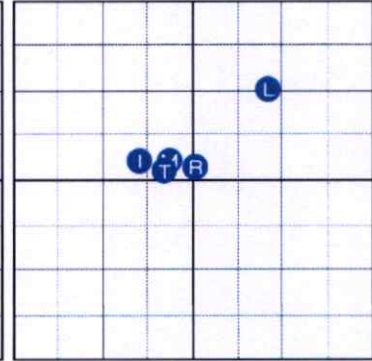
ตัวอักษร "จ" จังหวะที่ 1 คนที่ 1
รหัสนิ้วมือ 133 134 134 134 242



ตัวอักษร "จ" จังหวะที่ 1 คนที่ 2
รหัสนิ้วมือ 134 422 311 311 213

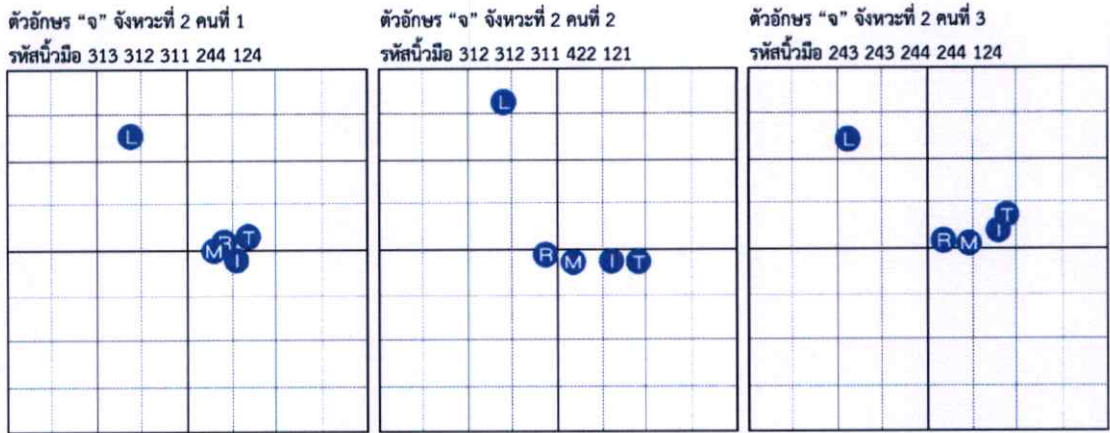


ตัวอักษร "จ" จังหวะที่ 1 คนที่ 3
รหัสนิ้วมือ 133 134 133 244 213



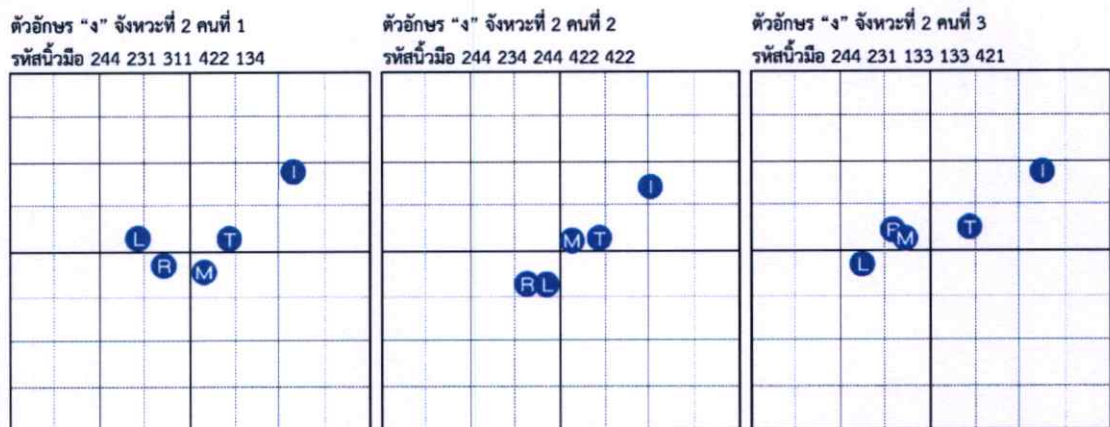
รูปที่ 4.17 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "จ" จังหวะที่ 1

จากรูปที่ 4.17 แสดงรหัสนิ้วโป้งและนิ้วชี้คนที่ 1 และคนที่ 3 มีรหัสเดียวกัน ส่วนนิ้วกลางและนิ้วนางของทั้ง 3 มีความแตกต่างกัน พิจารณานิ้วก้อยคนที่ 2 และคนที่ 3 มีรหัสเดียวกัน ตำแหน่งนิ้วชี้ นิ้วกลาง และนิ้วนางจะกระจุกตัวอยู่บริเวณกึ่งกลางด้านซ้ายของตาราง ส่วนตำแหน่งนิ้วก้อยจะอยู่บริเวณด้านบนขวา



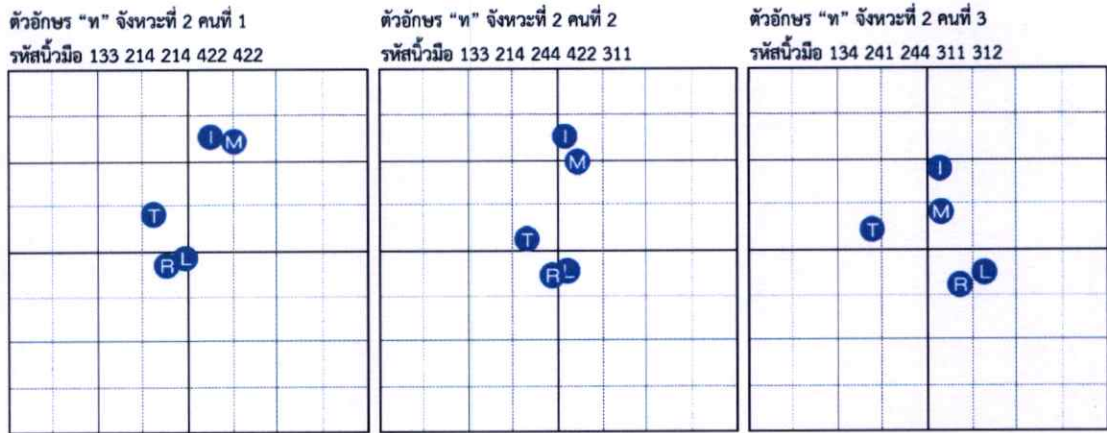
รูปที่ 4.18 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "จ" จังหวะที่ 2

จากรูปที่ 4.18 แสดงรหัสนิ้วโป่งทั้ง 3 คนมีรหัสที่แตกต่างกัน ส่วนนิ้วชี้และนิ้วกลางคนที่ 1 และคนที่ 2 มีรหัสเดียวกัน นิ้วนางและนิ้วก้อยคนที่ 2 และคนที่ 3 มีรหัสเดียวกัน ตำแหน่งนิ้วโป่ง นิ้วชี้ นิ้วกลาง และนิ้วนางจะกระจุกตัวอยู่บริเวณกึ่งกลางด้านขวาของตาราง ส่วนตำแหน่งนิ้วก้อยจะอยู่บริเวณด้านบนซ้ายสลับซ้าย-ขวากับรหัสตัวอักษร "จ" จังหวะที่ 1



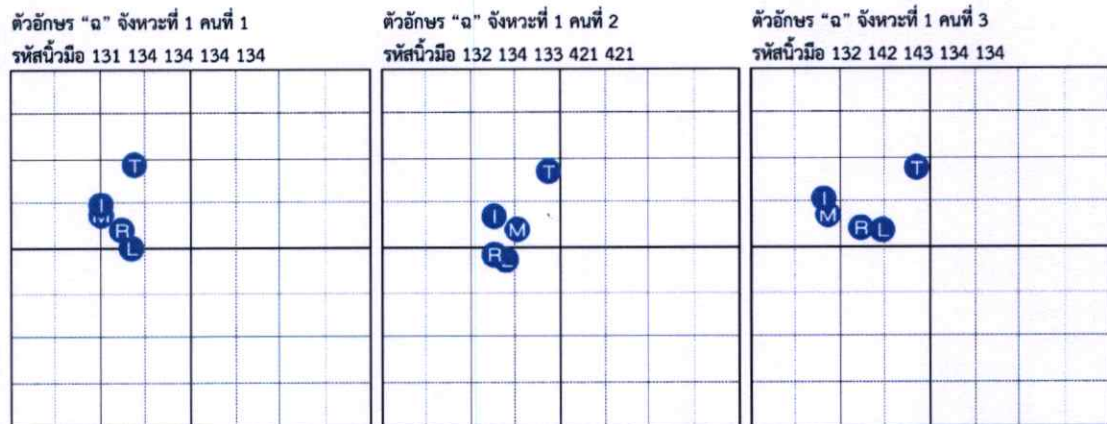
รูปที่ 4.19 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "ง" จังหวะที่ 2

จากรูปที่ 4.19 แสดงรหัสนิ้วโป่งทั้ง 3 คนมีรหัสเดียวกัน ส่วนนิ้วชี้และนิ้วนางคนที่ 1 และคนที่ 3 มีรหัสเดียวกัน นิ้วกลางและนิ้วก้อยของทั้ง 3 มีความแตกต่างกัน ตำแหน่งนิ้วโป่ง นิ้วกลาง นิ้วนาง และนิ้วก้อยจะอยู่บริเวณกึ่งกลางของตาราง ส่วนตำแหน่งนิ้วชี้จะอยู่บริเวณด้านบนขวา



รูปที่ 4.20 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "ท" จังหวะที่ 2

จากรูปที่ 4.20 จะเห็นได้ว่ารหัสนิ้วโป้ง นิ้วชี้ และนิ้วนางคนที่ 1 และคนที่ 2 มีรหัสเดียวกัน นิ้วกลางคนที่ 2 และคนที่ 3 มีรหัสเดียวกันคือ "244" ส่วนนิ้วก้อยทั้ง 3 คน มีรหัสที่แตกต่างกันไป ตำแหน่งนิ้วโป้ง นิ้วนาง และนิ้วก้อยจะอยู่บริเวณกึ่งกลางของตาราง ส่วนนิ้วชี้และนิ้วกลางอยู่บริเวณด้านบนขวา



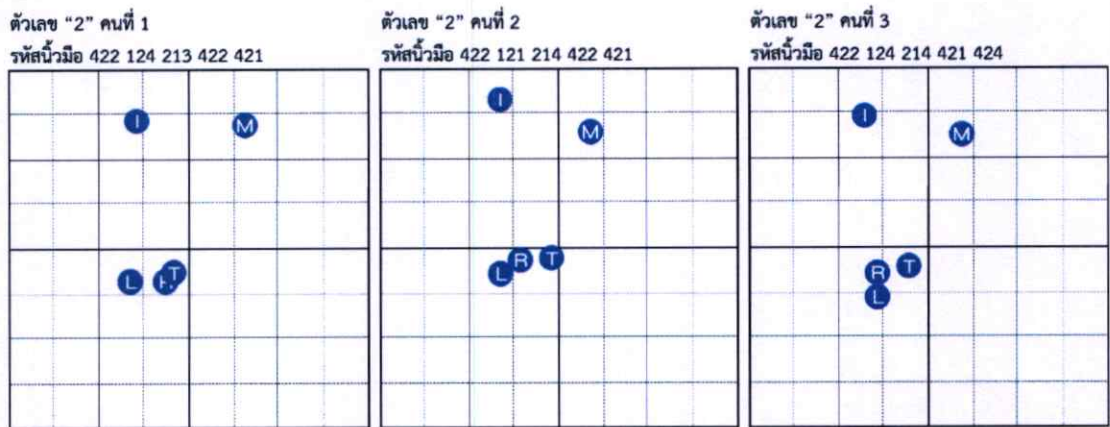
รูปที่ 4.21 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวอักษร "ฉ" จังหวะที่ 1

