

การพัฒนาระบบเพื่อสร้างแผนผังการเดินทางภายในอาคารจากวิดีโอและข้อมูลการเคลื่อนที่

วัชรพล เอื้อพูนวิริยะ

TNI

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

บัณฑิตวิทยาลัย สาขาเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

ปีการศึกษา 2557

A SYSTEM DEVELOPMENT FOR CREATING INDOOR FLOOR PLAN OF WALKING ROUTE
USING VIDEO AND MOVEMENT DATA

Vacharapol Euapoonviriya

TNI

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Information Technology

Graduate School

Thai-Nichi Institute of Technology

Academic Year 2014

หัวข้อวิทยานิพนธ์

การพัฒนาระบบเพื่อสร้างแผนผังการเดินทางภายในอาคาร
จากวิดีโอ และข้อมูลการเคลื่อนที่

โดย

วัชรพล เอื้อพูนวิริยะ

สาขาวิชา

เทคโนโลยีสารสนเทศ

อาจารย์ที่ปรึกษาวิทยานิพนธ์

ดร. สะพรั่งสิทธิ์ มฤตุสาร

บัณฑิตวิทยาลัย สถาบันเทคโนโลยีไทย-ญี่ปุ่น อนุมัติให้บัณฑิตวิทยาลัยนี้เป็น
ส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

.....คณบดีบัณฑิตวิทยาลัย

(รองศาสตราจารย์ ดร. พิชิต สุขเจริญพงษ์)

วันที่.....เดือน.....พ.ศ.....

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ดร. ศุภกร สิทธิไชย)

.....กรรมการ

(ดร. กรกฎ เหมสถาปัตย์)

.....กรรมการ

(ดร. จูติพร เลิศรัตน์เดชากุล)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์

(ดร. สะพรั่งสิทธิ์ มฤตุสาร)

วัชรพล เอื้อพูนวิริยะ : การพัฒนาระบบเพื่อสร้างแผนผังการเดินทางภายในอาคารจากวิดีโอ
และข้อมูลการเคลื่อนที่. อาจารย์ที่ปรึกษา : ดร. สะพรั่งสิทธิ์ มฤตสาธิต, 117 หน้า.

การพัฒนาระบบเพื่อสร้างแผนผังการเดินทางที่อ้างอิงจากการเดินจริงของผู้ใช้ผ่านทางสมาร์ทโฟน
ระบบปฏิบัติการ Android ร่วมกับการใช้โปรแกรมที่พัฒนาขึ้นเพื่อใช้งานบนเครื่องคอมพิวเตอร์ โดย
ที่สมาร์ทโฟนจะใช้ในการเก็บข้อมูลวิดีโอการเดินทาง รวมถึงการเก็บข้อมูลการเคลื่อนที่จากเซ็นเซอร์ที่
ถูกติดตั้งในสมาร์ทโฟน เช่น Accelerometer หรือ Gyroscope แต่จะไม่รวมถึงการใช้งานข้อมูลจาก
GPS เนื่องจากอาจเกิดการอับสัญญาณเพราะเป็นการใช้งานในอาคาร อีกทั้งยังมีความแม่นยำไม่
เพียงพอเมื่อนำมาใช้งานในอาคารอีกด้วย เมื่อทำการเก็บข้อมูลที่ต้องการเรียบร้อยแล้ว จะทำการส่ง
ต่อข้อมูลไปยังคอมพิวเตอร์และใช้งานโปรแกรมที่พัฒนาขึ้นอีกตัวหนึ่งเพื่อวิเคราะห์ข้อมูลวิดีโอโดย
ประมวลผลจากเส้นขอบที่เกิดขึ้นระหว่างพื้นและกำแพงในการบอกลักษณะพื้นทางเดินและสร้างเป็น
แผนผังการเดินทางได้อย่างสมบูรณ์โดยที่ระยะทางในแผนที่ที่มีอัตราส่วนของระยะทางเทียบเคียงกับการ
เดินจริงโดยอ้างอิงจากข้อมูลการเดินทางที่เก็บมาได้

บัณฑิตวิทยาลัย

สาขาวิชา เทคโนโลยีสารสนเทศ

ปีการศึกษา 2557

ลายมือชื่อนักศึกษา

ลายมือชื่ออาจารย์ที่ปรึกษา

VACHARAPOL EUAPOONVIRIYA : A SYSTEM DEVELOPMENT FOR CREATING
INDOOR FLOOR PLAN OF WALKING ROUTE USING VIDEO AND MOVEMENT
DATA. ADVISOR : DR. SAPRANGSIT MRUETUSATORN, 117 PP.

System development for creating indoor floor plans based on the actual walking route of a person using an Android application on a smartphone working with a PC-based application on a PC has been proposed. Via a smartphone, the Android application is used for capturing indoor video and also obtaining various movement data from many types of mobile sensors e.g. accelerometer, gyroscope, etc., however, except for GPS due to the possibility of signal blocking and insufficient accuracy for the indoor purpose. Then all the collected data are sent to the PC to analyze and process using the PC-Based application to detect the edge between the floor and the wall from video and merge into a consolidated floor plan. In addition, the collected movement data is calculated and estimate the actual walking distance used for adjusting scale to correspond with the actual floor plan, which the complete floor plan of actual walking route is finally generated.

TNI

Graduate School

Student's Signature.....

Field of Study Information Technology

Advisor's Signature.....

Academic Year 2014

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลงได้ด้วยความกรุณาของอาจารย์ ดร. สะพรั่งสิทธิ์ มฤตุสาธร อาจารย์ที่ปรึกษาที่กรุณาให้คำปรึกษาสำหรับทุกช่วงของการวิจัยรวมถึงการให้คำวิจารณ์ต่างๆ ที่ทำให้ผลงานเสร็จสมบูรณ์ลงได้ผู้วิจัยขอกราบขอบพระคุณด้วยความซาบซึ้งเป็นอย่างสูง

ผู้วิจัยขอกราบขอบพระคุณคณะกรรมการสอบทุกท่าน ได้แก่ ดร. ศุภกร สิทธิไชย ผู้เป็นประธานกรรมการในการสอบ ดร. กรกฏ เหมสถาปัติย์ และ ดร. จูติพร เลิศรัตน์เดชากุล ที่กรุณาตรวจวิทยานิพนธ์ฉบับนี้ให้สมบูรณ์

ผู้วิจัยขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ที่เป็นผู้เบิกทางให้กับการศึกษารวมถึงการทำงานวิจัยชิ้นนี้ ขอบขอบคุณเพื่อนๆ พี่ๆ น้อง รวมคณะที่ฟันฝ่าร่วมแรงร่วมใจเล่าเรียนและให้คำแนะนำต่างๆ จนทำให้มีโอกาสทำให้วิจัยชิ้นนี้ให้ลุล่วงได้

สุดท้ายนี้ผู้วิจัยหวังว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์ในการนำไปใช้ในอนาคตต่อไปขอขอบคุณอีกครั้ง

วัชรพล เอื้อพูนวิริยะ

TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ	ฉ
สารบัญ	ช
สารบัญตาราง	ณ
สารบัญรูป	ญ
บทที่	
1 บทนำ	1
สภาวะความเป็นมา แนวทางเหตุผล และปัญหา	1
ความมุ่งหมายของการวิจัย	3
ขอบเขตของการวิจัย	3
ประโยชน์ที่คาดว่าจะได้รับ	4
นิยามศัพท์เฉพาะ	4
2 หลักการพื้นฐาน เอกสาร และงานวิจัยที่เกี่ยวข้อง	6
หลักการพื้นฐาน	6
3 ขั้นตอนการดำเนินงาน	19
ภาพรวมการดำเนินงาน	19
การรวบรวมความต้องการของระบบ	20
การวิเคราะห์ ความต้องการของระบบ	20
การออกแบบและพัฒนาระบบ	22
การพัฒนาระบบบนคอมพิวเตอร์ด้วยเครื่องมือ MATLAB	27
4 ผลการทดลอง	36
การเก็บข้อมูลและทดสอบสมมุติฐานต่างๆ ในการเดิน	36
การเก็บข้อมูลเพื่อนำมาใช้ในการทดลอง	37

สารบัญ (ต่อ)

บทที่	หน้า
4	การบันทึกผลการทดลอง 58
5	การวิเคราะห์ผลการทดลอง สรุปผลงานวิจัย และข้อเสนอแนะ 61
	วิเคราะห์ผลการทดลอง และสรุปผลงานวิจัย 61
	ข้อเสนอแนะ 63
บรรณานุกรม	 65
ภาคผนวก	 69
	ภาคผนวก ก. ซอร์สโค้ดโปรแกรมบนสมาร์ทโฟน Android 70
	ภาคผนวก ข. ซอร์สโค้ดโปรแกรมในเครื่องมือ MATLAB..... 91
	ภาคผนวก ค. ภาพผลลัพธ์แผนผังการเดิน 109
ประวัติย่อผู้วิจัย	 117

TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

สารบัญตาราง

ตาราง	หน้า
1.1 แผนงานและระยะเวลาในการดำเนินงาน.....	5
4.1 ระยะและจำนวนการก้าวในการทดสอบ.....	36
4.2 จำนวนก้าวจริงและจำนวนก้าวที่ได้จากระบบ.....	37
4.3 ผลการทดลองจากการเดินในเส้นทาง 10 เส้นทางใน 4 สถานที่.....	59



TNI

สารบัญรูป

รูป	หน้า
2.1	แผนการหมุนพร้อมสัญลักษณ์ใน Android..... 8
2.2	แผนภาพการทำงานในงานวิจัยของ Yinxiao และ Stanley 9
2.3	ผลลัพธ์งานวิจัยของ Yinxiao และ Stanley 10
2.4	เส้นตรงที่ได้จากการใช้กระบวนการ Hough Transform 10
2.5	กระบวนการเชื่อมจุดหรือเส้นเข้าด้วยกันเพื่อแยกพื้นทางเดินออกจากส่วนอื่นของภาพ 11
2.6	ผลลัพธ์ของพื้นที่ได้จากวิธีการของ Gillium 12
2.7	ความเร่งที่นำมาพล็อตกราฟเพื่อหารูปแบบการเดินและนับจำนวนก้าว 13
2.8	กราฟที่ได้จาก Accelerometer ในการเดินรูปแบบต่างๆ..... 13
2.9	การวิเคราะห์ช่วงระหว่างการเดิน 14
2.10	(ก) ผลลัพธ์การถ่ายภาพโดยถ่ายหมุนรอบตัวเอง (ข) ภาพการเคลื่อนที่ไปยังเก็บ ข้อมูลภาพห้องถัดไป 15
2.11	ผลลัพธ์ของการสร้างแผนผังบ้าน 16
2.12	รูปเปรียบเทียบระหว่างการหาเส้นขอบด้วยอนุพันธ์อันดับหนึ่งและอนุพันธ์อันดับสอง 17
2.13	รูปเส้นตรงในระบบพิกัด Cartesian และระบบพิกัด Polar 18
2.14	กราฟที่นำค่า x และ y ในพิกัดระบบ Cartesian พล็อตกราฟในพิกัดระบบ Polar..... 18
3.1	ภาพรวมการทำงานของระบบสร้างแผนผังการเดิน 22
3.2	รูปหน้าจอแสดงผลของโปรแกรมบนสมาร์ตโฟน 23
3.3	ขั้นตอนการทำงานของโปรแกรมบนสมาร์ตโฟน 25
3.4	แผนผังการประมวลผล MATLAB..... 27
3.5	ภาพที่ได้จากวิดีโอทุก 1 วินาที ของการเดิน..... 28
3.6	ขั้นตอนการวิเคราะห์รูปแบบพื้นทางเดิน..... 29
3.7	รูปต้นฉบับ 29
3.8	การแปลงรูปให้เป็นรูป Grayscale 29
3.9	การใช้ Canny Edge Detector เพื่อหาเส้นขอบจากภาพ Gray Scale..... 30
3.10	ผลลัพธ์หลังการประมวลผลเพื่อหาเส้นตรงจากภาพ 30
3.11	ตัวแปรรูปแบบทางเดินในลักษณะของ Bit 31
3.12	ภาพของชิ้นส่วนแบบ Top View ที่ใช้ในการสร้างแผนผังการเดิน..... 32

สารบัญรูป (ต่อ)

รูป	หน้า
3.13 Flow Chart การสร้างแผนผังการเดิน	34
3.14 ก) ถึง ญ) รูปแผนผังการเดินที่ทำการเพิ่มเติมไปในแต่ละช่วงวินาทีของการเดิน	34
3.15 ผลลัพธ์สุดท้ายของการสร้างแผนผังการเดิน	35
4.1 รูปแบบการเดินที่ใช้ในการทดลอง (ก) เส้นทางที่ 1 (ข) เส้นทางที่ 2 (ค) เส้นทางที่ 3 (ง) เส้นทางที่ 4.....	37
4.2 การเดินที่ Sino Thai ชั้น 10	40
4.3 การเดินที่ LPN อาคาร A ชั้น 9	41
4.4 การเดินที่สถาบันเทคโนโลยีไทย – ญี่ปุ่น อาคาร A ชั้น 6	42
4.5 การเดินที่สถาบันเทคโนโลยีไทย – ญี่ปุ่น อาคาร A ชั้น 6	43
4.6 ผลลัพธ์ Grayscale ของการเดินที่ Sino Thai ชั้น 10	44
4.7 ผลลัพธ์ Grayscale ของการเดินที่ LPN อาคาร A ชั้น 9	45
4.8 ผลลัพธ์ Grayscale ของการเดินที่สถาบันเทคโนโลยีไทย – ญี่ปุ่น อาคาร A ชั้น 6	46
4.9 ผลลัพธ์ Grayscale ของการเดินที่สถาบันเทคโนโลยีไทย – ญี่ปุ่น อาคาร A ชั้น 5	47
4.10 ผลลัพธ์ Canny Edge Detection จากการเดินที่ Sino Thai ชั้น 10	48
4.11 ผลลัพธ์ Canny Edge Detection จากการเดินที่ LPN อาคาร A ชั้น 9	49
4.12 ผลลัพธ์ Canny Edge Detection จากการเดินที่สถาบันเทคโนโลยีไทย – ญี่ปุ่น อาคาร A ชั้น 6.....	50
4.13 ผลลัพธ์ Canny Edge Detection จากการเดินที่สถาบันเทคโนโลยีไทย – ญี่ปุ่น อาคาร A ชั้น 5.....	51
4.14 ผลลัพธ์ Hough Transform จากการเดินที่ Sino Thai ชั้น 10	52
4.15 ผลลัพธ์ Hough Transform จากการเดินที่ LPN อาคาร A ชั้น 9	53
4.16 ผลลัพธ์ Hough Transform จากการเดินที่สถาบันเทคโนโลยีไทย – ญี่ปุ่น อาคาร A ชั้น 6.....	54
4.17 ผลลัพธ์ Hough Transform จากการเดินที่สถาบันเทคโนโลยีไทย – ญี่ปุ่น อาคาร A ชั้น 5.....	55
4.18 ข้อมูล Pattern ที่ได้จากการวิเคราะห์เส้นตรงภาพซ้ายที่อาคาร Sino Thai และภาพขวาอาคาร LPN.....	56

สารบัญรูป (ต่อ)

รูป	หน้า
4.19 ข้อมูล Pattern ที่ได้จากการวิเคราะห์เส้นตรงภาพซ้ายที่อาคาร TNI ชั้น 6 และภาพขวาที่อาคาร TNI ชั้น 5	56
4.20 ข้อมูล Pattern ที่ได้จากการวิเคราะห์เส้นตรงที่ (ก) อาคาร Sino Thai และ (ข) อาคาร LPN ไคแรคทอรีที่ X0005	57
4.21 ข้อมูล Pattern ที่ได้จากการวิเคราะห์เส้นตรงที่อาคาร TNI (ก) ชั้น 6 และ (ข) ชั้น 5	57
4.22 ผลลัพธ์สุดท้ายของแผนผังการเดินทาง	58
ก.1 โครงสร้าง Code ของระบบ	71
ค.1 แผนผังการเดินทางที่ 1 เลี้ยวซ้าย	110
ค.2 แผนผังการเดินทางที่ 1 เลี้ยวขวา	110
ค.3 แผนผังการเดินทางที่ 2 เลี้ยวซ้าย	110
ค.4 แผนผังการเดินทางที่ 2 เลี้ยวขวา	110
ค.5 แผนผังการเดินทางที่ 3 เดินตรง	111
ค.6 แผนผังการเดินทางที่ 1 เลี้ยวซ้าย	112
ค.7 แผนผังการเดินทางที่ 1 เลี้ยวขวา	112
ค.8 แผนผังการเดินทางที่ 2 เลี้ยวซ้าย	112
ค.9 แผนผังการเดินทางที่ 2 เลี้ยวขวา	112
ค.10 แผนผังการเดินทางที่ 3 เดินตรง	113
ค.11 แผนผังการเดินทางที่ 1 เลี้ยวซ้าย	114
ค.12 แผนผังการเดินทางที่ 1 เลี้ยวขวา	114
ค.13 แผนผังการเดินทางที่ 2 เลี้ยวซ้าย	114
ค.14 แผนผังการเดินทางที่ 2 เลี้ยวขวา	114
ค.15 แผนผังการเดินทางที่ 3 เดินตรง	115
ค.16 แผนผังการเดินทางที่ 4	116

บทที่ 1

บทนำ

สภาวะความเป็นมา แนวทางเหตุผล และปัญหา

การเดินทางเป็นสิ่งที่มีความสำคัญและขาดไม่ได้สำหรับสิ่งมีชีวิตการเดินทางที่เกิดขึ้นนั้นมีด้วยกันหลายเหตุผล เช่น เพื่อติดต่อสื่อสารระหว่างกันท่องเที่ยว เป็นต้น การเดินทางในสถานที่ที่คุ้นเคยหรือรู้จักมักไม่เกิดปัญหาใดๆ เนื่องจากมีความคุ้นชินกับสถานที่เหล่านั้นเป็นอย่างดีแต่ถ้าเป็นการเดินทางไปยังสถานที่ที่ไม่คุ้นเคยหรือจำเส้นทางไม่ได้ อาจเกิดการหลงทางได้ซึ่งนอกจากจะทำให้เสียเวลา ในบางครั้งอาจจะทำให้เดินทางไปไม่ถึงยังจุดหมายและอาจเกิดผลกระทบต่างๆ ได้ สำหรับบุคคลทั่วไปตัวช่วยสำคัญที่ทำให้การเดินทางราบรื่นและถึงจุดหมายได้แม้ในสถานที่ที่ไม่คุ้นเคยนั้นคือแผนที่ โดยที่แผนที่นั้นมีประโยชน์ในด้านการทำให้ผู้ที่เดินทางสามารถทำความเข้าใจในเส้นทางการระบุบริเวณที่อยู่ปัจจุบัน หรือตำแหน่งของเป้าหมายที่ต้องการไป และเป็นสื่อกลางที่ทำให้ผู้เดินทางนำไปใช้ในการค้นหาเส้นทางในการเดินทางได้ซึ่งแผนที่นั้นมีด้วยกันหลายรูปแบบในช่วงแรกแผนที่จะอยู่ในรูปแบบของกระดาษขนาดต่างๆ หรือแผ่นป้ายที่ติดอยู่ตามสถานที่ต่างๆ แต่การใช้กระดาษ หรือแผ่นป้ายขนาดต่างๆ นั้นยังมีข้อจำกัดในด้านขนาดของแผนที่ซึ่งถ้าแผนที่ที่มีขนาดใหญ่จำเป็นจะต้องใช้กระดาษหรือแผ่นป้ายที่มีขนาดใหญ่ขึ้นตามเช่นกัน ทำให้ลำบากต่อการใช้งานหรือการพกพาเนื่องจากในบางครั้งผู้เดินทางมีความต้องการที่จะนำแผนที่เหล่านั้นติดตัวไปด้วยและการทำแผนที่จะต้องใช้คนเพื่อร่างสถานที่และเส้นทางตามความเข้าใจของผู้เขียนซึ่งมักมีความแม่นยำไม่สูงนักแต่ยังสามารถใช้ในการอธิบายการเดินทางได้ในภายหลังจึงมีผู้จัดทำข้อมูลแผนที่ให้อยู่ในรูปแบบของข้อมูลดิจิทัลแทนซึ่งเพิ่มความสะดวกเพราะสามารถย่อหรือขยายความละเอียดของแผนที่ได้โดยที่ขนาดของอุปกรณ์ที่ใช้ในการแสดงแผนที่มีขนาดเดิม วิธีการพื้นฐานในการสร้างแผนที่นั้นคือใช้ภาพถ่ายดาวเทียม GPS ถ่ายภาพพื้นที่ที่ต้องการทำแผนที่และทำการจำแนกสิ่งต่างๆ จากภาพเช่น อาคาร ถนน นำมาสร้างเป็นแผนที่ [1]

ทั้งหมดที่กล่าวมาข้างต้นเป็นการใช้งานแผนที่ในที่โล่งแจ้งหรือด้านนอกอาคาร สำหรับแผนที่ในอาคารโดยมากอยู่ในรูปแบบของบอร์ดหรือป้ายที่ติดตามส่วนต่างๆ ในอาคารโดยเจ้าของอาคารเป็นผู้มอบหมายให้มีการสร้างแผนที่หรือแผนผังทางเดินขึ้นเพื่อให้เพื่อช่วยเหลือผู้ที่เข้าใช้งานอาคารเพราะบางอาคารเป็นอาคารขนาดใหญ่และมีความซับซ้อนเช่น ห้างสรรพสินค้า โรงพยาบาล เป็นต้น บางครั้งเจ้าของอาคารจะนำแผนผังทางเดินเผยแพร่ไว้ตามพื้นที่สาธารณะ เช่น เว็บไซต์ของตนเองเพื่ออำนวยความสะดวกแก่ผู้เข้าใช้งานอาคาร

แต่ไม่ว่าจะเป็นการเดินทางกลางแจ้งหรือในอาคารการใช้งานแผนที่หรือแผนผังอาจเกิดความสับสนกับแผนที่หรือแผนผังได้เพราะต้องใช้เวลาทำความเข้าใจแผนที่และทิศทางเพราะผู้อ่านแผนที่ต้องทำความเข้าใจแผนผังและเปรียบเทียบกับสถานที่จริง ดังนั้นแล้วจึงมีผู้คิดค้นระบบนำทางหรือ Navigation System ขึ้นใช้งานร่วมกับแผนที่หรือแผนผังเพื่อช่วยในอำนวยความสะดวกในการเดินทางและง่ายต่อการใช้งาน การเดินทางกลางแจ้งหรือนอกอาคารนั้น ระบบจะให้ผู้ใช้ทำการระบุปลายทางที่ต้องการไปและให้ระบบทำการประมวลผลเพื่อเลือกเส้นทางและนำทางผู้ใช้ไปยังปลายทางได้โดยเริ่มต้นจากบริเวณที่อยู่ปัจจุบันซึ่งอ้างอิงข้อมูลตำแหน่งจาก GPS ระบบดังกล่าวมีหลากหลายเช่น Google Maps [2] แต่สำหรับการเดินทางภายในอาคารนั้นจะมีข้อจำกัดจากการใช้งาน GPS เพราะในบางอาคารนั้นสัญญาณ GPS อ่อนหรือไม่สามารถจับสัญญาณได้เพราะสัญญาณ GPS ถูกกีดขวางจากผนังอาคาร จึงมีนักวิจัยหลายกลุ่มเสนอการระบุตำแหน่งในอาคารและระบบนำทางในอาคาร เช่น การทำระบบช่วยเหลือบุคคลในสถานที่อันตรายโดยมีผู้ชำนาญเป็นผู้ประเมินแผนผังและนำทางให้กับผู้ประสบภัย [3] ระบบนำทางสำหรับผู้ที่มีความบกพร่องทางสายตา[4], [5], [6] โดยใช้เซ็นเซอร์ต่างๆ เป็นส่วนช่วยในการนำทาง ระบบที่นำทางให้กับบุคคลทั่วไปโดยนำไปเปรียบเทียบกับจุดเด่นต่างๆ ที่เก็บเป็นฐานข้อมูลในระบบ [7] หรือแม้แต่ระบบที่ใช้ในการระบุตำแหน่งภายในอาคารโดยใช้ภาพถ่าย [8], [9]

แต่ในงานวิจัยส่วนใหญ่จำเป็นต้องใช้อุปกรณ์หรือเซ็นเซอร์ เข้ามาประกอบในการนำทางหรือสร้างแผนผังทางเดิน เช่น การใช้ความแรงของสัญญาณ Wifi Ultrasound RFID GPS และอื่นๆ เป็นตัวช่วยในการจับตำแหน่งของผู้ใช้ซึ่งจำเป็นต้องมีการติดตั้งอุปกรณ์เพิ่มเติมภายในอาคารหรือตัวผู้ใช้เพื่อให้ระบบทำงานได้ ซึ่งอาจมีความยุ่งยากในการจัดเตรียมอุปกรณ์ต่างๆ รวมถึงราคาของอุปกรณ์บางชิ้นมีราคาแพงอีกด้วย

ดังนั้นผู้วิจัยมีแนวความคิดเลียนแบบธรรมชาติในการเดินทางและจำเส้นทางของมนุษย์โดยใช้อุปกรณ์จำนวนน้อยและใกล้ตัวผู้ใช้ที่สุดเท่าที่จะเป็นไปได้ โดยการเดินของมนุษย์นั้นจะจำลักษณะของทางเดินและระยะทางคร่าวๆ และเมื่อเดินจนจำได้แล้ว ก็จะทำให้เกิดความเคยชินและไม่หลงทางอีกต่อไปซึ่งถ้าทำการเปรียบเทียบสถานการณ์ที่เกิดขึ้นเมื่อต้องไปเดินในสถานที่ต่างๆ คือ ตาถูกใช้ในการเก็บข้อมูลภาพในการเดิน ว่ามีทางแยกแบบใดและที่ใดบ้างในระหว่างการเดินพร้อมทั้งการจำทิศทางการเดินว่ามีเดินเลี้ยวที่แยกใดบ้าง รวมถึงระยะทางการเดินว่าเดินเส้นทางช่วงใดใกล้หรือไกลเพียงใดซึ่งมักจะเป็นการประมาณการด้วยสายตาและจำนวนก้าวโดยประมาณในการเดิน ซึ่งถ้านำไปเปรียบเทียบกับอุปกรณ์อิเล็กทรอนิกส์แล้ว จะมีอุปกรณ์ที่สามารถทำงานดังกล่าวได้คือ กล้องถ่ายรูปเพื่อใช้ในการเก็บภาพการเดินทั้งหมด Gyroscope ที่สามารถตรวจจับการหมุนหรือการเลี้ยวได้ และ Accelerometer ที่ใช้ในการตรวจจับการเคลื่อนที่ที่เกิดขึ้นและเซ็นเซอร์ทั้ง 3 ชนิดนั้นอยู่ภายในสมาร์ทโฟนในปัจจุบัน

ดังนั้นจุดมุ่งหมายในของงานวิจัยในครั้งนี้คือการพัฒนาระบบที่สามารถทำการเก็บข้อมูล การเดินภายในอาคารและนำข้อมูลเหล่านั้นไปประมวลผลเพื่อทำการสร้างแผนผังการเดินในอาคารได้

ความมุ่งหมายของการวิจัย

พัฒนาระบบให้กับบุคคลทั่วไปเพื่อใช้ในการเก็บข้อมูลการเดินและสร้างแผนผังการเดินใน อาคารที่เป็นทางเดินตรงหรือทางเลี้ยวในแนวระนาบได้

ขอบเขตของการวิจัย

1. การใช้งานระบบเพื่อสร้างแผนผังการเดินภายในอาคารลักษณะทางเดิน
 - 1.1 เป็นทางเดินที่มีกำแพงทั้ง 2 ด้าน
 - 1.2 ทางเดินที่ไม่ใช่ลักษณะห้องโถงหรือห้องในอาคาร
 - 1.3 เป็นทางเดินในแนวระนาบ
 - 1.4 เป็นทางเดินตรง ไม่เป็นทางโค้ง
 - 1.5 เป็นทางเลี้ยวซ้ายหรือขวาหรือสามแยกหรือสี่แยกที่ตั้งมุมฉากกับทางเดินตรง
 - 1.6 ลายพื้นเป็นสีเดียวกันหรือพื้นที่ไม่เป็นรูปแบบตายตัว
 - 1.7 รอยต่อระหว่างพื้นและกำแพงโดยรวมเห็นได้อย่างชัดเจน
2. การเก็บข้อมูลการเดิน จัดเก็บข้อมูล 3 ชนิด
 - 2.1 ข้อมูลจากวิดีโอเพื่อเก็บรูปแบบของทางเดิน
 - 2.2 ข้อมูลจาก Gyroscope เพื่อเก็บข้อมูลการหมุนตัว
 - 2.3 ข้อมูลจาก Step Detector เพื่อเก็บข้อมูลการก้าว
3. ปัจจัยที่ไม่ครอบคลุมในการประมวลผลเพื่อสร้างแผนผังการเดิน
 - 3.1 ความแตกต่างด้านความสูงของการถ่ายภาพวิดีโอ
 - 3.2 ความสว่างของแสงที่ไม่เพียงพอต่อการประมวลผล
 - 3.3 ระยะก้าวที่ไม่สม่ำเสมออย่างมีนัยสำคัญ
 - 3.4 ความเร็วที่ใช้ในการเดินเป็นความเร็วในการเดินปกติ
 - 3.5 ภาพที่ไม่คมชัดที่เกิดจากการเคลื่อนที่
4. สิ่งที่ยอยู่นอกเหนือขอบเขตงานวิจัย
 - 4.1 การปรับแต่งค่าการตรวจจับการก้าวของผู้ใช้โดยสมาร์ทโฟน
 - 4.2 ความแตกต่างของสภาพแวดล้อมและผู้เดิน เช่น ความเหนื่อยล้าก่อนทำการเดิน ร้องเท้าที่ใส่ขณะทำการเดิน การแต่งกาย