จากรูปที่ 4.21 จะเห็นได้ว่ารหัสนิ้วโป้งคนที่ 2 และคนที่ 3 มีรหัสเดียวกัน นิ้วชี้คนที่ 1 และคนที่ 2 มีรหัสที่เหมือนกัน นิ้วกลางของทั้ง 3 มีความแตกต่างกัน ส่วนนิ้วกลางทั้ง 3 คน มีรหัสที่เหมือนกันคือ "122" ส่วนนิ้วนาง และนิ้วก้อยคนที่ 1 กับคนที่ 3 มีรหัสเดียวกัน พิจารณาความสัมพันธ์นิ้วนางและนิ้วก้อยมีของทั้ง 3 คนมีรหัสเดียวกัน ตำแหน่งนิ้วมือ 5 นิ้วจะกระจุกตัวอยู่บริเวณด้านบนซ้ายของตาราง มีเพียงนิ้วโป้งที่กระจายตัวออกมาด้านบนขวาเล็กน้อยเท่านั้น



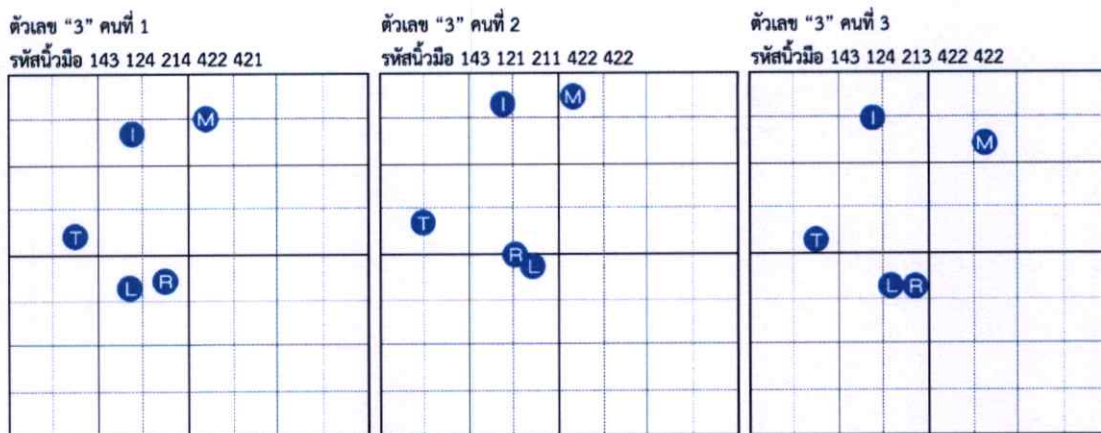
รูปที่ 4.22 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวเลข “1”

จากรูปที่ 4.22 แสดงรหัสนิ้วโป่งคนที่ 2 และคนที่ 3 มีรหัสเดียวกัน นิ้วชี้ของทั้ง 3 มีรหัสที่เหมือนกัน นิ้วกลางคนที่ 1 จากทั้ง 3 คนมีรหัสแตกต่างกันในหลักที่ 3 เท่านั้น ส่วนนิ้วนางคนที่ 2 จากทั้ง 3 คนมีรหัสแตกต่างกันในหลักที่ 3 เท่านั้น ตำแหน่งนิ้วโป่ง นิ้วกลาง นิ้วนาง และนิ้วก้อยจะอยู่บริเวณกึ่งกลางด้านล่างของตาราง ส่วนตำแหน่งนิ้วชี้จะอยู่บริเวณด้านบนบนซ้ายของตาราง



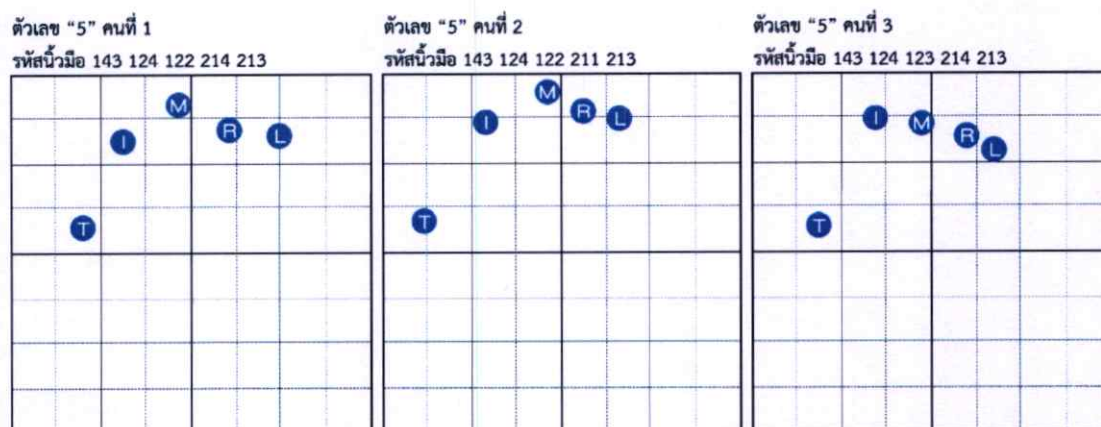
รูปที่ 4.23 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวเลข “2”

จากรูปที่ 4.23 แสดงรหัสนิ้วโป่งของทั้ง 3 คนที่เหมือนกัน ส่วนนิ้วชี้คนที่ 1 และคนที่ 3 มีรหัสเดียวกัน พิจารณานิ้วชี้คนที่ 2 จะแตกต่างกันกับคนที่ 1 และคนที่ 3 ในหลักที่ 3 เท่านั้น ส่วนนิ้วกลางคนที่ 2 และคนที่ 3 มีรหัสเดียวกัน นิ้วกลางคนที่ 1 จะมีรหัสตำแหน่งที่ 3 เท่านั้นที่แตกต่าง เช่นเดียวกับนิ้วนาง และนิ้วก้อยที่มีเพียงคนที่ 3 ที่รหัสในตำแหน่งที่ 3 แตกต่างกัน ตำแหน่งนิ้วโป่ง นิ้วนาง และนิ้วก้อยจะกระจุกตัวอยู่บริเวณด้านล่างซ้ายของตาราง ส่วนตำแหน่งนิ้วชี้ นิ้วกลางจะอยู่บริเวณด้านบน



รูปที่ 4.24 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวเลข "3"

จากรูปที่ 4.24 จะเห็นได้ว่ารหัสนิ้วโป้ง และนิ้วนางของทั้ง 3 คนมีรหัสที่เหมือนกัน ส่วนนิ้วชี้ คนที่ 1 และคนที่ 3 มีรหัสเดียวกัน รหัสนิ้วกลางของทั้ง 3 คนมีความแตกต่างกันไปในหลักที่ 3 ส่วน นิ้วก้อยมีเพียงคนที่ 1 ที่มีรหัสหลักที่ 3 แตกต่างกัน ตำแหน่งนิ้วโป้ง นิ้วชี้ นิ้วนาง และนิ้วก้อยจะอยู่บริเวณด้านซ้ายของตาราง ส่วนตำแหน่งนิ้วกลางจะอยู่บริเวณด้านขวาบน



รูปที่ 4.25 ตำแหน่งแต่ละนิ้วมือ 5 นิ้ว ตัวเลข "5"

จากรูปที่ 4.25 แสดงรหัสนิ้วโป้ง นิ้วชี้ และนิ้วนางทั้ง 3 คนที่เหมือนกัน ส่วนนิ้วกลางคนที่ 1 และคนที่ 2 มีรหัสเดียวกัน นิ้วกลางของคนี 1 และคนที่ 2 มีรหัสเดียวกัน ส่วนนิ้วกลางคนที่ 3 มีเพียงรหัสหลักที่ 3 แตกต่างจากคนที่ 1 และคนที่ 2 พิจารณานิ้วนางของคนี 1 และคนที่ 3 มีรหัสเดียวกัน ส่วนนิ้วนางคนที่ 2 มีเพียงรหัสหลักที่ 3 แตกต่างจากคนที่ 1 และคนที่ 3 ตำแหน่งของนิ้วมือทั้ง 5 จะกระจายอยู่บริเวณด้านบนของตาราง โดยนิ้วโป้งอยู่ด้านล่างซ้ายของนิ้วอื่น ๆ

ผลการเข้ารหัสภาษามือตัวอักษรภาษาไทยแต่ละคนผู้วิจัยได้ทำการเลือกรหัสนิ้วมือที่ตรงกันมากที่สุดของรหัสการสะกดแต่ละนิ้วมือจำนวน 42 ตัวอักษร เพื่อนำมาสร้างออโตมาตาแบบจำกัดในขั้นตอนต่อไป โดยแบ่งประเภทตารางตามจังหวัดการเคลื่อนไหวแสดงดังตารางที่ 4.2 ถึงตารางที่ 4.4

ตารางที่ 4.2 รหัสการเข้ารหัสรูปแบบ THSL การสะกดนิ้วมือ 1 จังหวะ จำนวน 15 ตัวอักษร

ตัวอักษร	นิ้วโป้ง	นิ้วชี้	นิ้วกลาง	นิ้วนาง	นิ้วก้อย
ก	422	124	211	432	314
ด	132	243	311	422	134
ส	133	421	422	423	314
พ	422	131	423	314	314
ห	421	241	244	423	314
บ	421	121	122	214	213
ร	422	123	121	422	311
ว	314	124	123	214	314
ด	422	123	422	311	312
ฟ	421	421	122	211	213
ล	143	123	311	311	312
ย	143	422	422	311	231
ม	243	422	133	311	311
น	244	422	133	311	311
อ	131	422	422	311	311

พิจารณາตารางที่ 4.2 จะเห็นว่าตัวอักษร “ม” และ “น” เท่านั้นที่มีรหัสนิ้วมือ 3 หลักทั้ง 5 นิ้ว ที่เหมือนกัน ส่วนรหัสแต่ละนิ้วมือของตัวอักษรอื่นๆ มีความแตกต่างกันไป

ตารางที่ 4.3 รหัสการเข้ารหัสรูปแบบ THSL การสะกดนิ้วมือ 2 จังหวะ จำนวน 24 ตัวอักษร

ตัวอักษร	จังหวะที่	นิ้วโป้ง	นิ้วชี้	นิ้วกลาง	นิ้วนาง	นิ้วก้อย
ข	1	422	124	211	432	314
	2	422	123	432	432	423
ค	1	422	124	211	432	314
	2	422	123	212	314	421
ฅ	1	422	124	211	432	314
	2	143	124	211	422	422
ถ	1	132	243	311	422	134
	2	422	123	432	432	423