ประโยชน์ที่คาดว่าจะได้รับ

1. ระบบสามารถนำไปใช้ในการเก็บข้อมูลการเดินทางและสร้างแผนผังการเดินทางได้
2. ข้อมูลที่เก็บในระบบสามารถใช้งานต่อเนื่องกับระบบอื่นๆ ได้ เช่น ระบบนำทางในอาคาร ระบบค้นหาสถานที่ในอาคาร เป็นต้น
3. เพิ่มความสะดวกและลดเวลาในการสร้างแผนผังการเดินทางได้

นิยามศัพท์เฉพาะ

1. **ทางเดินในอาคาร** คือ ส่วนของทางเดินในอาคารซึ่งเป็นทางเชื่อมต่อระหว่างห้องภายในอาคารเดียวกัน
2. **GPS (Global Positioning System)** คือ ระบบระบุพิกัดบนพื้นโลกจากดาวเทียมส่งข้อมูลเป็นเส้นตรงและต้องไม่มีสิ่งกีดขวาง มีวัตถุประสงค์หลายอย่างเช่น เพื่อใช้ทางการทหาร เพื่อการพาณิชย์ หรือเพื่อวัตถุประสงค์ทั่วไป
3. **RFID (Radio Frequency Identification)** คือ อุปกรณ์ไร้สายที่ใช้คลื่นสนามแม่เหล็กในการส่งข้อมูลมักใช้ในการระบุข้อมูลของวัตถุหรือสินค้าที่ติดแท็กของ RFID เข้าไป
4. **สมาร์ทโฟน (Smart Phone)** คือ โทรศัพท์มือถือที่มาพร้อมกับระบบปฏิบัติการที่มีความสามารถเพิ่มเติมนอกจากการโทรศัพท์ สามารถติดตั้งแอปพลิเคชันเพิ่มเติมได้ และมีอุปกรณ์หรือเซ็นเซอร์อื่นเชื่อมต่อมาด้วยตัวอย่างเช่น กล้องถ่ายรูป อุปกรณ์เล่นเพลง ระบบ GPS รวมถึงการเชื่อมต่อกับ Internet ฯลฯ
5. **Android** คือ ระบบปฏิบัติการบนสมาร์ทโฟนที่มีพื้นฐานมาจากระบบปฏิบัติการ Linux พัฒนาขึ้นโดยบริษัท Google
6. **API (Application Program Interface)** คือ ชุดคำสั่งเพื่อการโปรแกรมมิ่งเป็นช่องทางในการติดต่อกันระหว่างโปรแกรมกับระบบปฏิบัติการ หรือ โปรแกรมกับโปรแกรมด้วยกัน ประโยชน์ของ API คือ การทำให้ผู้พัฒนาสามารถใช้งานระบบปฏิบัติการหรือโปรแกรมที่ผู้เขียนเปิดช่องทางไว้ได้อย่างมีประสิทธิภาพและไม่ต้องเข้าใจวิธีการทำงานภายในทำให้การโปรแกรมมิ่งสะดวกและรวดเร็วมากขึ้น
7. **Accelerometer** คือ อุปกรณ์ที่ใช้ในการตรวจจับการความเร่งในการเคลื่อนที่ เป็นอุปกรณ์ต่อพ่วงพื้นฐานชิ้นหนึ่งในสมาร์ทโฟน
8. **Gyroscope** คือ อุปกรณ์ที่ใช้ในการตรวจจับองศาในการหมุน เป็นอุปกรณ์ต่อพ่วงพื้นฐานชิ้นหนึ่งในสมาร์ทโฟน

บทที่ 2

หลักการพื้นฐาน เอกสาร และงานวิจัยที่เกี่ยวข้อง

หลักการพื้นฐาน

งานวิจัยนี้ ผู้วิจัยได้ศึกษาหลักการพื้นฐาน เอกสารงานวิจัยที่เกี่ยวข้อง และได้นำเสนอตามหัวข้อต่อไปนี้

1. การเก็บข้อมูลจากเซ็นเซอร์ที่ต่อเชื่อมกับสมาร์ทโฟน
2. การระบุพื้นที่ทางเดินจากภาพถ่าย
3. การคำนวณระยะทางการเดิน
4. การสร้างแผนผังในอาคาร
5. ทฤษฎี Image Processing ที่เกี่ยวข้อง

1. การเก็บข้อมูลจากเซ็นเซอร์ที่ต่อเชื่อมกับสมาร์ทโฟน

สมาร์ทโฟนในปัจจุบันมีการติดตั้งเซ็นเซอร์ชนิดต่างๆ เพื่อประโยชน์ในการใช้งานหลากหลายรูปแบบ เช่น การนำกล้องถ่ายรูปที่มีคุณภาพสูงมาเพื่อการถ่ายรูปและถ่ายวิดีโอที่มีภาพคมชัด GPS เพื่อประโยชน์ในการค้นหาตำแหน่งปัจจุบันของผู้ใช้งานสมาร์ทโฟนมักนำไปใช้ร่วมกับบริการด้านแผนที่เช่น Google Maps หรือแอปพลิเคชันแผนที่อื่นๆ Accelerometer และ Gyroscope เพื่อใช้ในการจับการเคลื่อนไหวในทิศทางต่างๆ มักถูกใช้ในหลากหลายรูปแบบเช่น การจับพฤติกรรมขณะใช้สมาร์ทโฟน เกม หรือ ใช้ในการนับจำนวนก้าวเดิน เป็นต้น

เซ็นเซอร์เหล่านี้นอกจากจะถูกติดตั้งเข้ากับสมาร์ทโฟนเพื่อการใช้งานกับระบบปฏิบัติการหรือแอปพลิเคชันของผู้ผลิตเอง ผู้ผลิตระบบปฏิบัติการยังได้เปิดช่องทางในการเรียกใช้งานเซ็นเซอร์เหล่านี้ให้สำหรับผู้พัฒนาแอปพลิเคชันรายอื่นเรียกใช้งานอีกด้วย สำหรับระบบปฏิบัติการ Android นั้นมีจำนวน API หรือ Application Program Interface มากมาย [10] โดย API ที่ใช้ในงานวิจัยนี้คือ กล้องถ่ายรูป และเซ็นเซอร์ที่เกี่ยวข้องกับการเคลื่อนไหว (Motion Sensor)

สำหรับ API ของกล้องจะใช้ในการเก็บข้อมูลการเดินที่เป็นภาพต่อเนื่องหรือวิดีโอเพื่อใช้เป็นข้อมูลในการประมวลรูปแบบของพื้นที่ทางเดินและนำไปใช้ในการสร้างแผนผังการเดินโดยมี Class ที่ใช้งานดังนี้

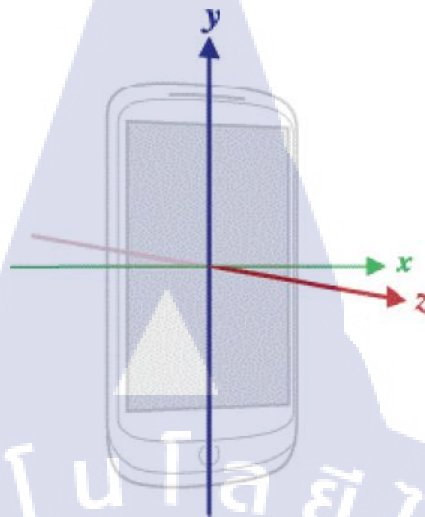
1. `Android.Hardware.Camera` จะเป็น Class หลักที่ใช้ในการสั่งงานกล้องของสมาร์ทโฟนให้เริ่มทำการบันทึกหรือหยุดการบันทึกวิดีโอ

2. `Android.View.SurfaceHolder` และ `Android.View.SurfaceView` ใช้ในการแสดงภาพขณะบันทึกวิดีโอบนหน้าจอของสมาร์ทโฟน

API ของ Motion Sensors นั้นจะเน้นที่การเก็บข้อมูลประกอบที่เกิดขึ้นขณะเดินเช่น การตรวจนับจำนวนก้าวในการเดิน หรือ การตรวจจับการเลี้ยวเกิดขึ้นในช่วงใดบ้างโดยในการพัฒนาแอปพลิเคชันนั้นจะต้องมีประกาศ Implement Listener ที่ชื่อว่า `Android.Hardware.Sensor Event Listener` เพื่อให้แอปพลิเคชันสามารถรับข้อมูลจากเซ็นเซอร์ตรวจจับการเคลื่อนไหวได้ ซึ่งค่าที่เซ็นเซอร์ส่งคืนกลับมาให้นั้นจะกลับมาที่เมธอด `On Sensor Changed` ซึ่งภายในเมธอดนั้นจะทำการคัดแยกเฉพาะข้อมูลจากเซ็นเซอร์ที่ต้องการเท่านั้น ซึ่งในที่นี้จะป็นเซ็นเซอร์ 2 ชนิด คือ `Sensor.TYPE_STEP_DETECTOR` เพื่อใช้ในการตรวจจับการก้าวว่าเกิดการก้าวในช่วงเวลาใดบ้าง และ `Sensor.TYPE_GYROSCOPE` เพื่อใช้ในการตรวจจับการเลี้ยวที่เกิดขึ้นในระหว่างการเดินในช่วงเวลาใดบ้าง

Step Detector เป็นเซ็นเซอร์ใหม่ที่เริ่มใช้งานได้ตั้งแต่ Android OS Version 4.4 [11] ขึ้นไป โดยที่เซ็นเซอร์นี้เป็นการทำงานร่วมกันของเซ็นเซอร์ภายในหลายเซ็นเซอร์ประกอบกันในรูปแบบของอุปกรณ์ที่กินพลังงานต่ำ ทำการประมวลผลโดยนำค่าที่ได้จาก Accelerometer ทำการวิเคราะห์เพื่อหาผลการก้าวที่เกิดขึ้นโดยเปิดการทำงานของเซ็นเซอร์ Step Detector และรอค่าที่ส่งกลับมาจากเซ็นเซอร์เท่านั้น ระบบเป็นผู้ทำการวิเคราะห์และประมวลผลการก้าวของผู้ใช้แทน

ค่าที่ Step Detector ส่งกลับมานั้นจะเป็นค่า 1.0 เท่านั้นเพื่อเป็นการบอกให้กับผู้เรียกใช้ Listener ว่ามีการก้าวเกิดขึ้นแล้ว ส่วนค่าที่ Gyroscope ส่งกลับมานั้นจะเป็นค่าในหน่วยของ Radian ต่อวินาทีซึ่งเป็นตัวเลขที่ระบุค่าการหมุนที่เกิดขึ้นโดยที่ตัวเลขนั้นจะถูกส่งออกมาเป็น 3 ค่าด้วยกันโดยที่แต่ละตัวนั้นจะเป็นค่าการเกิดการหมุนของแกน X Y และ Z ตามรูปที่ 2.1 สำหรับงานวิจัยนี้สนใจการหมุนเฉพาะในแกน X ซึ่งเป็นแกนที่เกิดจากการหมุนไปทางซ้ายหรือทางขวาเท่านั้น สำหรับค่าที่เกิดขึ้นได้สำหรับค่าที่ได้จาก Gyroscope นั้นจะเป็นได้ทั้งค่าบวกและค่าลบ เป็นค่าบวกเมื่อเกิดการเลี้ยวไปทางซ้ายหรือทิศทางทวนเข็มนาฬิกา และเป็นค่าลบเมื่อเกิดการเลี้ยวไปทางขวาหรือทิศทางตามเข็มนาฬิกา

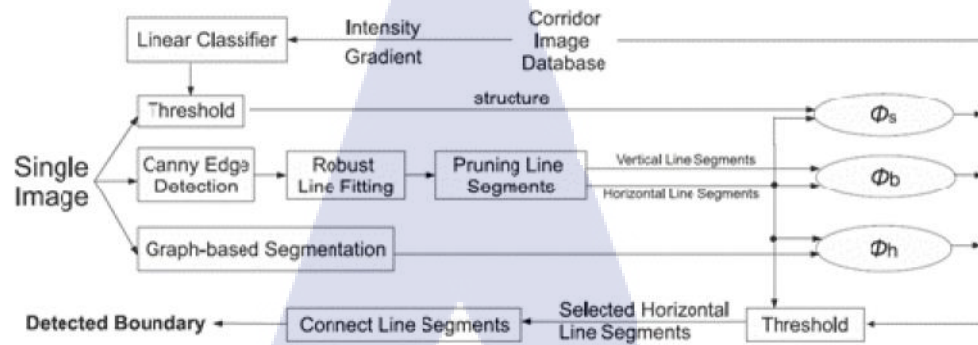


รูปที่ 2.1 แกนการหมุนพร้อมสัญลักษณ์ใน Android [12]

2. การระบุพื้นทางเดินจากภาพถ่าย

การระบุพื้นทางเดินเป็นพื้นฐานสำคัญสำหรับระบบนำทางเป็นอย่างมาก เพราะใช้ในการระบุพื้นที่ที่สามารถเดินได้ให้กับผู้ใช้หรือระบบอัตโนมัติเช่นหุ่นยนต์

สำหรับการระบุพื้นทางเดินจากภาพถ่ายนั้นมีด้วยกันหลายงานวิจัย งานวิจัยของ Yinzhao Li; and Stanley T. Birchfield [13] ที่เสนอวิธีการในการระบุเส้นขอบระหว่างพื้นและองค์ประกอบอื่นออกจากกันจากภาพถ่ายหนึ่งภาพ โดยเริ่มต้นที่การหาเส้นขอบทั้งหมดด้วย Canny Edge Detector และทำการแยกเส้นที่เป็นเส้นในแนวแกนนอนกับเส้นในแนวแกนตั้งออกจากกัน และทำการคัดเส้นที่สั้นออกจากระบบ เพื่อคัดแยกส่วนที่ไม่ใช่เส้นขอบระหว่างพื้นกับกำแพงออกไปตามการทำงานใน รูปที่ 2.2



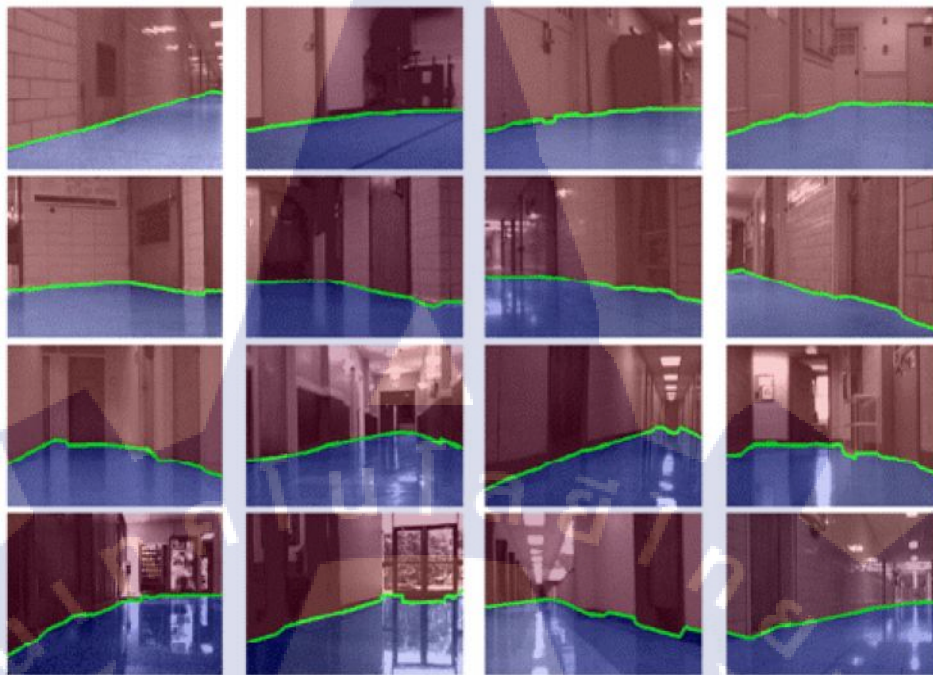
รูปที่ 2.2 แผนภาพการทำงานในงานวิจัยของ Yinxiao และ Stanley

เมื่อคัดกรองเฉพาะเส้นแนวนอนที่เกี่ยวข้องแล้วนำมาคำนวณกับค่าน้ำหนักอีก 3 ชนิดเพื่อเป็นการยืนยันว่าเส้นแนวนอนดังกล่าวเป็นเส้นที่แบ่งระหว่างพื้นทางเดินกับกำแพง โดยค่าน้ำหนักทั้ง 3 ชนิดในสมการที่ (2.1) ได้แก่

1. Structure Score นำภาพต้นฉบับมาทำการเปลี่ยนเป็นภาพไบนารีที่มีการปรับปรุงการทำงานเพื่อหาส่วนที่ควรจะเป็นพื้นจากสมมุติฐานว่าพื้นทางเดินในภาพถ่ายมักมีความสว่างมากกว่ากำแพง
2. Bottom Score เพื่อหาเส้นในแนวนอนที่เป็นพื้นของภาพโดยที่เส้นเหล่านี้จะต้องมีแนวโน้มที่มีการติดกับเส้นแนวนอนตั้งช่วยในการยืนยันว่าเป็นจุดแบ่งระหว่างพื้นกับกำแพง
3. Homogenous Score ใช้การจับกลุ่มสีในภาพโดยมีสมมุติฐานว่าพื้นทางเดินมักมีพื้นที่ในรูปมากที่สุด

เมื่อได้ค่าน้ำหนักทั้ง 3 ชนิดแล้วนำมาแทนในสมการ 2.1 โดยที่ $\phi_s(l_h)$ $\phi_b(l_h)$ และ $\phi_h(l_h)$ คือ ค่า Structure Score Bottom Score และ Homogenous Score ของแต่ละเส้นแนวนอนตามลำดับ ส่วน $w_s w_b$ และ w_h เป็นค่าน้ำหนักแต่ละชนิดซึ่งจะมีน้ำหนักไม่เท่ากันโดยในงานวิจัยนี้จะกำหนดค่า w_s เท่ากับ 1.6 w_b เท่ากับ 0.75 และ w_h เท่ากับ 1.0 แล้วนำไปเปรียบเทียบกับค่า Threshold สำหรับทุกเส้นในภาพ เพื่อได้ผลลัพธ์ในการระบุส่วนที่เป็นพื้นทางเดินแยกออกจากส่วนของกำแพงและอื่นๆ ได้โดยตัวอย่างของผลลัพธ์ตามรูปที่ 2.3

$$\phi_{\text{total}}(l_h) = w_s \phi_s(l_h) + w_b \phi_b(l_h) + w_h \phi_h(l_h) \quad (2.1)$$



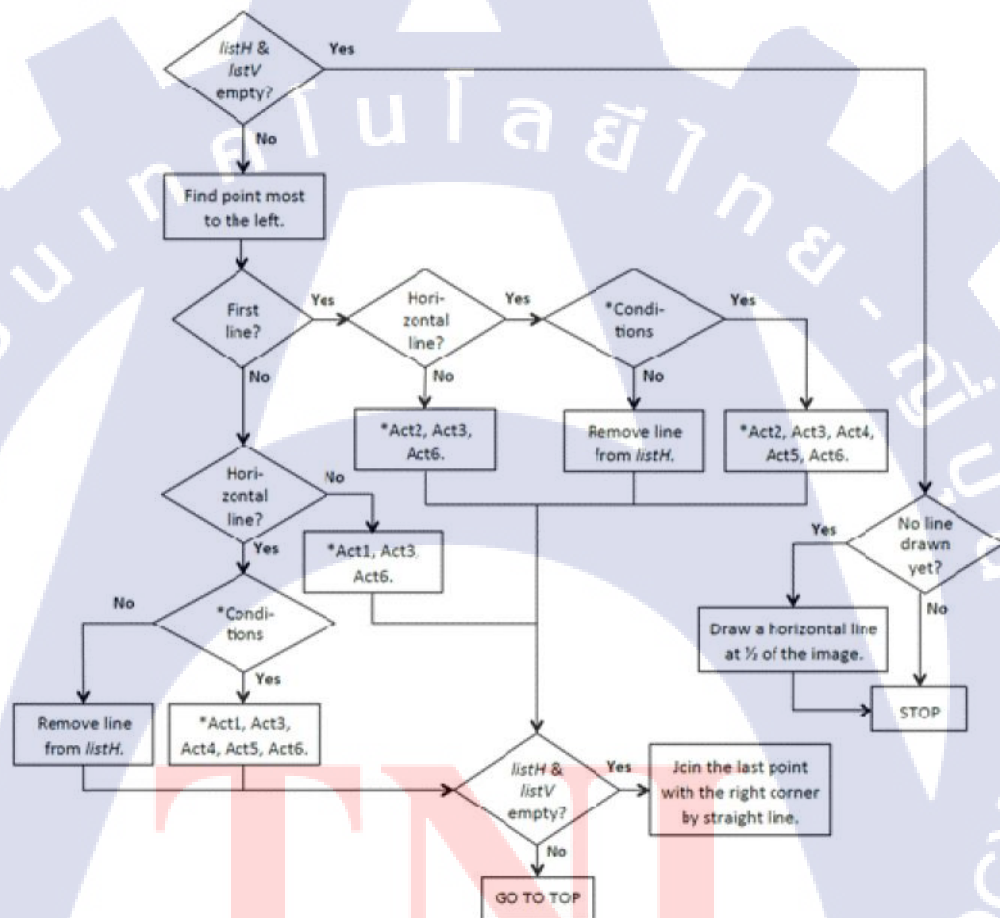
รูปที่ 2.3 ผลลัพธ์งานวิจัยของ Yinxiao และ Stanley

งานวิจัยที่มีลักษณะคล้ายกันคือ สามารถแยกพื้นทางเดินออกจากส่วนอื่นของภาพได้ด้วย ภาพถ่ายหนึ่งภาพคืองานวิจัยของ Gillium Casas Barcelo [14] วิธีการในช่วงต้นจะคล้ายกับวิธีการของ Yinxiao และ Stanley แต่จะแตกต่างหลังจากที่หาเส้นขอบด้วยวิธีการของ Canny Edge Detector แล้วใช้กระบวนการ Hough Transform ในการหาเส้นตรงตามรูปที่ 2.4



รูปที่ 2.4 เส้นตรงที่ได้จากการใช้กระบวนการ Hough Transform

เส้นตรงทั้งหมดที่ได้นำไปแบ่งเป็น 2 กลุ่ม คือ เส้นในกลุ่มแนวแกนตั้งซึ่งองศาของเส้นตรงอยู่ในช่วงระหว่าง 10 องศาจากแนวแกนตั้ง และอยู่ในช่วง 65 องศาจากเส้นแนวแกนนอนโดยเส้นทั้งสองกลุ่มนี้จะต้องมีส่วนหนึ่งอยู่บริเวณครึ่งภาพด้านล่างด้วยสมมุติฐานที่ว่าพื้นทางเดินจะอยู่ด้านล่างของภาพเสมอฉะนั้นกระบวนการค้นหาพื้นทางเดินจะทำการประมวลผลเฉพาะครึ่งภาพด้านล่างเท่านั้น เมื่อแยกเส้นออกได้เป็น 2 กลุ่มเรียบร้อยแล้วนำเข้าประมวลผลเพื่อเชื่อมจุดหรือเส้นด้วยการทำงานตามรูปที่ 2.5 และผลลัพธ์ตามรูปที่ 2.6



รูปที่ 2.5 กระบวนการเชื่อมจุดหรือเส้นเข้าด้วยกันเพื่อแยกพื้นทางเดินออกจากส่วนอื่นของภาพ

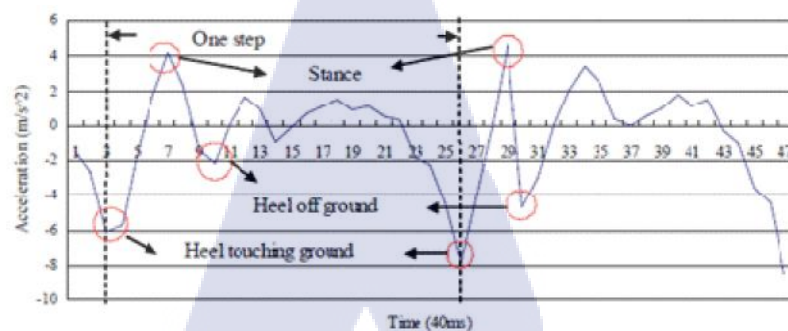


รูปที่ 2.6 ผลลัพธ์ของพื้นที่ได้จากวิธีการของ Gillium

3. การคำนวณระยะทางการเดิน

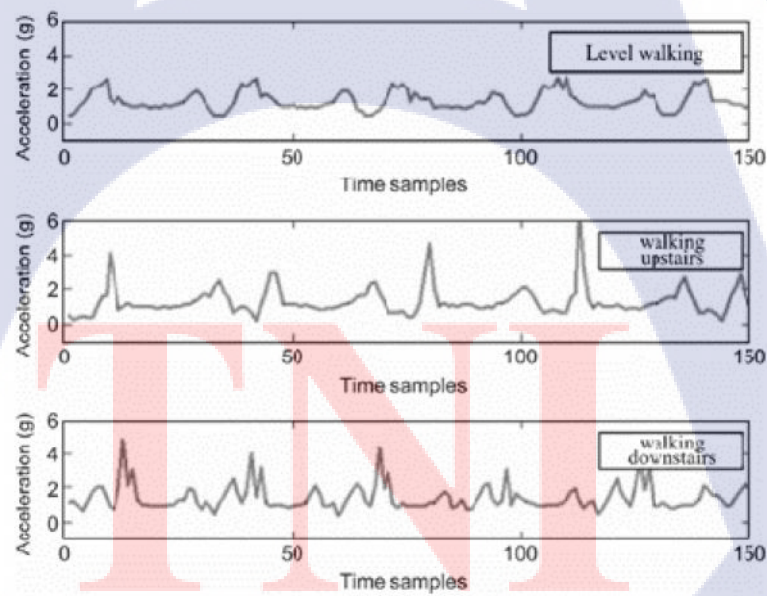
การวัดระยะทางในการเดินนั้นมีหลายวิธี วิธีที่แพร่หลายคือการใช้งาน GPS ในการระบุตำแหน่งระหว่างการเดินหรือวิ่งพร้อมกับคำนวณระยะทางที่เกิดระหว่างจุดสองจุดตลอดช่วงของการทำงานของระบบ และทำการรวมระยะทางที่เกิดขึ้นพร้อมกับแสดงแผนที่การเดินในระหว่างหรือเสร็จสิ้นการทำงานผลเป็นที่น่าพอใจสำหรับการเดินทางระยะไกลๆ แต่จากเอกสารของหน่วยงานที่ดูแลเกี่ยวกับระบบ GPS ในประเทศสหรัฐอเมริกา [15] พบว่า GPS มีความคลาดเคลื่อนสูงสุดที่ 7.8 เมตร ซึ่งเป็นตัวเลขที่สูงถ้าต้องเดินทางในระยะสั้น รวมถึง GPS อาจใช้งานได้ไม่เต็มประสิทธิภาพในบางอาคาร

วิธีหนึ่งที่มีความนิยมคือการใช้จำนวนก้าวในการเดินและคำนวณหาระยะทางการเดิน เช่น งานวิจัยของ Wen Yuah Shih; Liang Yu Chen; and Kun Chan Lan [16] ที่ใช้วิธีการของ Simple Harmonic Motion (SHM) นำมาปรับปรุงเพื่อใช้งานบนสมาร์ทโฟนสำหรับการตรวจจับความเร่งที่เกิดขึ้นในระหว่างการเดิน ซึ่งใช้ Accelerometer และ Gyroscope ในการตรวจจับความเร่ง ค่าที่ได้จาก Accelerometer นั้นจำเป็นต้องหักความเร่งของแรงโน้มถ่วงในแนวแกนตั้งออกเสียก่อนจึงนำมาใช้คำนวณมิเช่นนั้นจะส่งผลทำให้การคำนวณระยะทางในช่วงหลังคลาดเคลื่อนได้ หลังจากนั้นนำความเร่งทั้งหมดมาพล็อตกราฟเพื่อหารูปแบบการเดินและจำนวนก้าวที่เกิดขึ้นในการเดินตามรูปที่ 2.7 หลังจากนั้นนำความเร่งเหล่านั้นคำนวณระยะทางในการเดินโดยทำการอินทิเกรตเป็นจำนวนสองครั้งเพื่อคำนวณหาความเร็วจากการอินทิเกรตครั้งแรกเทียบกับเวลา และได้ผลลัพธ์เป็นระยะทางจากการอินทิเกรตเทียบกับเวลาอีกครั้ง

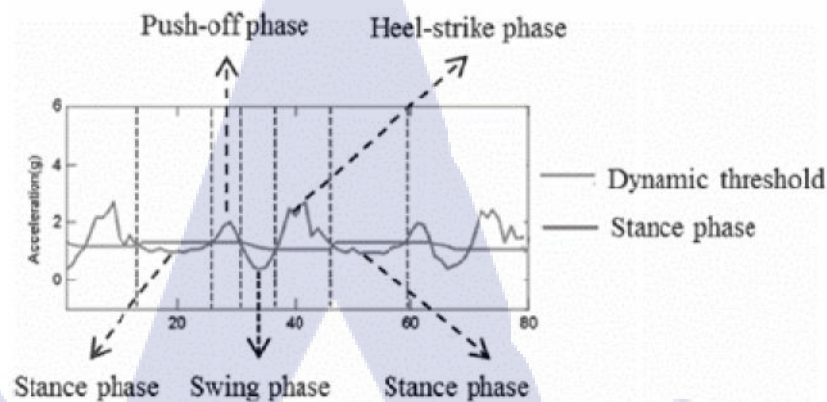


รูปที่ 2.7 ความเร่งที่นำมาพล็อตกราฟเพื่อหารูปแบบการเดินและนับจำนวนก้าว

งานวิจัยของ Jeen-Shing Wang, Che-Wei Lin, C.Yang; and Yu-Jen Ho [17] เป็นงานวิจัยที่ทำการวิเคราะห์รูปแบบการเดินที่ได้จาก Accelerometer โดยอุปกรณ์นั้นจะติดที่บริเวณข้อเท้าของผู้ใช้งานโดยวิจัยกับการเดินขึ้นที่สูง การเดินขึ้นบันได และการเดินลงบันได โดยได้กราฟระหว่างเดินตามรูปที่ 2.8 และนำไปวิเคราะห์ถึงรูปแบบค่าจาก Accelerometer ที่เกิดขึ้นในช่วงระหว่างการเดินว่าเป็นช่วงในของการเดินตามรูปที่ 2.9