ตารางที่ 4.3 (ต่อ) รหัสการเข้ารหัสรูปแบบ THSL การสะกดนิ้วมือ 2 จังหวะ จำนวน 24 ตัวอักษร

ตัวอักษร	จังหวะที่	นิ้วโป้ง	นิ้วชี้	นิ้วกลาง	นิ้วนาง	นิ้วก้อย
ฐ	1	132	243	311	422	134
	2	311	123	212	314	314
ฒ	1	132	243	311	422	134
	2	143	124	211	311	311
ท	1	132	243	311	422	134
	2	421	121	122	211	213
ฎ	1	132	243	311	422	134
	2	143	123	211	213	231
ศ	1	133	421	422	423	314
	2	422	123	432	432	423
ษ	1	133	421	422	423	314
	2	311	123	212	314	314
ช	1	133	421	422	423	314
	2	133	241	421	422	423
ป	1	422	131	423	314	314
	2	422	123	432	432	423
ผ	1	422	131	423	314	314
	2	422	123	212	314	421
ภ	1	422	131	423	314	314
	2	143	124	211	422	422
ฮ	1	422	124	311	432	314
	2	143	123	122	214	213
ฎ	1	422	123	422	311	312
	2	422	123	432	432	423
ฬ	1	421	421	122	211	213
	2	422	123	432	432	423
พ	1	143	123	311	311	312
	2	422	123	432	432	423
จ	1	422	421	133	133	242
	2	241	243	243	422	142

ตารางที่ 4.3 (ต่อ) รหัสการเข้ารหัสรูปแบบ THSL การสะกดนิ้วมือ 2 จังหวะ จำนวน 24 ตัวอักษร

ตัวอักษร	จังหวะที่	นิ้วโป้ง	นิ้วชี้	นิ้วกลาง	นิ้วนาง	นิ้วก้อย
ญ	1	143	422	422	311	224
	2	422	123	432	432	423
ณ	1	244	422	133	311	311
	2	422	123	432	432	423
ง	1	244	422	133	311	311
	2	133	241	244	422	424
ท	1	132	134	422	422	311
	2	134	241	244	422	422
ฉ	1	131	134	134	134	421
	2	134	241	244	422	422

ตารางที่ 4.4 รหัสการเข้ารหัสรูปแบบ THSL การสะกดนิ้วมือ 3 จังหวะ จำนวน 3 ตัวอักษร

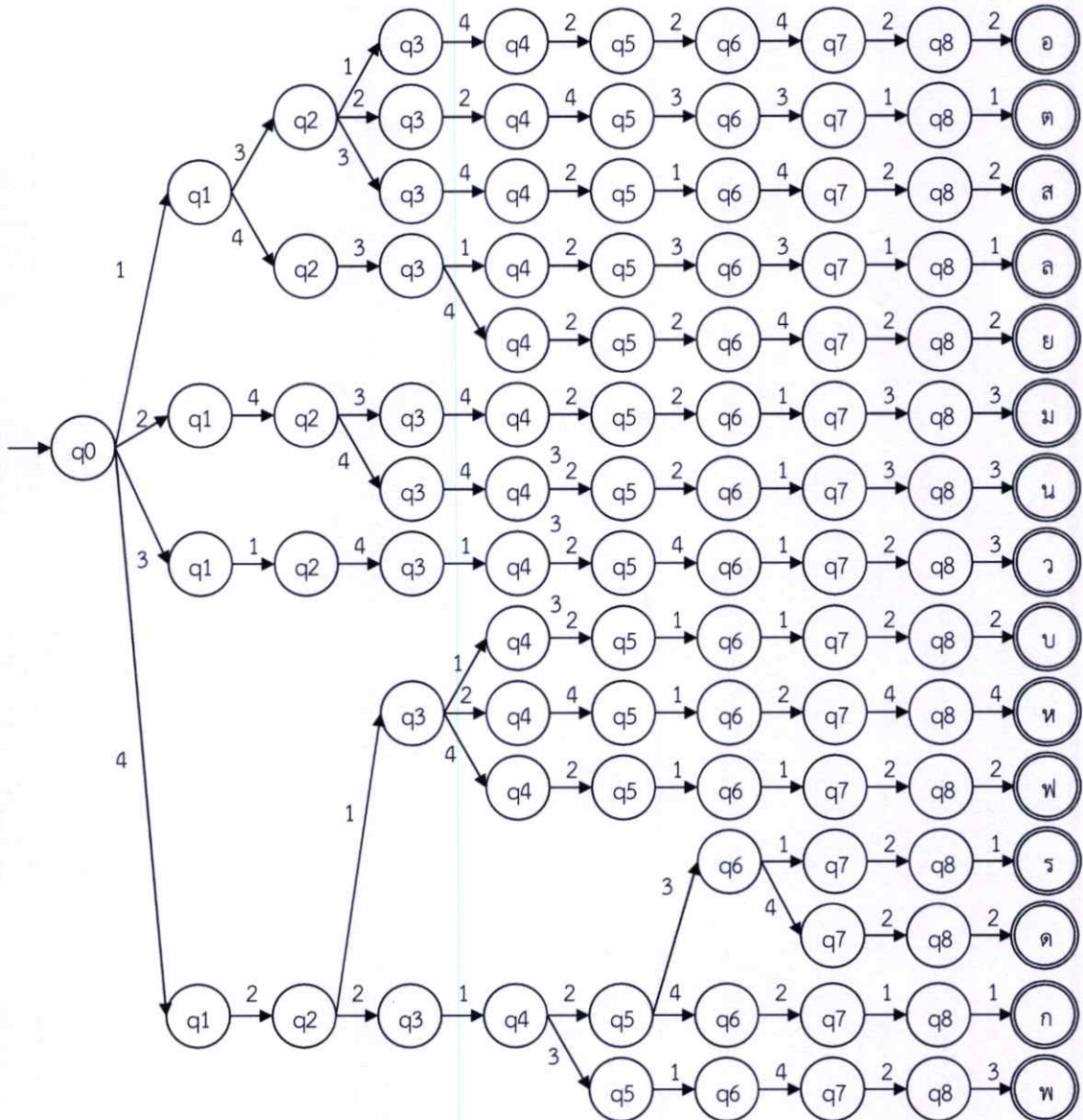
ตัวอักษร	จังหวะที่	นิ้วโป้ง	นิ้วชี้	นิ้วกลาง	นิ้วนาง	นิ้วก้อย
ธ	1	132	134	422	422	311
	2	134	241	244	422	422
	3	311	123	432	432	314
ช	1	131	134	134	134	421
	2	134	241	244	422	422
	3	311	123	432	432	314
ณ	1	131	134	134	134	421
	2	134	241	244	422	422
	3	311	123	212	314	314

4.3 ผลการหาลำดับและจำนวนของนิ้วมือ

การหาลำดับและจำนวนของนิ้วมือ เป็นขั้นตอนหลังจากการปรับค่าข้อมูลในยูทึ่ให้รูปแบบที่เหมาะสม เพื่อลดการประมวลผลและการจัดเก็บข้อมูลรหัสตัวอักษรภาษาไทย ซึ่งผลการหาลำดับและจำนวนของนิ้วมือที่เหมาะสมโดยไม่พิจารณาตัวอักษร “ม” และตัวอักษร “น” ที่มีรหัสเหมือนกันทั้ง 5 นิ้วมือ ทำให้ไม่สามารถจำแนกตัวอักษรได้ คือ นิ้วโป้ง นิ้วชี้ และนิ้วกลาง ตามลำดับ

4.4 ผลการสร้างต้นแบบอโตมาตาแบบจำกัดภาษามือตัวอักษรภาษาไทย

หลังจากเข้ารหัสภาษามือตัวอักษรภาษาไทยจำนวน 24 ท่าทางดังรูปที่ 3.9 และหาลำดับและจำนวนของนิ้วมือโดยใช้อัลกอริทึมดังรูปที่ 3.11 ขั้นตอนต่อไปเป็นการสร้างต้นแบบอโตมาตาแบบจำกัดภาษามือตัวอักษรภาษาไทยโดยใช้อัลกอริทึมดังรูปที่ 3.12 ตัวอย่างผลลัพธ์การต้นแบบอโตมาตาแบบจำกัดภาษามือตัวอักษรภาษาไทย 1 จึงหวะแสดงดังรูปที่ 4.26

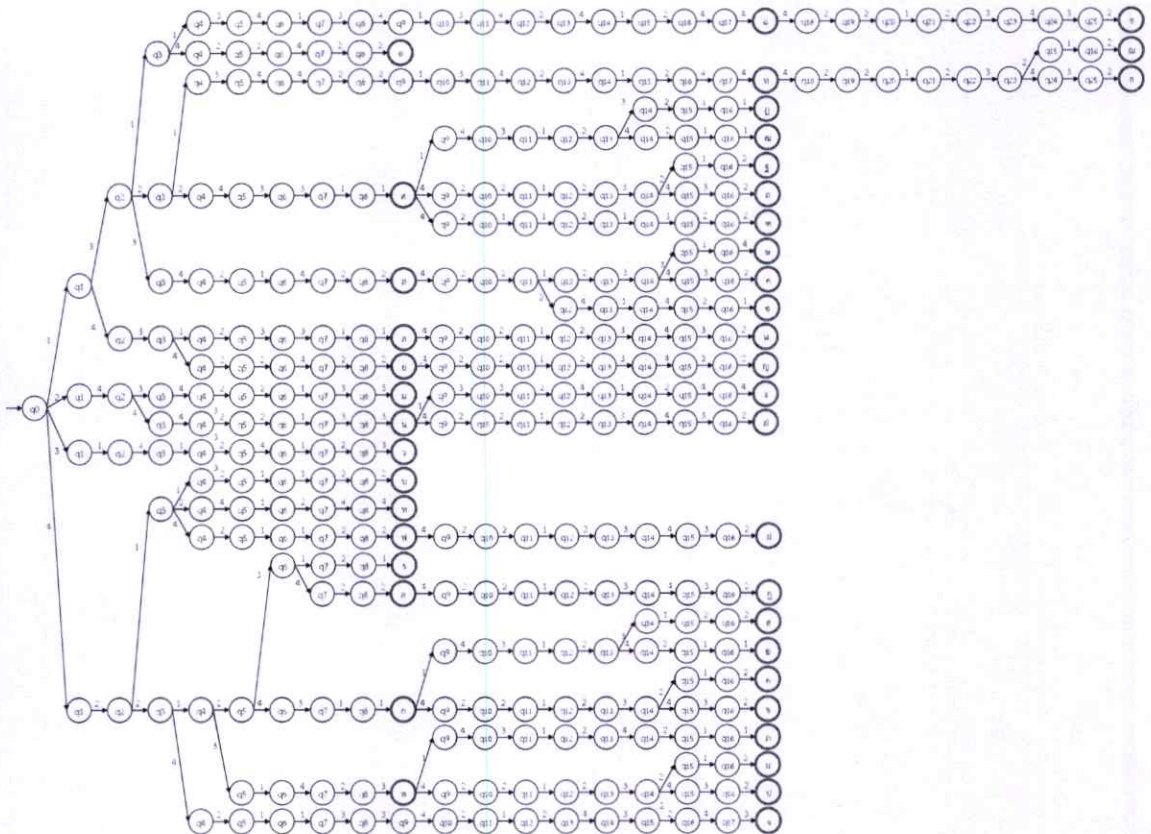


รูปที่ 4.26 ตัวอย่างอโตมาตาแบบจำกัดตัวอักษรภาษาไทย 1 จึงหวะ จำนวน 15 ตัวอักษร

จากรูปที่ 4.26 แสดงอโตมาตาแบบจำกัดตัวอักษร “ต” โดยนำรหัสภาษามือแต่ละนิ้วมือนิ้วมือตัวอักษรภาษาไทยจึงหวะที่ 1 จากตารางที่ 4.2 พิจารณาตัวอักษร “ต” เนื่องจากข้อมูลรหัสการเข้ารหัสรูปแบบ THSL การสะกดนิ้วมือ 1 จึงหวะ นิ้วโป้ง มีเพียงตัวอักษร “ต” ที่รหัสเป็น “132” จึงกำหนดให้โหนดสุดท้ายหรือโหนดในรหัสหลักที่ 3 เป็นสถานะยอมรับ

พิจารณาตัวอักษร “ล” เนื่องจากรหัสนิ้วโป้งของตัวอักษร “ล” กับตัวอักษร “ย” มีรหัสเดียวกัน จึงนำรหัสนิ้วชี้ของตัวอักษร “ล” มาสร้างโหนดต่อจากโหนดนิ้วโป้ง และเมื่อพิจารณารหัสนิ้วชี้ตัวอักษร “ล” กับ “ย” มีความแตกต่างกัน จึงกำหนดให้โหนดสุดท้ายหรือโหนดในรหัสหลักที่ 3 ของนิ้วชี้เป็นสถานการณ์ย่อยรับ

เมื่อนำรหัสภาษามือแต่ละนิ้วตัวอักษรภาษาไทยทั้งหมดที่ 2 และทั้งหมดที่ 3 จากตารางที่ 4.3 และตารางที่ 4.4 มาสร้างต่อจากรูปที่ 4.26 จะได้ต้นแบบออโตมาตาแบบจำกัดตัวอักษรภาษาไทยจำนวน 40 ตัวอักษรแสดงดังรูปที่ 4.27



รูปที่ 4.27 ต้นแบบออโตมาตาแบบจำกัดตัวอักษรภาษาไทยจำนวน 42 ตัวอักษร

4.5 ผลการทดสอบการแปลงภาษามือเป็นตัวอักษร

จากการทดสอบการแสดงผลทางภาษามือแปลงกับบุคคลทั่วไปจำนวน 30 คน โดยใช้อัลกอริทึมในหัวข้อ 3.8 ผลลัพธ์ความถูกต้องในการแปลงภาษามือเป็นตัวอักษรภาษาไทยแบ่งตามจังหวัดการเคลื่อนไหวแสดงดังตารางที่ 4.5 ถึงตารางที่ 4.7 ตามลำดับ

ตารางที่ 4.5 เปอร์เซ็นต์ความถูกต้องในการแปลงภาษามือเป็นตัวอักษรภาษาไทยการสะกดนิ้วมือ
1 จังหวะ จำนวน 15 ตัวอักษร

ตัวอักษร	เปอร์เซ็นต์ความถูกต้อง	ตัวอักษร	เปอร์เซ็นต์ความถูกต้อง
ก	70.00%	ล	63.33%
ต	80.00%	ย	56.67%
ส	73.33%	ม	33.33%
พ	73.33%	น	43.33%
ห	83.33%	อ	80.00%
บ	86.67%		
ร	90.00%		
ว	76.67%		
ด	86.67%		
ฟ	66.67%		

จากตารางที่ 4.5 พบว่ามีค่าเฉลี่ยความถูกต้องในการแปลงภาษามือเป็นตัวอักษรภาษาไทยการสะกดนิ้วมือ 1 จังหวะเป็น 70.89% พิจารณาแต่ละตัวอักษรจะเห็นได้ว่าตัวอักษรที่มีค่าเฉลี่ยความถูกต้องมากที่สุดคือตัวอักษร “ร” รองลงมาคือตัวอักษร “บ” และตัวอักษร “ด” ส่วนตัวอักษรที่มีค่าเฉลี่ยความถูกต้องต่ำกว่า 50% ได้แก่ตัวอักษร “ม” และตัวอักษร “น”

ตารางที่ 4.6 เปอร์เซ็นต์ความถูกต้องในการแปลงภาษามือเป็นตัวอักษรภาษาไทยการสะกดนิ้วมือ
2 จังหวะ จำนวน 24 ตัวอักษร

ตัวอักษร	เปอร์เซ็นต์ความถูกต้อง	ตัวอักษร	เปอร์เซ็นต์ความถูกต้อง
ข	63.33%	ผ	63.33%
ค	76.67%	ภ	63.33%
ฌ	76.67%	ฮ	66.67%
ถ	80.00%	ฎ	73.33%
ฐ	86.67%	ฬ	83.33%
ฒ	73.33%	ฬ	70.00%
ฑ	83.33%	จ	60.00%
ฎ	66.67%	ญ	50.00%
ศ	80.00%	ณ	56.67%

ตารางที่ 4.6 (ต่อ) เปอร์เซ็นต์ความถูกต้องในการแปลงภาษามือเป็นตัวอักษรภาษาไทยการสะกด
นิ้วมือ 2 จังหวะ จำนวน 24 ตัวอักษร

ตัวอักษร	เปอร์เซ็นต์ความถูกต้อง	ตัวอักษร	เปอร์เซ็นต์ความถูกต้อง
ษ	80.00%	ง	76.67%
ช	60.00%	ท	73.33%
ป	83.33%	ฉ	80.00%

จากตารางที่ 4.6 พบว่ามีค่าเฉลี่ยความถูกต้องในการแปลงภาษามือเป็นตัวอักษรภาษาไทยการสะกดนิ้วมือ 2 จังหวะเป็น 71.94% โดยตัวอักษรทั้ง 24 ตัวอักษรที่มีค่าเฉลี่ยความถูกต้อง 50.00% ขึ้นไป พิจารณาแต่ละตัวอักษรจะเห็นได้ว่าตัวอักษรที่มีค่าเฉลี่ยความถูกต้องมากที่สุดคือตัวอักษร “ฐ” ส่วนตัวอักษรที่มีค่าเฉลี่ยความถูกต้องน้อยที่สุดคือตัวอักษร “ฬ” และตัวอักษร “ญ”

ตารางที่ 4.7 เปอร์เซ็นต์ความถูกต้องในการแปลงภาษามือเป็นตัวอักษรภาษาไทยการสะกดนิ้วมือ
3 จังหวะ จำนวน 3 ตัวอักษร

ตัวอักษร	เปอร์เซ็นต์ความถูกต้อง
ธ	82.22%
ช	83.33%
ณ	80.00%

จากตารางที่ 4.7 พบว่ามีค่าเฉลี่ยความถูกต้องในการแปลงภาษามือเป็นตัวอักษรภาษาไทยการสะกดนิ้วมือ 3 จังหวะเป็น 81.85% โดยตัวอักษรทั้ง 3 ตัวอักษรที่มีค่าเฉลี่ยความถูกต้อง 80.00% ขึ้นไป พิจารณาแต่ละตัวอักษรจะเห็นได้ว่าตัวอักษรที่มีค่าเฉลี่ยความถูกต้องมากที่สุดคือตัวอักษร “ช” ส่วนตัวอักษรที่มีค่าเฉลี่ยความถูกต้องน้อยที่สุดคือตัวอักษร “ณ”

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

การแปลงท่าทางภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทย โดยใช้อุปกรณ์ลิโม่ชั้นในการตรวจจับตำแหน่งแกน x และ y ของมือและนิ้วมือ ซึ่งผู้วิจัยทำการจัดเก็บข้อมูลจากผู้ทดสอบบุคคลทั่วไปจำนวน 30 คน ข้อมูลได้จากอุปกรณ์ลิโม่ชั้นนำมาทำให้อยู่ในรูปแบบที่เหมาะสมเพื่อลดความผิดพลาดในการแปลงภาษามือเป็นตัวอักษรภาษาไทย และทำการหาค้นหาตัวอักษรภาษาไทยจากตำแหน่งแกน x และ y ของมือและนิ้วมือโดยใช้อัลกอริทึมการค้นหาอัตโนมัติแบบจำกัดตัวอักษรภาษาไทยในหัวข้อ 3.8 พบว่าความถูกต้องในการแปลงภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทยจำนวน 42 ตัวอักษรมีค่าเฉลี่ย 72.27% เนื่องจากการเก็บข้อมูลภาษามือของบุคคลทั่วไปบางท่านไม่มีความรู้ทางภาษามือ และลักษณะท่าทางภาษามือที่มีความใกล้เคียงกันทำให้มีการจำแนกตัวอักษรผิดพลาด พิจารณาแต่ละประเภทการสะกดนิ้วมือพบว่าการแปลงภาษามือเป็นตัวอักษรภาษาไทยการสะกดนิ้วมือ 3 จังหวะ มีค่าเฉลี่ยความถูกต้อง 81.85% ซึ่งมากกว่าการแปลงภาษามือเป็นตัวอักษรภาษาไทยการสะกดนิ้วมือ 1 จังหวะที่มีค่าเฉลี่ยความถูกต้อง 70.89% และการสะกดนิ้วมือ 2 จังหวะที่มีค่าเฉลี่ยความถูกต้อง 71.94% เนื่องจากจำนวนท่าทางการสะกดนิ้วมือ 3 จังหวะที่มากขึ้นทำให้มีการจำแนกตัวอักษรได้ถูกต้องมากกว่าสะกดนิ้วมือ 1 จังหวะ และ 2 จังหวะ

เมื่อพิจารณาตัวอักษรที่มีความถูกต้องมากที่สุดคือตัวอักษร “ร” มีค่าเฉลี่ยความถูกต้อง 90.00% เนื่องจากมีลักษณะท่าทางที่ไม่ซ้ำกับท่าทางของตัวอักษรอื่น รองลงมาคือตัวอักษร “บ” “ด” และตัวอักษร “ฐ” ที่มีค่าเฉลี่ย 86.67% เมื่อพิจารณาพบว่าตัวอักษรที่มีค่าเฉลี่ยความถูกต้องสูงเนื่องจากมีลักษณะท่าทางภาษามือที่สามารถตรวจจับได้ง่าย ไม่มีการซ้อนทับกันของนิ้วมือ ส่วนค่าเฉลี่ยความผิดพลาดของตัวอักษร “ร” มีค่าเฉลี่ย 10.00% และตัวอักษร “บ” “ด” รวมทั้งตัวอักษร “ฐ” ที่มีค่าเฉลี่ยความผิดพลาด 13.33% เกิดจากขนาดนิ้วมือแต่ละคนมีความยาวไม่เท่ากัน ตัวอักษรที่มีความถูกต้องในการแปลงเป็นตัวอักษรน้อยที่สุดคือตัวอักษร “ม” และตัวอักษร “น” เนื่องจากลักษณะท่าทางภาษามือแต่ละนิ้วมีความใกล้เคียงกันมาก โดยมีเพียงนิ้วนางเท่านั้นที่แตกต่างกัน และเนื่องจากผู้วิจัยได้ใช้การตรวจจับท่าทางของมือและนิ้วมือรูปแบบ 2 มิติ จึงทำให้เกิดความผิดพลาดในการแปลงภาษามือเป็นตัวอักษรภาษาไทย