รูปที่ 2.8 กราฟที่ได้จาก Accelerometer ในการเดินรูปแบบต่างๆ



รูปที่ 2.9 การวิเคราะห์ช่วงระหว่างการเดิน

การหาค่าประมาณของระยะทางการเดินจากการนับก้าวเดินของผู้ใช้เป็นอีกหนึ่งวิธีการในการหาระยะทางการเดินได้เช่นกัน โดยระยะทางการเดินมีสูตรคำนวณตามสมการที่ 2 โดยที่จำนวนก้าวคือ จำนวนก้าวได้จากการนับก้าวการเดินของผู้ใช้ไม่ว่าจะนับด้วยตนเองหรือใช้อุปกรณ์นับก้าว เช่น Pedometer หรือสมาร์ตโฟนที่มีโปรแกรมนับก้าว สำหรับระยะก้าวนั้นสามารถวัดค่าเฉลี่ยระยะก้าวของตนเองหรือสามารถคำนวณระยะก้าวโดยประมาณได้จากสมการที่ 3 [18] โดยที่ค่าคงที่มีค่าเท่ากับ 0.413 ถ้าเป็นเพศชาย และเท่ากับ 0.415 ถ้าเป็นเพศหญิง

$$\text{ระยะทางการเดิน} = \text{จำนวนก้าว} \times \text{ระยะก้าว} \quad (2.2)$$

$$\text{ระยะก้าว} = \text{ความสูงของผู้เดิน} \times \text{ค่าคงที่} \quad (2.3)$$

4. การสร้างแผนผังในอาคาร

งานวิจัยสำหรับการสร้างแผนผังอาคารในแบบอัตโนมัติหรือกึ่งอัตโนมัติ เช่น งานวิจัยของ Hyojeong Shin; Yohan Chon; and Hojung Sha สร้างระบบ Smart SLAM [19] โดยใช้หลักการของ Wi-fi Fingerprint กับเซ็นเซอร์ Accelerometer และเข็มทิศดิจิทัลช่วยในการสร้างแผนผัง ซึ่งหลักการของ Wi-fi Fingerprint นั้นคือการตรวจจับความแรงของสัญญาณเพื่อใช้ในการคำนวณหาระยะห่างของอุปกรณ์ที่ทำการต่อเชื่อมกับ Access Point ภายในอาคารในการระบุตำแหน่งของอุปกรณ์ที่ผู้ใช้ถืออยู่ แต่สำหรับงานวิจัยนี้จำเป็นต้องทำงานร่วมกับ Access Point ของอาคารในการระบุตำแหน่งซึ่งในบางอาคารอาจไม่มีการติดตั้ง Access Point หรือสัญญาณอาจทั่วถึงทั้งอาคาร

งานวิจัยของ Aditya Sankar; and Steven M.Seitz [20] พัฒนาแอปพลิเคชันบนสมาร์ตโฟน ระบบปฏิบัติการ iOS เพื่อทำการถ่ายวิดีโอและเก็บข้อมูลประกอบอื่นๆ นำไปสร้างแผนผังภายในบ้าน จากการถ่ายวิดีโอโดยการหมุนรอบตัวเองที่บริเวณกลางห้องและทำการเคลื่อนที่ไปเก็บข้อมูลทุกห้อง ภายในบ้านตามรูปที่ 2.10 หลังจากนั้นระบบจะให้ผู้ใช้ช่วยทำการยืนยันและช่วยเหลือด้านข้อมูล เพื่อให้ระบบสามารถสร้างแผนผังของบ้านได้ตามรูปที่ 2.11 ได้แผนผังในรูปแบบของบ้านพักอาศัย หรือห้องในอาคาร แต่ไม่ได้มีการกล่าวถึงการสร้างทางเดินที่เชื่อมต่อกันระหว่างห้องภายในอาคาร



(ก) ผลลัพธ์ได้จากการถ่ายโดยการหมุนรอบตัวเองบริเวณกลางห้อง



(ข) ภาพในขณะที่ย้ายห้อง

รูปที่ 2.10 (ก) ผลลัพธ์การถ่ายภาพโดยถ่ายหมุนรอบตัวเอง (ข) ภาพการเคลื่อนที่ไปยังเก็บข้อมูลภาพห้องถัดไป



รูปที่ 2.11 ผลลัพธ์ของการสร้างแผนผังบ้าน

5. ทฤษฎี Image Processing ที่เกี่ยวข้อง

การประมวลผลของงานวิจัยนี้ส่วนหนึ่งเกี่ยวข้องกับการจัดการและวิเคราะห์ข้อมูลภาพจากวิดีโอ ดังนั้นจึงนำเทคนิคทาง Image Processing ช่วยในการวิเคราะห์รูปภาพเพื่อได้ข้อมูลไปใช้ในการสร้างแผนผังการเดินทางต่อไป โดยข้อมูลที่เป็นเป้าหมายสำคัญได้แก่ พื้นทางเดิน โดยข้อมูลพื้นทางเดินจากในภาพนั้น มีองค์ประกอบการประมวลผลคือ การแบ่งแยกระหว่างพื้นกับทางเดินหรือก็คือการค้นหาเส้นขอบที่อยู่ระหว่างพื้นกับกำแพงภายในภาพ และการค้นหาเส้นตรงนั้นมาจากสมมุติฐานว่า ธรรมชาติของพื้นและกำแพงนั้นจะมีเส้นที่แยกระหว่างสองสิ่งนี้ไว้อย่างชัดเจน และในทางเดินส่วนใหญ่จะเป็นลักษณะของเส้นตรง ดังนั้น ในงานวิจัยนี้ได้แบ่งหัวข้อหลักของการวิเคราะห์ภาพด้วยกระบวนการทาง Image Processing ไว้เป็น 2 หัวข้อดังนี้คือ การค้นหาเส้นขอบ และการค้นหาเส้นตรง

1. การค้นหาเส้นขอบเส้นขอบคือบริเวณที่มีความแตกต่างของสีระหว่างช่วงสองช่วงที่อยู่ติดกัน ซึ่งจะทำให้เห็นความแตกต่างได้อย่างชัดเจนว่าเป็นเส้นหรือช่วงที่แบ่งระหว่างสองส่วนออกจากกันหลักการในการหาเส้นขอบนั้นมีสองประเภทคือ การหาเส้นขอบโดยใช้อนุพันธ์อันดับหนึ่ง และการใช้อนุพันธ์อันดับสองตามรูปที่ 2.12

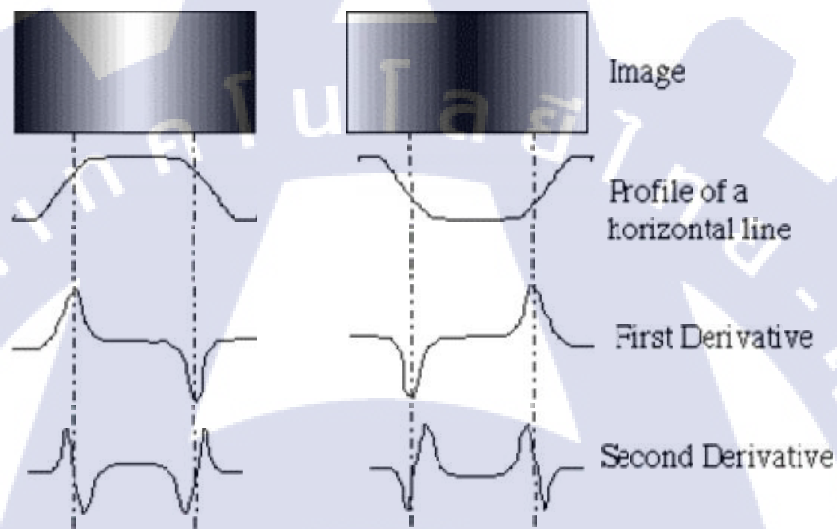
วิธีการค้นหาเส้นขอบจากภาพวิธีหนึ่งคือ วิธีการของ Canny Edge Detection [21] ซึ่งนำหลักการของการค้นหาเส้นขอบโดยใช้อนุพันธ์อันดับหนึ่งร่วมกับกระบวนการอื่นๆ ซึ่งเป็นที่นิยมทั้งการใช้งานทั่วไปและในหลายงานวิจัย การค้นหาเส้นขอบ โดยวิธีการของ Canny Edge Detection นั้นมีขั้นตอนการทำงานดังนี้

1.1 การ Smooth ภาพด้วย Gaussian Filter (Smoothing with Gaussian Filter) เพื่อทำการลดสิ่งรบกวนภายในภาพหรือ Noise

1.2 การคำนวณค่า Gradient (Gradient Calculation) เพื่อขนาดและทิศทางนำไปเปรียบเทียบในขั้นต่อไป

1.3 การขจัดค่าที่ไม่มากที่สุดออก (Non-Maxima Suppression) โดยจะทำให้เส้นขอบที่หาได้มีขนาดเล็กที่สุด

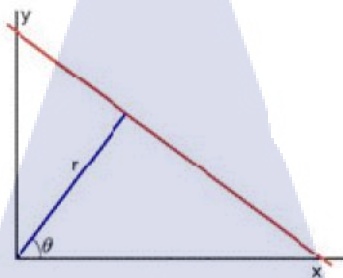
1.4 การกำหนดค่าขีดแบ่ง (Thresholding) เพื่อใช้ในการกรองเอาเส้นขอบที่ไม่ใช่เส้นขอบจริงซึ่งมีผลจาก Noise ที่ตกค้างอยู่ออก



รูปที่ 2.12 รูปเปรียบเทียบระหว่างการหาเส้นขอบด้วยอนุพันธ์อันดับหนึ่งและอนุพันธ์อันดับสอง [22]

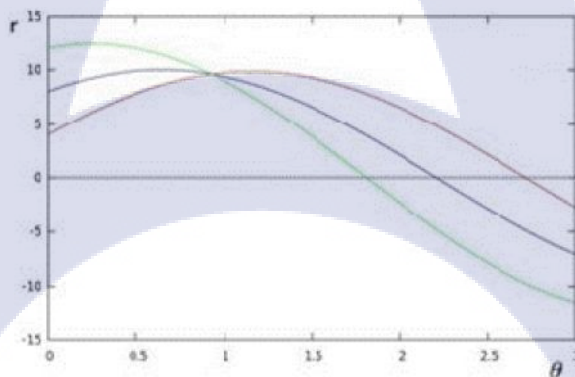
2. การค้นหาเส้นตรงวิธีการของ Hough Transform [23] สามารถหาเส้นตรงได้เนื่องจาก Hough Transform นั้นจะใช้ในการตรวจสอบลักษณะของเส้นในรูปแบบต่างๆ เช่น เส้นตรง วงกลม หรือ เส้นโค้งต่างๆ โดย Hough Transform มีหลักการคือ การนำสมการเส้นตรงมาเขียนอยู่ในระบบพิกัดแบบ Cartesian และระบบพิกัดแบบ Polar ตามรูปที่ 2.13 และนำมาจัดรูปสมการตามสมการที่ 2.4

$$r_0 = x_0 * \cos(\theta) + y_0 * \sin(\theta) \quad (2.4)$$



รูปที่ 2.13 รูปเส้นตรงในระบบพิกัด Cartesian และระบบพิกัด Polar [24]

การหาเส้นตรงจะเริ่มจากการนำค่า x และ y ในระบบพิกัด Cartesian พล็อตลงในกราฟของระบบพิกัด Polar สำหรับทุกจุดที่เป็นเส้นขอบภายในภาพ ถ้าเส้นในกราฟมีการตัดกันเกิดขึ้นหมายความว่าจุดทั้งสองเป็นจุดที่มาจากเส้นตรงเส้นเดียวกัน ตัวอย่างจากรูปที่ 2.14 ซึ่งแสดงให้เห็นว่า 3 จุดที่แสดงบนกราฟมาจากแนวเส้นตรงเดียวกัน



รูปที่ 2.14 กราฟที่นำค่า x และ y ในพิกัดระบบ Cartesian พล็อตกราฟในพิกัดระบบ Polar [24]

บทที่ 3

ขั้นตอนการดำเนินงาน

การดำเนินงานขั้นตอนของการพัฒนาระบบหรือแอปพลิเคชันเพื่อตอบสนองกับความต้องการโดยเนื้อหาในบทนี้จะพูดถึงขั้นตอนการดำเนินงานตั้งแต่เริ่มต้นเพื่อพัฒนาระบบขั้นตอนการดำเนินงานนั้นประกอบไปด้วย 4 ส่วน คือ

1. ภาพรวมการดำเนินงาน
2. การรวบรวมความต้องการของระบบ
3. การวิเคราะห์ความต้องการของระบบ
4. การออกแบบและพัฒนาระบบ

ภาพรวมการดำเนินงาน

งานวิจัยนี้เป็นการพัฒนาระบบขึ้นเพื่อตอบสนองความต้องการ ซึ่งกระบวนการในการพัฒนาระบบหรือซอฟต์แวร์นั้นมีกระบวนการหลากหลายให้เลือกใช้ หนึ่งในจำนวนนั้นคือ SDLC หรือ Software Development Life Cycle ซึ่งเป็นกระบวนการที่กล่าวถึงขั้นตอนในการพัฒนาซอฟต์แวร์ที่มีขั้นตอนชัดเจนและสมบูรณ์ โดยที่ SDLC นั้นได้แบ่งช่วงหรือ Phase ในการพัฒนาซอฟต์แวร์เป็น 5 ช่วงคือ

1. Requirements Phase หรือ ช่วงรวบรวมความต้องการของระบบ
2. Analysis Phase หรือ ช่วงการวิเคราะห์ความต้องการ
3. Design Phase หรือ ช่วงการออกแบบระบบ
4. Implementation Phase หรือ ช่วงการพัฒนาระบบ
5. Maintenance Phase หรือ ช่วงการดูแลรักษาระบบ

ซึ่งขั้นตอนทั้ง 5 ช่วงเป็นการทำงานแบบต่อเนื่องกันโดยข้อมูลในแต่ละช่วงจะเป็นข้อมูลตั้งต้นสำหรับการทำงานในช่วงถัดไป และทำงานต่อเนื่องไปในทุกช่วงของการพัฒนาระบบ จะกระทั่งระบบเสร็จสมบูรณ์

เนื้อหาและรายละเอียดในแต่ละหัวข้อจะอธิบายในหัวข้อถัดไป เนื่องจากงานวิจัยนี้ไม่มีเนื้อหาเกี่ยวกับการดูแลรักษาระบบ ดังนั้นจะคงเหลือหัวข้อ การรวบรวมความต้องการของระบบ การวิเคราะห์ความต้องการ และ การออกแบบระบบและพัฒนาระบบ ที่จะรวมในหัวข้อเดียวกัน

การรวบรวมความต้องการของระบบ

จุดเริ่มต้นของการพัฒนาระบบคือ การค้นหาความต้องการหรือ Requirement ซึ่งเป็นสิ่งที่กำหนดรูปแบบและทิศทางของระบบที่เป็นผลลัพธ์ รวมถึงความสามารถที่ระบบสามารถตอบสนองกับผู้ใช้หลังจากที่พัฒนาระบบเสร็จสิ้นแล้ว ความต้องการนั้นมาได้จากหลายแหล่ง โดยทั่วไปคือผู้ใช้ที่ประสบปัญหาและต้องการระบบใหม่ๆ เพื่อตอบสนองกับการแก้ปัญหาเหล่านั้นได้ หรือมีความต้องการที่จะพัฒนาสิ่งใหม่ๆ เพื่อเพิ่มความสะดวกในการใช้งาน

การรวบรวมความต้องการของงานวิจัยนี้ เริ่มจากความต้องการสร้างแผนผังการเดินทางภายในอาคาร เมื่อได้ความต้องการเบื้องต้นแล้วหลังจากนั้นจึงค้นหาข้อมูลของระบบหรือวิธีการต่างๆ ที่มีอยู่ในปัจจุบันที่เกี่ยวข้องกับการเดินทางในอาคาร เช่น ระบบสร้างแผนผังในอาคารด้วยอุปกรณ์ชนิดต่างๆ ระบบนำทางทั้งในอาคารและนอกอาคาร ระบบค้นหาตำแหน่งในอาคารและข้อมูลอ้างอิงอื่นๆ ที่เกี่ยวข้องกับการเดินทางเช่น การวัดระยะทางการเดินในรูปแบบต่างๆ เมื่อทำการตรวจสอบข้อมูลสิ่งที่ค้นพบคือ ยังไม่มีระบบสร้างแผนผังการเดินทางที่สามารถสร้างแผนผังการเดินทางได้ด้วยอุปกรณ์ของผู้ใช้เพียงอย่างเดียวกล่าวคือ ยังจำเป็นต้องอาศัยข้อมูลจากอุปกรณ์อื่นๆ ที่มีอยู่ในอาคารเช่น การใช้ความแรงของสัญญาณ Wi-Fi เป็นอุปกรณ์ช่วยในการสร้างแผนผังการเดินทาง ซึ่งการกระจายสัญญาณ Wi-Fi ในบางอาคารอาจไม่ถึงทั่วถึงหรือยังจำเป็นต้องใช้มนุษย์เป็นผู้ช่วยระบบในการทำแผนที่หลังจากเก็บข้อมูลเรียบร้อยแล้ว

จากบทที่ 1 และบทที่ 2 ข้อมูลที่เกี่ยวข้องกับการเดินและแผนผังทางเดินในอาคารสามารถสรุปความต้องการของระบบใหม่ได้ 6 ข้อ ดังนี้

1. ระบบสามารถเก็บข้อมูลรูปภาพการเดินทางได้
2. ระบบสามารถเก็บข้อมูลลักษณะการเคลื่อนที่ หรือ เคลื่อนไหวได้ในขณะเดินได้
3. การระบุรูปแบบพื้นทางเดินได้
4. ระบบสามารถสร้างแผนผังการเดินทางได้จากข้อมูลข้อ 1 2 และข้อ 3
5. ระบบสามารถนำข้อมูลที่จัดเก็บได้ นำไปใช้งานต่อเนื่องสำหรับระบบอื่นได้
6. ระบบใช้งานง่ายและใช้อุปกรณ์น้อยชิ้น

การวิเคราะห์ ความต้องการของระบบ

ขั้นตอนถัดไปเมื่อทำการสรุปความต้องการของระบบได้คือการวิเคราะห์ความต้องการของระบบทำการวิเคราะห์เพื่อเลือกวิธีการในการแก้ปัญหา โดยเริ่มจากความต้องการระบบที่สามารถเก็บข้อมูลรูปภาพการเดินทางได้ เบื้องต้นการเก็บข้อมูลภาพนั้นอุปกรณ์ที่เป็นที่นิยมคือ กล้องถ่ายรูป ซึ่งกล้องถ่ายรูปมีหลายประเภท ขนาดและราคา แต่เมื่อคิดถึงความต้องการของระบบที่นำไปใช้ในการเก็บข้อมูลการเดินทาง ฉะนั้น การใช้กล้องถ่ายรูปเพื่อทำการถ่ายรูปในทุกช่วงของการเดินนั้น จะทำให้เกิด

ความยากลำบากในการใช้งานซึ่งขัดกับความต้องการที่ 6 ดังนั้นทางเลือกในเก็บภาพการเดินได้อย่างต่อเนื่องคือ การถ่ายวิดีโอแทนการถ่ายภาพ เพราะการถ่ายวิดีโอเปรียบเสมือนการถ่ายภาพแบบต่อเนื่อง และทำการเก็บข้อมูลง่ายกว่าเมื่อเทียบกับการถ่ายรูปต่อเนื่องเองทีละรูป สำหรับอุปกรณ์ที่นำมาใช้ เมื่อคิดถึงอุปกรณ์ที่อยู่ใกล้ตัวและสามารถถ่ายวิดีโอได้นั้น คือ สมาร์ทโฟนซึ่งในปัจจุบันความสามารถในการถ่ายรูป หรือ ถ่ายวิดีโอ นั้นถือเป็นความสามารถขั้นพื้นฐานของสมาร์ทโฟนทั่วไป อีกทั้งสมาร์ทโฟนยังมีขนาดเล็กและน้ำหนักเบาเหมาะสำหรับการใช้งานขณะเดิน อีกทั้งความละเอียดของภาพก็สูงเพียงพอในการนำไปวิเคราะห์ด้วยกระบวนการ Image Processing ดังนั้นจึงสรุปที่นำสมาร์ทโฟนในการเก็บข้อมูลวิดีโอการเดินเพื่อใช้ในการตอบสนองกับความต้องการที่ 1

สำหรับความต้องการเก็บข้อมูลการเคลื่อนที่หรือการเคลื่อนไหวในขณะเดิน การเดินมีองค์ประกอบที่เกี่ยวข้องคือ อุปกรณ์ต้องสามารถจับการเคลื่อนที่หรือการเดินได้ และ อุปกรณ์ต้องสามารถจับการเลี้ยวที่เกิดขึ้นในขณะเดินได้ เนื่องจากความต้องการข้อที่ 6 คือต้องการความสะดวกในการใช้งานและใช้งานอุปกรณ์น้อยชิ้น ดังนั้นจึงนำสมาร์ทโฟน มาเป็นตัวเลือกแรกในการพิจารณา ซึ่งพบว่าสมาร์ทโฟนสามารถเก็บข้อมูลการเดินทั้งสองส่วนได้ด้วยการใช้เซ็นเซอร์สองชนิดที่ติดตั้งอยู่ในสมาร์ทโฟนทั่วไปคือ Accelerometer และ Gyroscope และในสมาร์ทโฟนบางรุ่นนั้นมีเซ็นเซอร์ที่การทำงานคล้ายกับการทำงานของ Pedometer คือการนับจำนวนก้าวและตรวจจับการก้าวอีกด้วย จึงเลือกใช้สมาร์ทโฟนเพื่อตอบสนองความต้องการที่ 2 เช่นกัน

ส่วนสำคัญสำหรับสมาร์ทโฟนเรื่องหนึ่งคือระบบปฏิบัติการของสมาร์ทโฟนที่นำมาใช้ เนื่องจากระบบปฏิบัติการที่ต่างกัน ฟังก์ชันการใช้งานรวมถึงวิธีการพัฒนาระบบก็แตกต่างกันด้วย งานวิจัยนี้ทำการเลือกระบบปฏิบัติการ Android ด้วยเหตุผลหลักคือ เป็นระบบปฏิบัติการแบบเปิด คือ ผู้พัฒนาทุกคนสามารถทำการพัฒนาระบบได้อย่างอิสระ ไม่มีค่าใช้จ่าย ซึ่งผลพลอยได้คือระบบสามารถเผยแพร่ให้กับผู้ใช้ทั่วไปได้ง่ายรวมถึงราคาของสมาร์ทโฟน Android มีหลายช่วงราคา ซึ่งนับเป็นข้อดีคือ ผู้ใช้ที่ใช้สมาร์ทโฟนราคาประหยัดก็สามารถใช้งานและเข้าถึงระบบที่ทำการพัฒนาได้ สำหรับความต้องการระบุรูปแบบพื้นทางเดินและสร้างแผนผังการเดิน เนื่องจากข้อมูลที่เก็บเป็นภาพจากวิดีโอ ดังนั้นเครื่องมือที่ใช้ต้องมีความสามารถในการนำภาพออกมาจากวิดีโอและทำการวิเคราะห์ภาพเพื่อให้ได้มาซึ่งข้อมูลที่จะนำไปใช้ในการระบุรูปแบบพื้นทางเดินเครื่องมือหนึ่งที่สามารถทำหน้าที่นี้ได้คือ MATLAB โดยเฉพาะอย่างยิ่งเมื่อทำงานร่วมกับส่วนเสริมที่มีชื่อว่า Image Processing Toolbox ที่เป็นเครื่องมือที่ใช้ในการจัดการด้าน Image Processing โดยเฉพาะ และสิ่งที่นอกเหนือจากการประมวลผลภาพแล้ว การประมวลผลเพื่อสร้างแผนผังการเดินหลังจากวิเคราะห์รูปแบบพื้นทางเดินเรียบร้อยแล้ว เพราะการสร้างแผนผังการเดินนั้นจำเป็นต้องใช้ข้อมูลส่วนอื่นประกอบในการสร้างด้วย และ MATLAB สามารถทำโปรแกรมมิ่งในส่วนกระบวนการสร้างแผนผังการ

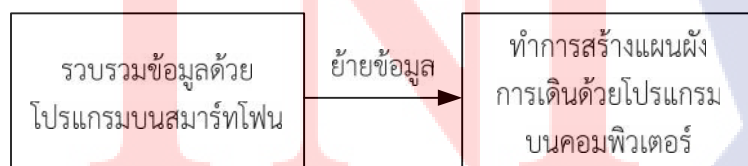
เดินได้ จึงยังทำให้การประมวลผลต่อเนื่องซึ่งเป็นผลดีต่อการใช้งานและลดความผิดพลาด หัวข้อนี้สามารถตอบสนองกับความต้องการที่ 3 และ 4

ความต้องการข้อที่ 5 คือการนำข้อมูลที่เก็บได้นำไปใช้งานในระบบอื่น เนื่องจากระบบนี้จะเป็นการเก็บข้อมูลการเดินทางและสร้างเป็นแผนผังการเดินทาง ดังนั้นข้อมูลที่ทำการเก็บมาคือ ข้อมูลวิดีโอหรือภาพที่ได้จากวิดีโอ ข้อมูลการเดินทาง และ ข้อมูลการเลี้ยว ซึ่งข้อมูลภาพหรือวิดีโอนั้นสามารถนำไปใช้งานกับระบบอื่นๆ ที่ต้องการภาพถ่ายในอาคารไปใช้ในการประมวลได้ เช่น ระบบค้นหาสถานที่ในอาคาร หรือ ระบบระบุตำแหน่งในอาคารที่ใช้ภาพถ่าย

การออกแบบและพัฒนาระบบ

การนำข้อมูลที่ได้จากการวิเคราะห์เพื่อทำการออกแบบระบบ รวมถึงการเลือกสภาพแวดล้อม หรือ Environment ของระบบที่เหมาะสมสำหรับการเลือกสภาพแวดล้อมหลังจากการวิเคราะห์ความต้องการของระบบสามารถแบ่งการทำงานได้เป็นสองส่วน คือ การเก็บข้อมูลการเดินทางและการสร้างแผนผังการเดินทางการเก็บข้อมูลการเดินทางนั้นใช้สมาร์ทโฟนในการเก็บข้อมูล ส่วนการสร้างแผนผังการเดินทางมีแนวคิดที่ใช้สมาร์ทโฟนในการประมวลผลโดยใช้ MATLAB เป็นเครื่องมือในการประมวลผล แต่ด้านข้อจำกัดของ MATLAB ที่ถึงแม้ว่า MATLAB จะมีเวอร์ชัน MATLAB on Mobile ก็ตามแต่ก็ยังข้อจำกัดที่ไม่สามารถระบุได้ว่าจะกระทบกับการพัฒนาระบบเพื่อให้สามารถสร้างแผนผังการเดินทางบนสมาร์ทโฟนหรือไม่ อีกทั้งทรัพยากรบนสมาร์ทโฟนทั่วไปมีน้อยกว่าเครื่องคอมพิวเตอร์ จึงเป็นเหตุผลที่แยกการพัฒนาเพื่อเก็บข้อมูลที่สมาร์ทโฟน และการพัฒนาระบบเพื่อสร้างแผนผังการเดินทางที่คอมพิวเตอร์

ดังนั้นภาพรวมการทำงานของระบบแบ่งออกเป็น 2 ส่วนคือ ส่วนของการเก็บข้อมูลการเดินทางซึ่งใช้สมาร์ทโฟนในการเก็บข้อมูล และส่วนของการประมวลผลสร้างแผนผังการเดินทางซึ่งใช้คอมพิวเตอร์ในการประมวลผลตามรูปที่ 3.1

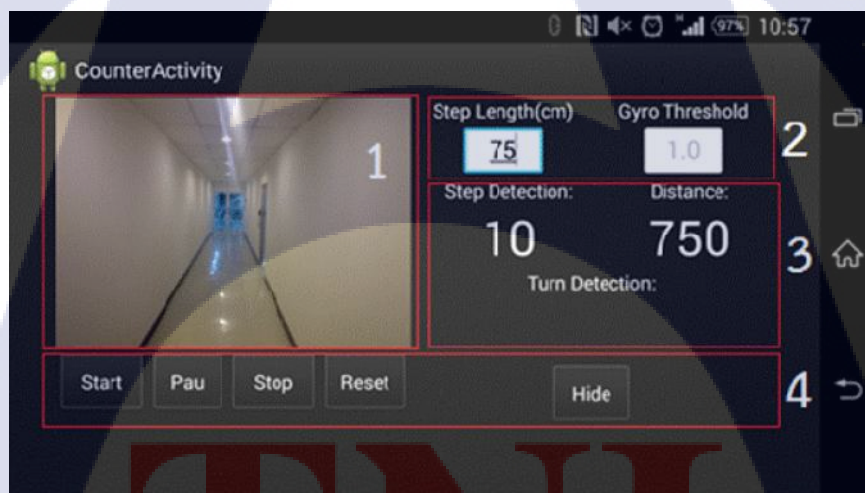


รูปที่ 3.1 ภาพรวมการทำงานของระบบสร้างแผนผังการเดินทาง

หัวข้อถัดไปเป็นการพัฒนาระบบบนสมาร์ตโฟน และการพัฒนาระบบบนคอมพิวเตอร์ ซึ่งจะมีเนื้อหาการออกแบบแทรกอยู่ในแต่ละหัวข้อและอธิบายการออกแบบรวมถึงวิธีการพัฒนาระบบการพัฒนาระบบบนสมาร์ตโฟน Android