จากผลการทดสอบการแปลงท่าทางภาษามือให้อยู่ในรูปแบบตัวอักษรภาษาไทย โดยใช้ อุปกรณ์ลิโม่ชั้นสามารถนำมาสร้างโปรแกรมแปลงภาษามือเพื่อใช้ในการสื่อสารระหว่างผู้ที่มีลักษณะความบกพร่องทางการได้ยินหรือสื่อความหมายกับบุคคลทั่วไป และสามารถนำไปเป็นแบบฝึกและแบบทดสอบการแสดงท่าทางภาษามือเป็นตัวอักษรภาษาไทยได้

5.2 ข้อเสนอแนะ

- 1) ควรจัดเก็บข้อมูลในรูปแบบ 3 มิติ เพื่อเพิ่มความถูกต้องในการแปลภาษามือ และการวิเคราะห์ที่ง่ายมากยิ่งขึ้น
- 2) พิจารณาการจัดข้อมูลให้อยู่ในรูปแบบที่เหมาะสมในระดับแต่ละนิ้วมือ เนื่องจากแต่ละบุคคลมีนิ้วแต่ละนิ้วยาวไม่เท่ากัน

บรรณานุกรม

- [1] สำนักสถิติพยากรณ์ สำนักงานสถิติแห่งชาติ, “การสำรวจความพิการ พ.ศ. 2550,” 2551, หน้า 6.
- [2] ศูนย์บริการถ่ายทอดการสื่อสารแห่งประเทศไทย. *ตู้ TTRS ตู้สื่อสารสำหรับคนพิการทางการได้ยิน ลดช่องว่างทางการสื่อสาร* [ออนไลน์]. เข้าถึงได้จาก: <http://ttrs.or.th/index.php/submenu1/news/156-it24hrs-ttrs>
- [3] Ratthawit Nuritanon. *Leap Motion Controller เทคโนโลยีการสั่งงานด้วยมือ* [ออนไลน์]. เข้าถึงได้จาก: <http://itsasontad.blogspot.com/2013/08/leap-motion-controller.html>
- [4] มลิวัดย์ ธรรมแสง, “*ล่ามภาษามือ,*” กรุงเทพฯ : จงเจริญการพิมพ์, ภาษามือ เล่ม 2, 2539.
- [5] มูลนิธิอนุเคราะห์คนหูหนวก ในพระบรมราชินูปถัมภ์, เว็บไซต์มูลนิธิพัฒนาคนพิการไทย. *การสะกดนิ้วมือไทย* [ออนไลน์]. เข้าถึงได้จาก: <http://www.tddf.or.th/signlang/?contentid=0137>
- [6] ณัฐนันท์ แนวมาล, “โครงการออกแบบชุดอักษรสัญลักษณ์ภาษามือในระบบภาษาไทยและอเมริกัน สำหรับผู้ทางได้ยินกรณี ศึกษาโรงเรียนเศรษฐเสถียร ในพระราชูปถัมภ์,” *การประชุมวิชาการและเสนอผลงานวิจัย/สร้างสรรค์ระดับชาติและนานาชาติ “ศิลปการวิจัยและสร้างสรรค์ ครั้งที่ 5,* 25-27 มกราคม 2555, หน้า 939-946.
- [7] วัฒนพงศ์ สุทธภักดิ์, *การใช้งานชิพต์ร่วมกับวิธีการจัดกลุ่มในระบบการแปลภาษามือไทยด้วยวิธีการสะกดนิ้วมือ,* มหาวิทยาลัยเชียงใหม่, หน้า 8 ปี 2553, วิทยานิพนธ์
- [8] inkirby, Blognone. *รีวิว Leap Motion Controller พร้อม Airspace* [ออนไลน์]. เข้าถึงได้จาก: <https://www.blognone.com/node/47147>
- [9] Meshack Musundi. *Leap Motion: Move Cursor* [ออนไลน์]. เข้าถึงได้จาก: <http://www.codeproject.com/Articles/550336/Leap-Motion-Move-Cursor>
- [10] coloringpages24x7. *Gallery for product concept drawings* [ออนไลน์]. เข้าถึงได้จาก: <http://coloringpages24x7.com/product+concept+drawings>
- [11] Vincent Bouvier. *Le kit de développement Leap Motion en photos* [ออนไลน์]. เข้าถึงได้จาก: <http://monleap.fr/537-kit-developpement-leap-motion-photos/>
- [12] Lisa Eadicicco, LAPTOP Staff Writer. *HP to Incorporate Leap Motion Gesture Controls* [ออนไลน์]. เข้าถึงได้จาก: <http://www.laptopmag.com/articles/leap-motion-controls-hp>

- [13] ajangz. *Quad Tree* [ออนไลน์]. เข้าถึงได้จาก: <http://ajangz.exteen.com/20150528/quad-tree>
- [14] Sasakubo1717, วิกิตำรา ตำราเสรีเพื่อโลกเสรี. *โครงสร้างข้อมูล/บทนำ* [ออนไลน์]. เข้าถึงได้จาก: <https://th.wikibooks.org/wiki/โครงสร้างข้อมูล/บทนำ>
- [15] Marpe, D., Schwarz, H., Boße, S., et al.: 'Video compression using nested quadtree structures, leaf merging and improved techniques for motion representation and entropy coding', *IEEE Trans. Circuits Syst. Video Technol.*, 2010, 20, pp. 1676–1687, Jul 2016.
- [16] Amitava Chakraborty, Sudip Kumar De, and Ranjan Dasgupta, "Balancing of Quad Tree using Point Pattern Analysis", *World Academy of Science, Engineering and Technology*, vol.5 pp.105-108, Apr 2011.
- [17] เกียรติกุล เจียรนัยธนะกิจ, "บทที่ 2 ทฤษฎีการคำนวณ (Theory of Computation)," พิมพ์ครั้งที่ 2. กรุงเทพฯ : สำนักพิมพ์สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2557, หน้า 20-23.
- [18] Peter Linz, "An Introduction to formal languages and Automata," 4th ed. London, UK: *Jones and Bartlett Publishers*, ch.1, pp.26–28, 2006.
- [19] Qutaishat Munib et al., "American sign language (ASL) recognition based on Hough transform and neural networks," *Expert Syst. Appl.*, pp. 24-37, Jan. 2007.
- [20] Suwannee Phitakwinai et al., "Thai Sign Language Translation Using Fuzzy C-Means and Scale Invariant Feature Transform," *ICCSA 2008*, pp.1107-1119, 30 Jun. - 3 Jul. 2008.
- [21] Veronica Yenquenida Flamenco Cordova, "Finger Placement Correction for Static Gesture Recognition in American Sign Language," M.S. thesis, Dept. Eng. Technol., Western Carolina Univ., NC28723, US, 2014.
- [22] วิสันต์ ตั้งวงษ์เจริญ และจิรวรรธ ตุ่มศรี, "การเข้ารหัสรูปแบบภาษามือตัวอักษรภาษาไทยโดยใช้ควอดทรี," *การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 37*, 19-21 พฤศจิกายน 2557, เล่มที่ 2, หน้า 873-876.
- [23] Michel Marie Deza, Elena Deza. *Encyclopedia of Distances. Springer, Sci. Bus. Media*, p.94, 2009.

ภาคผนวก ก
ผลงานวิจัยที่ตีพิมพ์



[IMECS Home Page](#)

International Association of Engineers

International MultiConference of Engineers and Computer Scientists 2017



[IMECS Home Page](#)



IMECS 2017 Proceedings CD

Conference Period and Venue:

15-17 March, 2017
The Royal Garden Hotel, Kowloon, Hong Kong

Organizing Association:



International Association of Engineers

The papers in the CD consist of the proceedings of the conference mentioned above. They reflected the authors' opinions and their inclusion in the publication does not necessary constitute endorsement by the conference committees and the International Association of Engineers. The publication ethics and publication malpractice statement of our conference is available in the conference web page:

http://www.iaeng.org/IMECS2017/doc/publication_ethics.html

Copyright and Reprint Policy:

Abstracting is permitted with credit to the source. Libraries may photocopy the articles for private use of patrons in this proceedings publication. Copying of individual articles for non-commercial purposes is permitted without a fee, provided that credit to the source is given. For other copying, reprint, or republication permission purposes, permission from the publisher should be obtained.

Publisher:
Newswood Limited
International Association of Engineers

Proceedings Books:
ISDN (Vol I): 978-988-14047-3-2
ISBN (Vol II): 978-988-14047-7-0
ISSN: 2078-0958 (Print)
ISSN: 2078-0966 (Online)

Additional copies may be ordered from:
Newswood Limited
Unit 1, 1/F, 37-39 Hung To Road, Hong Kong

IAENG

E
ENGINEERING
LETTERS

IJCS
IAENG International Journal of
Computer Science

IJAM
IAENG International Journal of
Applied Mathematics



[IMECS Home Page](#)

International Association of Engineers

International MultiConference of Engineers and Computer Scientists 2017



[IMECS Home Page](#)

The 2017 IAENG International Conference on Artificial Intelligence and Applications

Home
Preface
Committees
Invited Talks
Conference Topics
ICAI17
ICBI17
ICCA17
ICCS17
ICCSA17
ICDMA17
ICBE17
ICEE/CAFICSE17
ICEP17
ICICWS17
ICIE17
ICINDE17
ICINDE(HES)17
ICOR17
ICSC17
ICSE17
WSCF17
WSMSE17
Coming Conferences

Original Papers

Fuzzy Modeling using Vector Quantization based on Input and Output Learning Data

Hirofumi Miyajima, Noritaka Shigei, and Hiromi Miyajima ----- pp1-6 [[Full Text](#)]

Human Action Recognition by Conceptual Features

Yu-Huei Cheng, and Ching-Ming Lai ----- pp7-13 [[Full Text](#)]

A Study on Applying Interactive Multi-objective Optimization to Multiagent Systems

Toshihiro Matsui ----- pp14-19 [[Full Text](#)]

Application of Backpropagation Neural Networks in Predicting Story Drift of Building

Reni Suryanita, and Hendra Jingga ----- pp20-23 [[Full Text](#)]

Cross-language Entity Linking Adapting to User's Language Ability

Jialiing Zhou, Fuminori Kimura, and Akira Maeda ----- pp24-29 [[Full Text](#)]

An Effective Computational Intelligence-Based Method for Mutagenic Primer Design

Yu-Huei Cheng, and Ching-Ming Lai ----- pp30-35 [[Full Text](#)]

Evaluation of Shaft Bearing Capacity of Single Driven Pile using Neural Network

Harnedi Maizir ----- pp36-39 [[Full Text](#)]

Story Creation for Fostering Historical Thinking Skill and its Support System

Yuta Miki, and Tomoko Kojiri ----- pp40-45 [[Full Text](#)]

Thai Sign Language Translation Using Leap Motion Controller

Jirawat Tumsri, and Warangkhana Kimpan ----- pp46-51 [[Full Text](#)]

A Reinforcement Learning for Freight Train with Augmented Collective Motions in a Marshaling

Yoichi Hirashima ----- pp52-57 [[Full Text](#)]

Thai Sign Language Translation Using Leap Motion Controller

Jirawat Tumsri and Warangkhan Kimpan

Abstract— In this article, algorithm for transforming sign language into Thai alphabets is presented to fill the gap in communication between the hearing impairment and normal people. Leap Motion Controller was applied to detect 5-finger-tip position and palm center in the form of X and Y axis. Then, decision tree was created by using the Quadtree technique and the research result on transforming Thai sign language into finite automata was applied to improve algorithm in creating finite automata of Thai alphabet sign language to increase efficiency and speed in processing sign language. The test result shows that it can discriminate 44-Thai alphabet sign language at 72.83% accuracy.