การพัฒนาระบบบนสมาร์ตโฟนระบบปฏิบัติการ Android ใช้เครื่องมือที่มีชื่อว่า ADT หรือ Android Development Tools จัดทำโดยบริษัท Google ซึ่งนำ Eclipse เป็นพื้นฐานของเครื่องมือที่ใช้ในการพัฒนาแอปพลิเคชันและผนวกส่วนเพิ่มเติมหรือ Plugin รวมถึงไลบรารีที่จำเป็นในการพัฒนาแอปพลิเคชันบนสมาร์ตโฟนเอาไว้ด้วยกันโดยพื้นฐานของการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android โครงสร้างของภาษามีความคล้ายคลึงกับการพัฒนาแอปพลิเคชันด้วยภาษา JAVA

สำหรับสมาร์ตโฟนที่จะทำลงแอปพลิเคชันที่พัฒนาเรียบร้อยแล้ว ใช้สมาร์ตโฟน Sony Xperia Z2 ระบบปฏิบัติการ Android เวอร์ชัน 4.4 KITKAT เพื่อให้รองรับกับ API Step Detector และ Step Counter ได้ โดยมีรูปแบบหน้าจอการใช้งานดังที่แสดงในรูปที่ 3.2



รูปที่ 3.2 รูปหน้าจอแสดงผลของโปรแกรมบนสมาร์ตโฟน

หน้าจอแสดงผลแบ่งการแสดงผลออกเป็น 4 ส่วน คือ

1. ส่วนแสดงภาพที่สามารถถ่ายได้ในขณะนั้นจากกล้อง
2. ส่วนการกำหนดค่าระยะก้าว และค่า Threshold ของ Gyroscope Threshold
3. ส่วนการแสดงผลจำนวนก้าวการเดิน ระยะทางการเดิน และทิศทางที่เลี้ยว
4. ปุ่มควบคุมการทำงาน

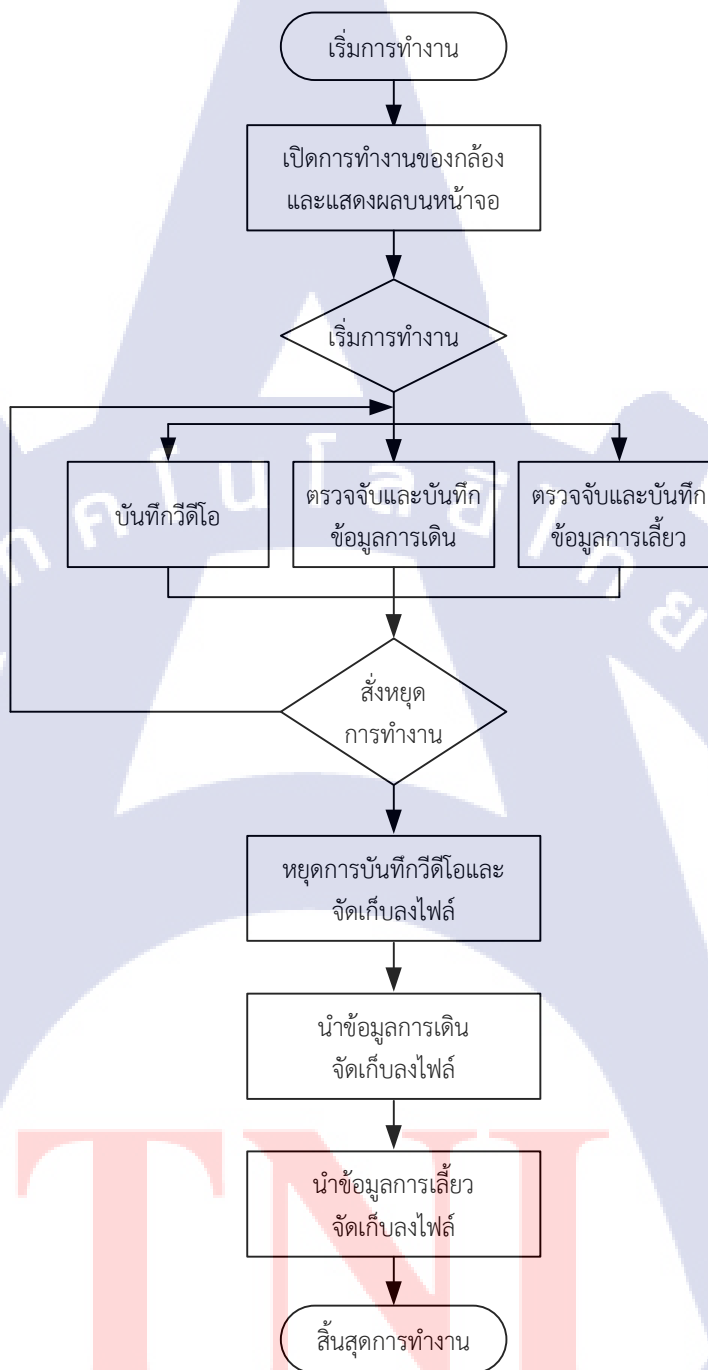
ลำดับการทำงานของแอปพลิเคชันบนสมาร์ตโฟนตาม Flow Chart ในรูปที่ 3.3 จะเริ่มจากผู้ใช้ทำการเปิดใช้งานแอปพลิเคชันบนสมาร์ตโฟน หลังจากนั้นผู้ใช้ทำการกดปุ่มเริ่มการทำงานเพื่อให้ระบบทั้งหมดทำการตรวจจับและเก็บข้อมูลซึ่งข้อมูลดังกล่าวได้แก่ ข้อมูลภาพที่ได้จากการถ่ายภาพวิดีโอ ข้อมูลการเดินที่ได้รับจากเซ็นเซอร์ Step Detector และข้อมูลการเลี้ยวที่ได้รับจากเซ็นเซอร์ Gyroscope จนกระทั่งผู้ใช้กดปุ่มหยุดการทำงาน หลังจากนั้นระบบจะทำการบันทึกข้อมูลทั้งหมดลงในไฟล์ โดยข้อมูลวิดีโอจะถูกจัดเก็บในรูปแบบของไฟล์ .mp4 ส่วนของข้อมูลการเดิน และข้อมูลการเลี้ยวจะถูกจัดเก็บในรูปแบบไฟล์ .txt โดยวิธีการตรวจจับและรายละเอียดการจัดเก็บข้อมูลของแต่ละไฟล์นั้นจะอธิบายในส่วนถัดไป



THAI-NICHI INSTITUTE OF TECHNOLOGY

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

TNIT



รูปที่ 3.3 ขั้นตอนการทำงานของโปรแกรมบนสมาร์ทโฟน

เนื้อหาส่วนนี้จะเป็นการอธิบายวิธีการเรียกใช้เซ็นเซอร์ทั้ง 3 ชนิด โดยเริ่มจากกล้องที่ใช้ในการถ่ายวิดีโอ ซึ่งมี Class ที่จำเป็นในการเรียกใช้ 3 Class ด้วยกัน คือ

1. android.hardware.Camera ใช้ในการติดต่อสั่งการกับกล้องเพื่อทำการถ่ายวิดีโอ
2. android.view.SurfaceHolder ใช้ในการส่งค่าภาพที่ได้จากกล้อง
3. android.view.SurfaceView เป็น Control ที่วางบนหน้าจอแสดงผลเพื่อแสดงผลที่ได้รับจาก Surface Holder

ส่วนต่อมาจะเป็นการทำงานกับเซ็นเซอร์ Step Detector และ Gyroscope ซึ่งสามารถรับข้อมูลได้เมื่อมีการเพิ่มเติม Implement Class ที่ชื่อว่า Sensor Event Listener ซึ่งถ้ามีการเปลี่ยนแปลงใดๆ ที่เซ็นเซอร์สามารถตรวจจับได้ จะเกิด Event ขึ้นที่เมธอด on Sensor Changed แล้วจึงไปแยกภายในเมธอดอีกครั้งว่า Event ที่เกิดขึ้นเป็น Event ที่เกิดจากเซ็นเซอร์ Step Detector ด้วยโค้ด `event.sensor.getType() == Sensor.TYPE_STEP_DETECTOR` หรือ Gyroscope ด้วยโค้ด `event.sensor.getType() == Sensor.TYPE_GYROSCOPE`

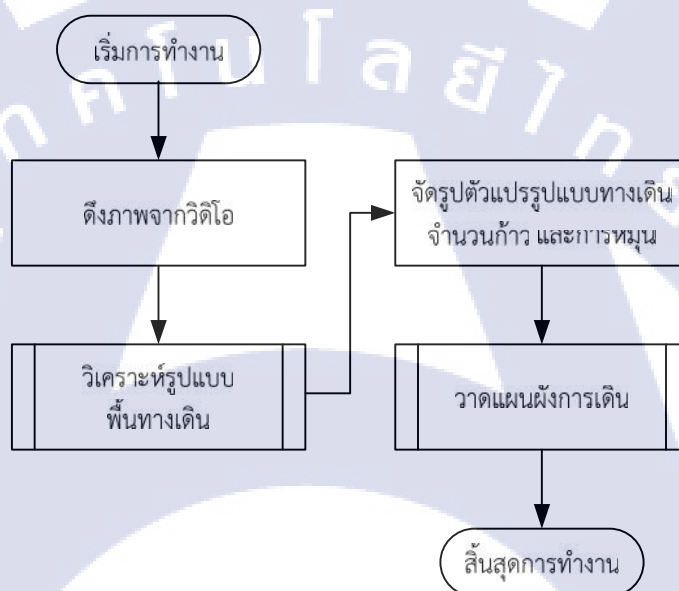
ค่าที่ได้จากเซ็นเซอร์ Step Detector นั้นจะเป็นค่า 1.0 เสมอโดยมีความหมายคือ ระบบสามารถจับการก้าวได้แล้ว 1 ก้าวและจะเกิดเฉพาะจังหวะที่ระบบสามารถตรวจจับการก้าวได้แล้วเท่านั้นและเมื่อตรวจจับการก้าวได้แล้วก็จะทำการเพิ่มจำนวนก้าวรวมของการเดินในครั้งนี้อีก 1 ก้าว

ค่าที่ได้จากเซ็นเซอร์ Gyroscope นั้นจะได้ค่าตัวเลขที่มีหน่วยเป็น rad/s ที่มีค่าได้ทั้งบวกและลบซึ่งค่าบวกหมายถึงเกิดการหมุนไปทางซ้ายหรือทิศทวนเข็มนาฬิกา และค่าลบหมายถึงเกิดการหมุนไปทางขวาหรือทิศตามเข็มนาฬิกาแต่สำหรับเซ็นเซอร์ Gyroscope นั้นจะค่ากลับมาให้กับเมธอดนี้ตลอดเวลา ฉะนั้นในระบบจึงต้องมีการตั้งค่า Threshold ไว้ค่าหนึ่งเพื่อกรองข้อมูลที่เป็นการหมุนตัวเพียงเล็กน้อยหรือเป็นการเดินทางตรงแต่มีการเอี้ยวตัวออกไปซึ่งช่วงการทดลองจากการเก็บข้อมูลไปใช้ในการประมวลผลพบว่าค่า Gyroscope ที่อยู่ในช่วงระหว่าง -1 ถึง 1 นั้นสามารถเก็บข้อมูลการเอี้ยวได้อย่างสมบูรณ์โดยที่ไม่พบสัญญาณการเอี้ยวในระหว่างการเดินทางตรงเลยดังนั้นในการออกแบบครั้งนี้จะใช้ค่า Threshold ที่น้อยกว่า -1 สำหรับการเอี้ยวขวา และมากกว่า 1 สำหรับการเอี้ยวซ้ายจึงจะถือว่าเกิดการเอี้ยวขึ้น และทำการบันทึกข้อมูลการเอี้ยวครั้งนั้นลงในระบบ

และสำหรับการบันทึกข้อมูลการเดินที่เกิดขึ้นนั้น จะมีการบันทึกเวลาในจังหวะที่เกิดขึ้นนั้นลงในระบบด้วย เพื่อนำเอาไปเปรียบเทียบกับข้อมูลภาพจากวิดีโอและนำเอาไปสร้างแผนผังได้อย่างถูกต้อง

การพัฒนาระบบบนคอมพิวเตอร์ด้วยเครื่องมือ MATLAB

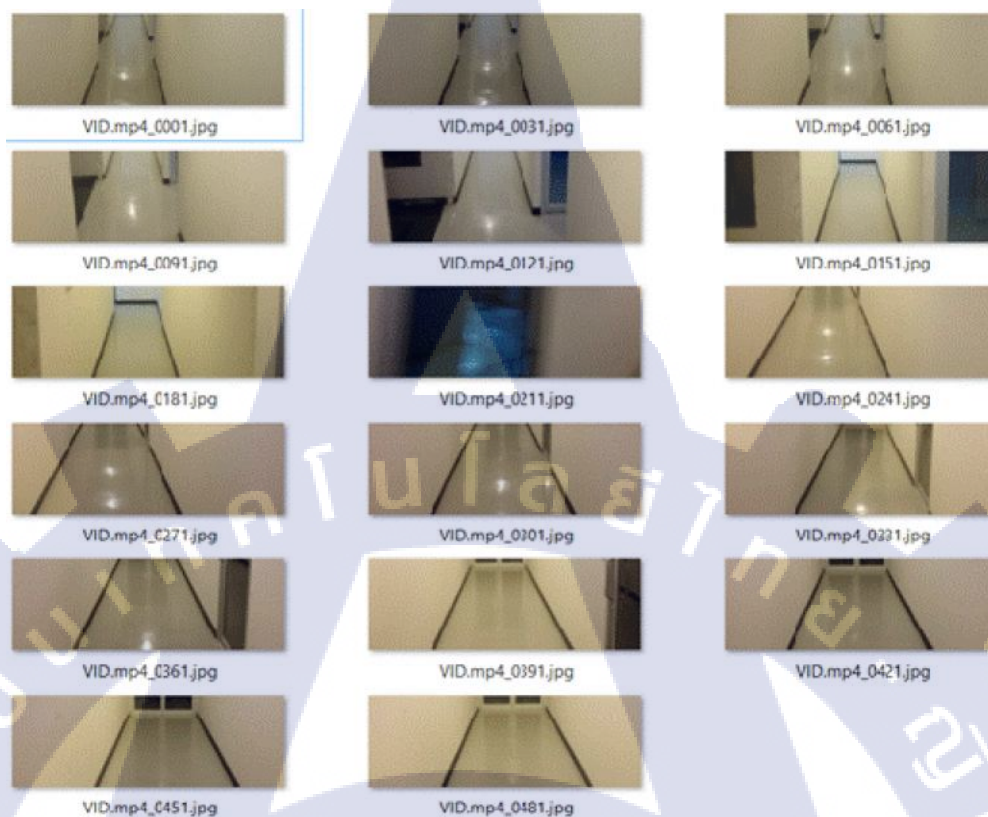
ส่วนของการพัฒนาระบบบนคอมพิวเตอร์นั้น เป็นการรับข้อมูลที่รวบรวมมาจากสมาร์ตโฟน คือ ข้อมูลวิดีโอ ข้อมูลการเดิน และข้อมูลการหมุน ซึ่งรวมอยู่ในไฟล์เดียวกัน ผู้ใช้จะทำการคัดลอกข้อมูลลงมายังเครื่องคอมพิวเตอร์ตามปลายทางที่กำหนดไว้ ซึ่งการประมวลผลทั้งหมดจะใช้เครื่องมือที่ชื่อว่า MATLAB และใช้โมดูลเสริมคือ Image Processing Toolbox ในการจัดการการประมวลผลภาพด้าน Image Processing ขั้นตอนการทำงานมีทั้งหมด 4 ขั้นตอนหลัก ซึ่งภาพรวมการประมวลผลแสดงที่รูปที่ 3.4