Index Terms— Finite automata, Hearing impairment people, Leap Motion Controller, Quadtree, Thai sign language

I. INTRODUCTION

IN daily life, human beings always need interpersonal communication. Communication is, therefore, an essential basic factor to be in a society. Normal people communicate mainly through speaking but the hearing impairment people use the sign language. One of the barriers in communication between the hearing impairment people and the normal people is that the normal people do not understand the sign language thus making a little difference in a sign language expression can completely change the meaning in communication.

The statistics during 2012-2014 indicated that 51.0% of the hearing impairment people need help [1]. The public sector has set the policy in helping and promoting the hearing impairment people to be able to communicate with normal people by developing Thailand Telecommunication Relay Service (TTRS) machine or the communication machine for the hearing impairment people to fill the gap in communication [2]. However, there is still a problem of insufficient TTRS machine service as there are only 120 service points available in various places, moreover, there is an accessibility problem for the hearing impairment people. When using TTRS machine service, the hearing impairment people need provided sign language translators.

Manuscript received December 27, 2016; revised February 11, 2017.

Jirawat Tumsri is with the Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang (KMIL), Chalongkrung Rd., Ladkrabang, Bangkok, 10520 Thailand (e-mail: 57605074@kmitl.ac.th).

Warangkhan Kimpan is with the Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang (KMIL), Chalongkrung Rd., Ladkrabang, Bangkok, 10520 Thailand (e-mail: warangkhan.ki@kmitl.ac.th).

The survey on the performance of public sector during 2012-2014, shows that the average of the hearing impairment people's quality of life is at a stable level (35.3%) and that there are 37.3% of the hearing impairment people who still face problems and difficulties in living. This shows that the service of the public or private sectors is still insufficient for the hearing impairment people to improve their quality of life [1].

According to the problems mentioned above and in order to help hearing impairment people to be able to communicate with normal people, this research focuses in developing computer program that can translate the sign language into Thai alphabets by applying Leap Motion Controller. This equipment detects the fingers and then Quadtree and finite automata for classifying are used in order to transform the position of each finger into a Thai alphabet.

II. THEORY AND RELATED WORKS

A. Thai Sign Language

Communication can be divided into 2 types: verbal communication in which words and alphabets are commonly used in socialization and non-verbal communication in which actions, gestures, tunes, eyesight, objects, signals, environments, and other expressions are used in communication. Showing hand symbols or sign language is a type of non-verbal communication for hearing impairment people.

In 1956, Khun Ying Kamala Krailuek designed the standard Thai sign language (THSL) by applying the fingerspelling principle of American sign language (ASL) in single hand fingerspelling. THSL consists of 44 alphabets and 23 vowels [3] as shown in Fig. 1.

1-step movement	2-step movement	3-step movement
ก:	ข:	ค:
ฅ:	ฉ:	ช:
ซ:	ฌ:	ฎ:
ด:	ต:	ถ:
ธ:	ดล:	ดฬ:
	ฉล:	ฉฬ:
	ชล:	ชฬ:
	ซล:	ซฬ:
	ฌล:	ฌฬ:
	ฎล:	ฎฬ:
	ดลฬ:	ดฬฬ:
	ตลฬ:	ถลฬ:
	ธลฬ:	

Fig. 1. Thai sign language diagram.
 (This figure is derived from the one described in [4] which already had a copyright permission from EECON37.)

Fig. 1 shows the Thai sign language in which the Thai alphabets are classified into 3 types according to the steps of movement as follows: Type I) 1-step movement in fingerspelling, consisting of 18 alphabets, e.g., η, θ, κ , and so on. Type II) 2-step movement in fingerspelling, consisting of 23 alphabets, e.g., υ, σ, π , and so on. Type III) 3-step movement in fingerspelling, consisting of 3 alphabets, α, υ , and ω [4].

B. Quadtree

Quadtree is a type of spatial data structures, similar to Q-tree in object-based image classification for solving the problem of extracting a large amount of data at the same time. Moreover, it can also reduce the number of irrelevant data extractions by continuously dividing the area in a two-dimension square into 4 equal parts in the form of plus sign, resulting in four and sixteen squares respectively. Then the decision tree is created from the results of the square division as shown in Fig. 2.

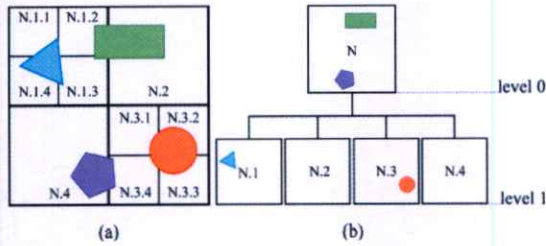


Fig. 2. Quadtree division for storing objects in decision tree.

Fig. 2 (a) shows the square N before dividing. The decision tree creates 1 node in level 0, called "root node", which is the node N as shown in Fig. 2 (b). After dividing the square N into 4 parts and comparing it to the decision tree in Fig. 2 (b), it can be indicated that the node N in level 0 consists of 4 child nodes in level 1 which are N.1, N.2, N.3, and N.4. The division of nodes in level 1 is the last division, therefore the child nodes in this level are called "leaf node". Every node in the decision tree stores the objects which can be the member of only one node. However, in each node, many objects can be stored. The objects can be the members of node N when the criteria are as follows: [5]

- 1) The objects are placed in N (triangle, rectangle, circle, and pentagon).
- 2) If the objects are placed in any child nodes of N, they are the members of those child nodes (triangle and circle).
- 3) If the objects are placed astride child nodes of N, they are the members of node N (rectangle and pentagon).

The Quadtree is mostly used in an object-based image classification, such as finding various positions on the map, providing spatial details of games, and so on. It is good for systematic data storage and reduction of the number of the other irrelevant space examinations.

C. Finite Automata

Finite automata is a model for studying the operation of computer. The overall elements of automata are divided into 3 parts consisting of input alphabet tape, processor, and data

storage. The input tape stores alphabets and sends them to the processor, the processor controls internal state that changes according to the input, whereas the storage stores the outcome obtained during the calculation or processing [6].

Finite automata consists of 2 types: deterministic finite automata and non-deterministic finite automata. The definition of deterministic finite automata can be explained by substitution of 5 factors as shown in (1) [7].

$$M = (Q, \Sigma, \delta, q_0, F) \tag{1}$$

When Q is any state finite sets

Σ is alphabet finite sets

$\delta : Q \times \Sigma \rightarrow Q$ is a transition function

q_0 is a initial state, for $q_0 \in Q$

F is a set of acceptance state for $F \subseteq Q$ and F is a final state assuming that finite automata accepts the input alphabet tape

The transitive diagram definition of finite automata in each diagram is the graph that identifies the direction as the followings:

- 1) There is Q as a set of all points and there are arrows pointing to the starting point.
- 2) There are 2 circles enclosing all points in F .
- 3) There is a line connecting point p to point q with the input symbol a , written as follows: $\delta(p, a) = q$

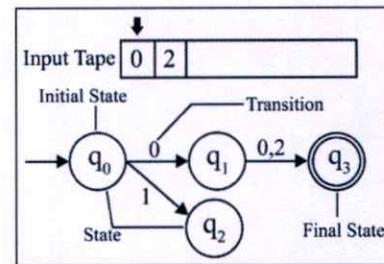


Fig. 3. Diagram of automata elements.

Fig. 3 is a written example of finite automata. It is merely one-way. In each movement of pointer at the input tape, automata will always be in the set of some states. When the reading starts, the input is 0. Therefore, the direction is from the starting state (q_0) to state (q_1). Then, the pointer is moved to the right for 1 alphabet. The process is repeated until all the input is completely read. If the current state is in the set of acceptance state (q_3), it shows that the obtained outcome is the acceptance state.

From (1), it can be substituted as follows: $M = (\{q_0, q_1, q_2, q_3\}, \{0, 1, 2\}, q_0, q_3)$ which can be written in the table format showing the state of receiving the input of 0 and 2 as in Table I:

TABLE I
STATE TRANSITION

State / Event	0	1	2
q_0	q_1	q_2	-
q_1	q_3	-	q_3
q_2	-	-	-
q_3	-	-	-

D. Leap Motion Controller

Leap Motion Controller is a sensor as shown in Fig. 4. It is used for detecting the movement position of hands and fingers. It can detect the movement of every knuckle and palm center with the precision at the level of 0.01 mm.

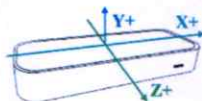


Fig. 4. 3-dimension detecting direction.

The limitation of Leap Motion Controller application is that too much or too little light affects the precision of infrared in the equipment in detecting the objects, and it cannot detect the object within 10 millimeters from the equipment, also the overlapped objects cannot be detected as the only largest one can be detected. Besides, the objects placed too close to each other may be detected as only one object.

E. Related Works

In 2007, Qutaishat Munib, Moussa Habeeb, Bayan Takturi, and Hiba Abed Al-Malik [8] presented the study on the development of automatic system for transforming American sign language by applying image processing to transform image into vector and comparing to the vector using neural network technique in learning the gesture set of sign language, which the rotation of gesture in the image did not affect the data extraction. The result of system test, by using 300 sample images of hand signals, revealed that the system could perceive the sign language gesture with 92.3% accuracy.

In 2008, Suwanee Phitakwinai, Sansanee Auephanwiriyaikul, and Nipon Theera-Umpon [9] proposed the study on Thai sign language translation using Fuzzy C-Means (FCM) and Scale Invariant Feature Transform (SIFT), which was the transform of sign language into 15 alphabets and 10 words by using camera as an input in processing. The algorithms used in this study were FCM and SIFT. The experiment result revealed that the system could transform sign language into alphabets and words with the accuracy of 82.19% and 55.08%, respectively. Whereas the nearby neighbor was at 3 and SIFT threshold was 0.7.

In 2014, Wisan Tangwongcharoen and Jirawat Tumsri [4] presented the study on decoding Thai sign language pattern using Quadtree and using Leap Motion Controller in detecting 5-fingertip positions. Those fingertip positions were analyzed by using Quadtree technique. The outcome consisted of the number codes for 44 alphabets, 3 digits for each one.

In 2015, Jirawat Tumsri and Warangkhan Kimpan [10] presented the study on code translation from the Thai sign language pattern to finite automata by creating finite automata from 1-step fingerspelling codes of 18 alphabets, one finger for 3-digit code. The obtained result was 1-step fingerspelling finite automata structure.

In this study, Quadtree technique [4] was applied in transforming finger positions into finger codes for 44 alphabets, and finite automata algorithm [10] was improved for supporting 3-step fingerspelling.

III. PROPOSED METHOD

A. Overview of the Method

The methods is divided into 2 main parts: creating finite automata prototype for Thai sign language as shown in Fig. 5 (a) and transforming sign language into Thai alphabet pattern as shown in Fig. 5 (b).

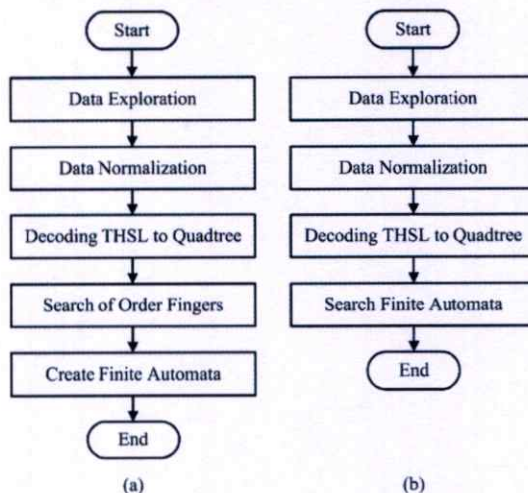


Fig. 5. 2 main parts of overview method.

From Fig. 5 (a) the creation of finite automata for Thai sign language starts by collecting the samples of Thai sign language in the pattern of the position on X and Y axis. Then, the sample data values are adjusted into a proper pattern, then Thai sign language is decoded. The obtained result in each finger is a 3-digit number. Then, the 3-digit code is used to find out finger order and number for creating finite automata prototype for Thai sign language in order to reduce the number of data or fingers. After getting the finger order and number, the finite automata prototype for Thai sign language is created.

Fig. 5 (b) shows the steps in transforming sign language into Thai alphabet pattern. It starts by the user showing the sign language through Leap Motion Controller and then the data are sent to the program. The value of the data, which are in the pattern of positions on X and Y axis, is adjusted to be in a proper pattern. Then, the positions on X and Y axis with the adjusted value are decoded into Thai sign language and the outcome of 1 alphabet is found out from finite automata prototype for Thai sign language.

B. Data Exploration

In order to collect the prototype data or the transformation of sign language into Thai alphabet pattern, the hand gestures must be in the parallel position to Leap Motion Controller. First, hand gesture, open hand or waving hand, must be shown for 3 seconds so that the hand sizes of both the master and examiner can be adjusted. After that the position of X and Y axis on fingertip and palm center are stored for 3 seconds for each gesture after started. The data of positions on X and Y axis on fingertips which derived from Leap Motion Controller, are used in this study as follows:

1) The creation of finite automata prototype for Thai sign language consists of positions on X and Y axis of 5 fingertips and on X and Y axis of palm center.

2) For transforming sign language into Thai alphabet pattern, receiving data value of each finger relies on the outcome of finding out finger order and number in the step of creating finite automata prototype for Thai sign language. For instance, the outcome of finding out finger order and number consists of thumb, index finger, and middle finger, while the position of X and Y axis are in the palm center.

C. Data Normalization

The adjustment of data value into the proper pattern is the master's and the examiner's hand size adjustment in order that their hands are in the same size, and determination of position on the same palm center is also done for accuracy in transforming sign language into Thai alphabet pattern.

The equation for finding the distance between 2 points in direct line or Euclidean distance [11] is shown in (2):

$$distance = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} \quad (2)$$

Where *distance* is the length from middle fingertip to palm center

X_1, Y_1 is X and Y axis on middle fingertip

X_2, Y_2 is X and Y axis on palm center

After the distance between middle fingertip and palm center is obtained, the X and Y axis positions are used in finding out the ratio as shown in (3) and the outcome of (3) is used to adjust the X and the Y axis positions of each finger as shown in (4) and (5) respectively. This adjustment is conducted to solve the problem of different hand sizes of the master and the examiner and reduce the error in transforming sign language into the Thai alphabet pattern.

$$ratio = distance_1 / distance_2 \quad (3)$$

Where *ratio* is the ratio between sign language of the master and examination

distance₁ is the length from middle fingertip to palm center of the master

distance₂ is the length from middle fingertip to palm center of the examination

$$X_{post} = X_{pre} * ratio \quad (4)$$

$$Y_{post} = Y_{pre} * ratio \quad (5)$$

Where X_{post}, Y_{post} is X and Y axis positions after adjusting the ratio

X_{pre}, Y_{pre} is X and Y axis positions before adjusting the ratio

ratio is the ratio between sign language of the master and examination

In addition to the adjustment of finger positions, there is also the shift of finger positions presenting the proper Quadtree table, which the position on palm center is always set as a fixed point in referring to other fingers by setting the palm center position on the midpoint of X and Y axis in the

Quadtree table. In doing so, it can cover all of the hand even though there is a shift position or changes of the positions of other fingers.

The shift of finger position is done by finding out the distance between the former palm center position (X_1, Y_1) and palm center position after being shifted to the midpoint of X and Y axis (X_2, Y_2) as shown in (2). After finding out the distance of the shift of palm center position, X and Y axis positions of each finger are adjusted as shown in (6) and (7).

$$X_{post} = X_{pre} + distance \quad (6)$$

$$Y_{post} = Y_{pre} + distance \quad (7)$$

Where X_{post}, Y_{post} is X and Y axis positions after adjusting the ratio

X_{pre}, Y_{pre} is X and Y axis positions before adjusting the ratio

distance is the length of shifting palm center position toward the midpoint of Quadtree table

The creation of finite automata prototype for THSL starts by finding out the value between middle fingertip and palm center as shown in (2). Then, X axis position is moved as shown in (6) and Y axis position as shown in (7) to transform sign language into the correct Thai alphabet pattern.

In adjusting data value into the proper pattern in the step of transforming sign language into Thai alphabet pattern, the values of X and Y axis positions on middle fingertip and on palm center are used in finding out the distance between 2 points as shown in (2). Then, the ratio of the distance is found out between middle fingertips and palm centers of the master and the examiner as shown in (3). After the ratio is obtained, the value of X axis position is adjusted as shown in (4) and of Y axis position as shown in (5). The next step is to move X axis position as shown in (6) and Y axis position as shown in (7).

D. Decoding THSL to Quadtree

Algorithm for decoding THSL by using Quadtree technique is a method for decoding the gesture of one finger. The input consists of X and Y axis positions on each finger, the number of alphabet codes, and the size of Quadtree square table, whereas the obtained outcome consists of 3-digit number code of each finger that can be transformed into the decision tree pattern as shown in Fig. 6. This result was extended from the one obtained in [4] which already had a copyright permission from EECON37.

E. Search of Order Fingers

Finger order search is a step in creating a finite automata prototype of THSL to reduce data storage and increase processing efficiency in transforming sign language into Thai alphabets. The data on X and Y axis of 5 fingers be entered and the outcome will be obtained in form of the finger order as described in the following algorithm.

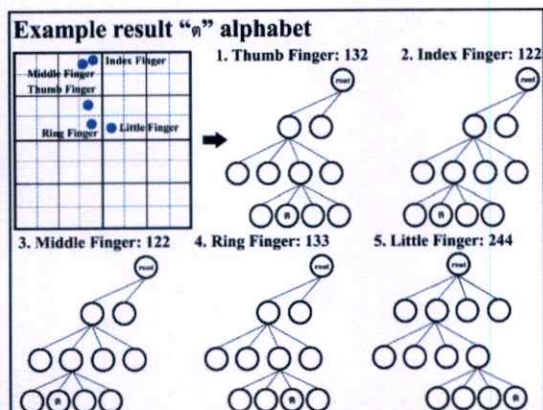


Fig. 6. Example of decision tree prototype for the alphabet "n" of 5 fingers.

Algorithm: Search of Order Fingers

```

1. found duplicate - true, number of finger - 5, finger number i - 1
2. While found duplicate
3.   Set temp finger number - 1, minimum code - 44
4.   While temp finger number <= number of finger
5.     number duplicate finger array - find code of finger duplicate
6.     If minimum code > number duplicate finger array Then
7.       minimum code - number duplicate finger array
8.       finger number i - type of finger
9.     End If
10.    Next temp finger number
11.  End while
12.  If found duplicate code of finger array Then
13.    Remove code of finger in temp finger number
14.    number of finger - number of finger - 1
15.    Next i
16.  Else
17.    found duplicate - false
18.  End If else
19. End while
20. Return finger number

```

The above algorithm is the steps of finger order and number search. First, the first finger order is searched by selecting to store each finger code of duplicate alphabet. For example, the codes on a thumb finger for the first, second, and third alphabets are 244, 123, and 244, respectively. Therefore, 244 is stored as an alphabet code of the thumb finger. Secondly, after all alphabets are completely searched, the algorithm examines whether the initial finger has more finger codes in storage than the current one. If so, the number of duplicated code is stored and the current finger is set as the first finger. Finally, the stored finger codes are examined whether there are any duplicate codes. If there are, the next finger or the second finger is then searched. In the search of the second finger, only the stored duplicate code of the first finger is considered and the outcome of the first finger from the initial search will not be considered. For example, if the outcome of the first finger is the thumb finger, the second round is, therefore, to consider the codes of index finger, middle finger, ring finger, and little finger. This process kept repeating until the duplicate finger code is no longer found or all of 5 fingers are completely considered. For example, the final outcome reveals that there are 3 fingers: the first one is the thumb finger, the second is the index finger, and the third is the middle finger.

F. Create Finite Automata

From (1), the factors of finite automata are set to be the code numbers of sign language in each digit of each finger and final state is the set of one alphabet. Therefore, the alphabet is encoded into sign language. Then, each digit is decoded by starting at the outcome of finger order and number search, which each finger has 3-digit code.

In algorithm for creating finite automata prototype, the input is THSL prototype for decoding 44 alphabets. The outcome is finite automata structure of THSL. The algorithm for creating finite automata prototype is shown below.

Algorithm: Create Finite Automata

```

1. alphabet = 1, number of alphabet = 44
2. While alphabet <= number of alphabet
3.   finger number = 1, current state = initial state
4.   While finger number <= size finger number
5.     Read code of finger number 3 integer
6.     For i = 1 to 3
7.       Get code of finger number integer i
8.       Find state code finger at current state
9.       If not found state code finger Then
10.        Create state finite automata
11.      End If
12.    Next i
13.  End for
14.  Next finger number
15. End while
16. state automata = final state
17. Next alphabet
18. End while
19. Return finite automata

```

The above algorithm is finite automata creating algorithm. Firstly, the current state pointing at the initial state and receiving 1-alphabet code is set. Then, the 3-digit code is read from the first finger, the first digit code is read, the current state is searched whether there is a movement direction toward another state of the first digit code. If it is not found, the state is created with the movement direction toward another state to be the first digit and the current state is set to point to the created state. The process is repeated until all of three digits are completely done. Secondly, when the state is created for all 3 digits, the code of the next finger is read and repeated the same process as creation steps of automata of the first finger code. Finally, when the creation is completed according to finger order and number for one alphabet, the automata of next alphabets are created until all of 44 alphabets are completely done.

Algorithm: Search Finite Automata

```

1. current state = first state of THSL finite automata
2. While EOF of input tape
3.   found = false
4.   Read value at current pointer of input tape
5.   While consider all transition or not found
6.     If value on transition == value of input tape Then
7.       Next state of current transition
8.       found = true
9.     Else
10.    Next transition of current state
11.  End if
12.  Next i
13. End if else
14. End while
15. Next pointer of input tape
16. End while
17. If state of current state == final state Then
18.   Get alphabet of current state
19. Return alphabet

```

G. Search Finite Automata

The inputs of search finite automata for Thai alphabets are the examiner's and the master's codes of each finger, and code for decoding in THSL pattern. The obtained outcome consists of Thai alphabets which is described in the algorithm of search finite automata.

At the starting, the current state is set to point to the first state. Then, the value is read from the pointer that points to the input tape by considering all directions of transition of current state. If value on transition is equal to value read from input tape, the current state point to transition state in that direction. Then, the pointer of the input tape is moved to the right and the direction of the transition is considered as done before. This process is repeated until the end of the input tape pointer, then this algorithm outcome from the acceptance state of current state is considered. If it is the acceptance state, the alphabet of current state is retrieved. If it is not, it reveals that no Thai alphabet is found.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

From the steps of THSL finite automata creation for 44 alphabets, the outcome from the step of decoding THSL to Quadtree was the code of 3-digit number of each alphabet on each finger. Then, the outcome from the step of decoding THSL to Quadtree was used for the search of finger order which reveals that three fingers namely thumb finger, index finger, and middle finger, can sequentially classify sign language of 44 Thai alphabets. The results of this step are shown in Table II. Finally, these results are used for creation of finite automata.

TABLE II
RESULT OF DECODING THSL TO QUADTREE METHOD

Alphabet	Thumb Finger	Index Finger	Middle Finger
๙	132	122	122
๙	422	133	422
๙	422	132	333
๙ (๙ + 1)	322 + 122	133 + 133	332 + 422
๙ (๙ + 1)	132 + 122	122 + 133	122 + 422
๙ (๙ + 1)	422 + 122	422 + 133	132 + 422
๙ (๙ + 1)	(132 + 422) + 122	(122 + 132) + 133	(122 + 133) + 422
๙ (๙ + 1)	(122 + 422) + 122	(132 + 132) + 133	(332 + 133) + 422
๙ (๙ + 2)	(122 + 422) + 422	(132 + 132) + 112	(332 + 133) + 333

Table II shows the code of 3-digit number of each alphabet on each finger. For example, the outcome in the second-movement group of alphabet "๙" was the results from the mixture of alphabet "๙" (thumb: 322, index finger: 133, middle finger: 332) and the number "1" (thumb: 122, index finger: 133, middle finger: 422). Whereas the outcome in the third-movement group of alphabet "๙" was the results from the mixture of alphabet "๙" in the second movement (the first movement, thumb: 132+422, index finger: 122+132, middle finger: 122+133) and the number "1" (thumb: 122, index finger: 133, middle finger: 422).

On the steps of transforming sign language into Thai alphabet pattern, the accurate outcome in transforming sign language into Thai alphabets is shown in Table III.

TABLE III
EXAMPLE RESULT OF SEARCH FINITE AUTOMATA

Alphabet	Accuracy	Alphabet	Accuracy	Alphabet	Accuracy
๙	80.00%	๙	63.33%	๙	75.56%
๙	73.33%	๙	80.00%	๙	73.33%
๙	86.67%	๙	83.33%	๙	66.67%

V. CONCLUSIONS

This research presented the transformation of sign language into Thai alphabets to help the hearing impairment people use to communicate with normal people. Two steps were applied in the study: 1) creation of the prototype of finite automata for sign language on 44 Thai alphabets which classified into 3 groups according to the finger movements and 2) transformation of sign language into Thai alphabets by using Quadtree and finite automata. The average score of obtained outcome accuracy in sign language transformation was 72.83%. The alphabets with the highest accuracy score, 86.67% were "๙" and "๙". Whereas the alphabets with the lowest accuracy score, 53.33% were "๙", "๙", and "๙".

REFERENCES

- [1] J. Bunyaratanasutorn, Proceeding Articles in 7th National Academic Conference on the Deft People, "Quality of Life of the Deft: Reflection of Monitoring and Evaluation of the 4th National Development Plan for Quality of Life of The Hearing Impairment People, 2015-2016", vol.2, pp. 33-58, 2015.
- [2] Thai Telecommunication Relay Service. TTRS Machine for the hearing impairment people: Reduction of communication gap [Online]. Available: <http://ttrs.or.th/index.php/submenu1/news/156-it24hrs-ttrs> [Nov 22, 2016].
- [3] M. Thammasaeng, "Sign Language," Bangkok: Jong Chareon Printing, Sign Language, vol. 2, 1996.
- [4] W. Tangwongcharoen and J. Tumsri, "Decoding Thai Sign Language Pattern using Quad Tree," in *Proc. of the 37th Electrical Engineering Conf. (EECON37)*, Khon Kaen, Thailand, vol. 2, pp. 873-876, 19-3 Nov, 2014.
- [5] ajangz. 2015. Quadtree [Online]. Available: <http://ajangz.exteen.com/20150528/quad-tree> [Nov 10, 2016].
- [6] K. Jiaranaithanakit, "Theory of Computation," 2nd ed. Bangkok: Technology Promotion Association (Thailand-Japan), 2014, ch. 2, pp. 20-23.
- [7] P. Linz, "An Introduction to formal languages and Automata," 4th ed. London, UK: Jones and Bartlett Publishers, 2006, ch. 1, pp. 26-28.
- [8] Q. Munib, M. Habeeb, B. Takruri, and H. A. Al-Malik, "American sign language (ASL) recognition based on Hough transform and neural networks", *Science Direct, Expert Syst. with Applicat.*, vol. 32, pp. 24-27, 2007.
- [9] S. Phitakwinai, S. Auephanwiriyaikul, and N. Theera-Umpon, "Thai Sign Language Translation Using Fuzzy C-Means and Scale Invariant Feature Transform," *ICCSA 2008*, pp. 1107-1119, 30 Jun-3 Jul 2008.
- [10] J. Tumsri and W. Kimpan, "Code Translation from Thai Sign Language Pattern to Finite Automata," in *Proc. of the 7th Conf. of Electrical Engineering Network of Rajamangala University of Technology 2015 (EENET 2015)*, A-one The Royal Cruise Hotel, Pattaya, Chonburi, Thailand, pp. 125-128, 27-29 May, 2015.
- [11] M. M. Deza and E. Deza, "Encyclopedia of Distances," Springer, Sci. Bus. Media, p. 94, 2009.



งานบัณฑิตศึกษาคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
คำรับรองวิทยานิพนธ์สำหรับการสำเร็จการศึกษาระดับบัณฑิตศึกษา

วันที่ 16 เดือน กรกฎาคม พ.ศ. 2561

ข้าพเจ้า นาย/นาง/นางสาว.....จิริวรรธ ตุ่มศรี.....รหัสประจำตัว.....57605074.....
 นักศึกษาหลักสูตรวิทยาศาสตรมหาบัณฑิต/ปรัชญาดุษฎีบัณฑิต สาขาวิชา.....วิทยาการคอมพิวเตอร์.....
 ขอรับรองว่าวิทยานิพนธ์เรื่อง

ชื่อภาษาไทยการพัฒนาโปรแกรมแปลภาษามือเป็นตัวอักษรภาษาไทยโดยใช้ออโตมาตาแบบจำกัด.....

ชื่อภาษาอังกฤษ.....Development of Translation Program from Thai Sign Language to Thai Alphabets.....
Using Finite Automata.....

ปี พ.ศ. (ปีที่จบการศึกษา).....2561.....

เป็นผลงานวิจัยของข้าพเจ้าแต่เพียงผู้เดียว โดยมีได้คัดลอกหรือละเมิดลิขสิทธิ์ของผู้อื่นและได้ผ่านการตรวจสอบความ
 ซ้ำซ้อนโดยใช้โปรแกรมเรียบร้อยแล้ว และได้แนบเอกสารการตรวจสอบการลอกเลียนงานวรรณกรรมที่ตรวจสอบจาก
 วิทยานิพนธ์ฉบับสมบูรณ์แล้ว

โปรแกรมอักขราวิสุทธิ์.....0.00.....% หรือโปรแกรม Turnitin.....%

อ้างถึงมติคณะกรรมการสภาวิชาการ ครั้งที่ 6/2560 เมื่อวันที่ 20 มิถุนายน พ.ศ.2560 เรื่อง“ขอความ
 เห็นชอบการใช้ระบบในการตรวจสอบการคัดลอกผลงาน” ทั้งนี้ ในการตรวจสอบการคัดลอกผลงานวิชาการที่มีเนื้อหา
 คล้ายกัน ต้องไม่เกิน 35 เปอร์เซ็นต์ ของผลงาน และต้องยืนยันการตรวจสอบโดยอาจารย์ที่ปรึกษา

ลงชื่อ.....
 (นางสาวจิริวรรธ ตุ่มศรี)
 นักศึกษา

ข้าพเจ้า ศ. / รศ. / ผศ. / ดร.วรางคณา กัมปาน.....อาจารย์ที่ปรึกษาวิทยานิพนธ์
 ได้ตรวจสอบวิทยานิพนธ์ของ นาย/นาง/นางสาว.....จิริวรรธ ตุ่มศรี.....แล้วขอรับรองว่าเป็นผลงานวิจัย
 ของนักศึกษาจริงและมีเนื้อหาสมบูรณ์ เห็นควรเข้ารับการสอบวิทยานิพนธ์เพื่อขอจบการศึกษาได้

ลงชื่อ.....
 (ผศ.ดร.วรางคณา กัมปาน)
 อาจารย์ที่ปรึกษาวิทยานิพนธ์

Plagiarism Checking Report

Created on Jul 16, 2018 at 14:54 PM

Submission Information

ID	SUBMISSION DATE	SUBMITTED BY	ORGANIZATION	FILENAME	STATUS	SIMILARITY INDEX
905943	Jul 16, 2018 at 14:54 PM	57605074@kmitl.ac.th	สถาบันเทคโนโลยี พระจอมเกล้าเจ้า คุณทหาร ลาดกระบัง	3_รวม.pdf	Completed	0.00 %

Match Overview

NO.	TITLE	AUTHOR(S)	SOURCE	SIMILARITY INDEX
No data available in table				

Match Details

TEXT FROM SUBMITTED DOCUMENT

TEXT FROM SOURCE DOCUMENT(S)

ประวัติผู้เขียน

ชื่อ	นางสาวจิริวรรณ ตุ่มศรี
วัน เดือน ปีเกิด	20 สิงหาคม พ.ศ. 2534
ที่อยู่	136 หมู่ 6 ตำบลท่าแลง อำเภอท่ายาง จังหวัดเพชรบุรี 76130
ประวัติการศึกษา	2557 วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ เกรดเฉลี่ย 3.22 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ 2560 วิทยาศาสตรมหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ เกรดเฉลี่ย 3.68 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ
ผลงานวิชาการ	Jirawat Tumsri, and Warangkhana Kimpan, "Thai Sign Language Translation Using Leap Motion Controller," Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2017, 15-17 March, 2017, Hong Kong, pp. 46-51.