รูปที่ 3.4 แผนผังการประมวลผล MATLAB

1. การดึงภาพจากวิดีโอ

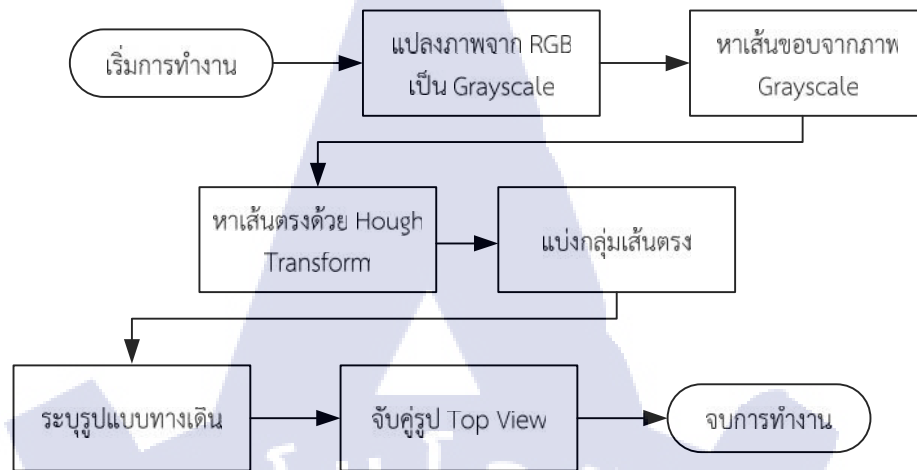
เมื่อทำการนำเข้าข้อมูลจากสมาร์ตโฟนเรียบร้อยแล้ว ทำการดึงภาพจากวิดีโอเพื่อเป็นตัวแทนในการเดินและจับพื้นทางเดินและนำไปสร้างแผนผัง โดยในงานวิจัยนี้จะทำการดึงภาพจากวิดีโอทุกๆ 1 วินาที โดยคำนวณจากเฟรมเรทที่สามารถดึงข้อมูลได้จากไฟล์วิดีโอ ดังรูปที่ 3.5 เพื่อให้ได้จำนวนภาพตามจำนวนวินาทีที่เดินจากการเดินแต่ละครั้ง โดยในแต่ละรูปนั้นจะทำการตัดรูปส่วนครึ่งบนออก เพราะในงานวิจัยนี้จะสนใจเฉพาะพื้นทางเดินและกำแพงด้านล่างเท่านั้น นอกจากช่วยตัดทอนข้อมูลที่ไม่จำเป็นแล้ว ยังลดเวลาในการประมวลผลในขั้นต่อไปอีกด้วย



รูปที่ 3.5 ภาพที่ได้จากวิดีโอทุก 1 วินาที ของการเดิน

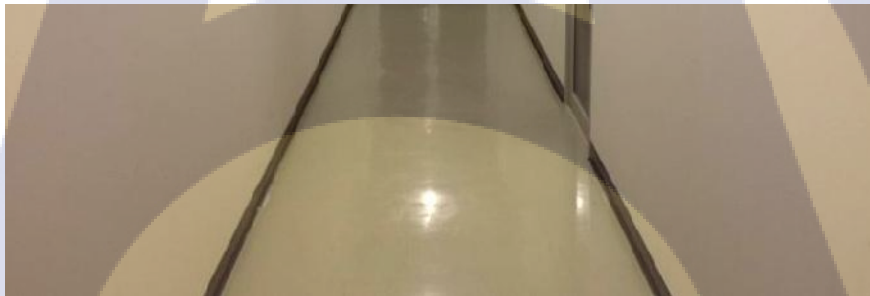
2. การวิเคราะห์รูปแบบพื้นทางเดิน

เป็นขั้นตอนในการวิเคราะห์รูปแบบของพื้นทางเดินจากภาพที่ได้จากข้อ 1 การนำภาพทั้งหมดที่ถูกคัดจากวิดีโอมาทำการหาเส้นขอบโดยจะแบ่งเป็นขั้นตอนย่อยทั้งหมด 6 ขั้นตอน ตามรูปที่ 3.6 คือ

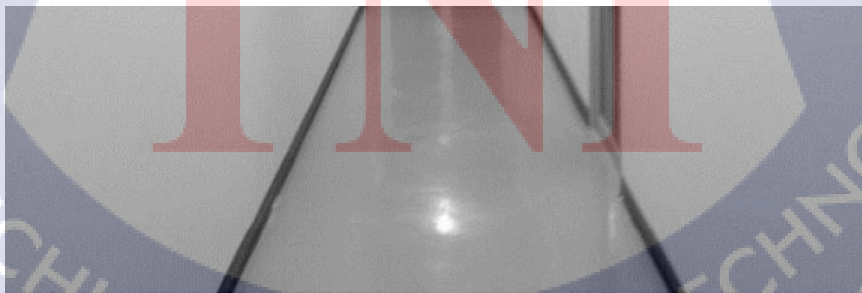


รูปที่ 3.6 ขั้นตอนการวิเคราะห์รูปแบบพื้นทางเดิน

2.1 การทำภาพสีต้นฉบับตามรูปที่ 3.7 ให้เป็นภาพ Grayscale โดยใช้คำสั่ง Rgb 2 Gray ได้ผลลัพธ์ตามรูปที่ 3.8



รูปที่ 3.7 รูปต้นฉบับ



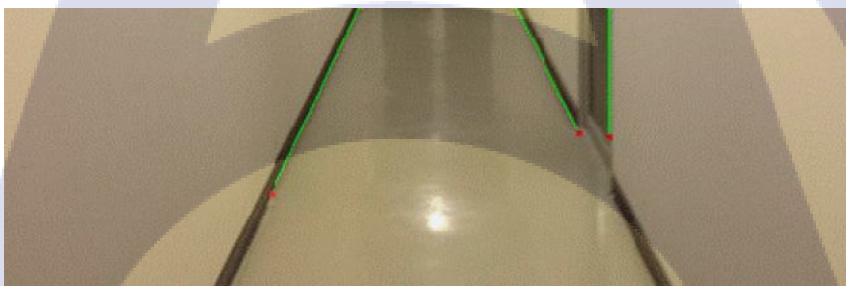
รูปที่ 3.8 การแปลงรูปให้เป็นรูป Grayscale

2.2 นำภาพ Grayscale หาเส้นขอบโดยใช้วิธีการของ Canny Edge Detector ได้ผลลัพธ์ตามรูปที่ 3.9



รูปที่ 3.9 การใช้ Canny Edge Detector เพื่อหาเส้นขอบจากภาพ Gray Scale

2.3 คัดแยกเส้นตรงจากเส้นที่ค้นพบได้โดยใช้หลักการของ Hough Transform และทำการเลือกเฉพาะเส้นตรงที่มีขนาดมากกว่า 30 Pixel ตามรูปที่ 3.10



รูปที่ 3.10 ผลลัพธ์หลังการประมวลผลเพื่อหาเส้นตรงจากภาพ

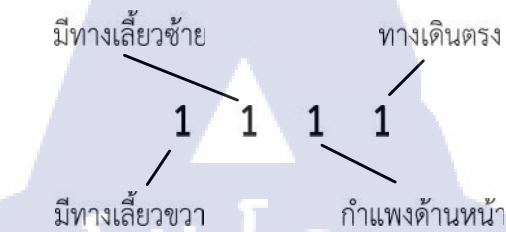
2.4 นำเส้นตรงมาวิเคราะห์เพื่อระบุรูปแบบของทางเดิน โดยจะแบ่งลักษณะของเส้นตรงออกเป็น 3 แบบ โดยใช้องศา Theta ที่ได้จากการคำนวณการ Hough Transform และทำให้เป็นค่า Absolute ก่อนคัดแยกเป็น 3 กลุ่มด้วยเงื่อนไขด้านล่าง ซึ่งได้มาจากการสังเกต รวมถึงข้อมูลงานวิจัยอื่นๆ [13], [14]

2.4.1 กลุ่มเส้นแนวตั้ง เมื่อค่า Theta อยู่ระหว่าง 84 ถึง 90 องศา

2.4.2 กลุ่มเส้นแนวนอน เมื่อค่า Theta อยู่ระหว่าง 0 ถึง 6 องศา

2.4.3 กลุ่มเส้นทางเดิน เมื่อค่า Theta อยู่ระหว่าง 13 ถึง 50 องศา

2.5 ทำการวิเคราะห์กลุ่มเส้นที่อยู่ภายในภาพ เพื่อทำการระบุรูปแบบของทางเดิน โดยรูปแบบของทางเดินนั้นจะอยู่ในรูปของ Bit โดยมีจำนวน 4 หลัก ซึ่งแต่ละหลักจะเป็นตัวแทนของรูปแบบทางเดินตามรูปที่ 3.11



รูปที่ 3.11 ตัวแปรรูปแบบทางเดินในลักษณะของ Bit

2.5.1 ถ้ามีเส้นอยู่ในกลุ่มของ กลุ่มเส้นทางเดิน จะถือว่าภาพนั้นมีทางเดินที่เป็นเส้นตรง และกำหนดค่า Bit หลักขวาสุดเป็น 1

2.5.2 ถ้ามีเส้นอยู่ในกลุ่มของ กลุ่มเส้นแนวนอน จะถือว่าภาพนั้นมีกำแพงด้านหน้าและกำหนดค่า Bit หลักที่ 2 จากขวาเป็น 1

2.5.3 ถ้ามีเส้นอยู่ในกลุ่มของ กลุ่มเส้นแนวตั้ง และเส้นอยู่ทางด้านซ้ายเมื่อเทียบจากกึ่งกลางของภาพ จะถือว่าภาพนั้นมีทางเลี้ยวซ้าย และกำหนด Bit หลักที่ 3 จากขวา เป็น 1

2.5.4 ถ้ามีเส้นอยู่ในกลุ่มของ กลุ่มเส้นแนวตั้ง และเส้นอยู่ทางด้านขวาเมื่อเทียบจากกึ่งกลางของภาพ จะถือว่าภาพนั้นมีทางเลี้ยวขวา และกำหนด Bit หลักซ้ายสุด เป็น 1

2.6 ทำการจับคู่ระหว่างค่าของรูปแบบทางเดินจากข้อที่ 2.6 จากรูปที่ 3.12 โดยมีเงื่อนไขดังนี้

2.6.1 เป็นรูปเส้นตรง 1s ถ้ามีค่าเท่ากับ 1 หรือ 2 หรือ 3

2.6.2 เป็นรูปโค้งซ้าย 1l ถ้ามีค่าเท่ากับ 4 หรือ 6

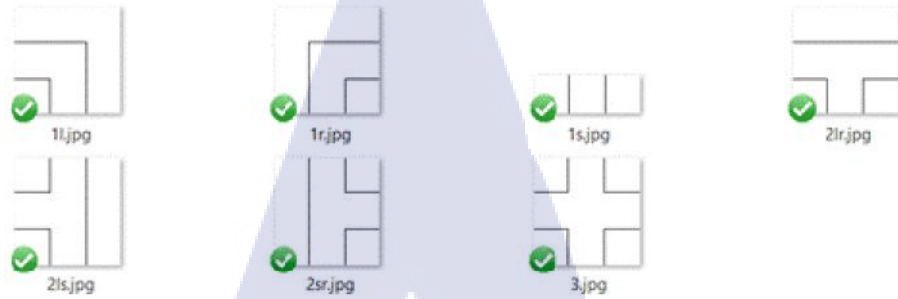
2.6.3 เป็นรูปทางตรงและทางแยกซ้าย 2ls ถ้ามีค่าเท่ากับ 5 หรือ 7

2.6.4 เป็นรูปทางแยกขวา 1r ถ้ามีค่าเท่ากับ 8 หรือ 10

2.6.5 เป็นรูปทางตรงและทางแยกขวา 2sr ถ้ามีค่าเท่ากับ 9 หรือ 11

2.6.6 เป็นรูปสามแยกซ้ายขวา 2lr ถ้ามีค่าเท่ากับ 12 หรือ 14

2.6.7 เป็นรูปสี่แยก 3lsr ถ้ามีค่าเท่ากับ 13 หรือ 15



รูปที่ 3.12 ภาพของชิ้นส่วนแบบ Top View ที่ใช้ในการสร้างแผนผังการเดิน

3. การจัดรูปแบบตัวแปรรูปแบบทางเดิน, จำนวนก้าว และการหมุน

การจัดรูปแบบเป็นการจัดตัวแปรต่างๆ ให้ง่ายต่อการนำไปสร้างแผนผังการเดินโดยการ จัดรูปแบบของตัวแปรในแต่ละวินาทีตั้งแต่วินาทีที่ 1 จนถึงวินาทีที่สิ้นสุดการเดิน ว่า ในวินาทีนั้นๆ มี รูปแบบทางเดินแบบใด มีจำนวนก้าวและมีการหมุนหรือการเลี้ยวหรือไม่โดยมีการจัดการด้วยเงื่อนไข ดังต่อไปนี้

3.1 ทำการรวมการเลี้ยวเป็นครั้งเดียวกัน เนื่องจากในระบบนับการเลี้ยวเมื่อมีช่วงการ เลี้ยวมากกว่า 1 วินาทีเป็นต้นไป เพราะระหว่างการเดินอาจเกิดการเลี้ยวแบบไม่ตั้งใจขึ้น ดังนั้นเพื่อ เป็นการยืนยันการเลี้ยวจำเป็นต้องตรวจจับการเลี้ยวด้วย 2 วินาที จะทำการแก้ไขตัวแปรโดยการ วาง Flag ในรูปแบบทางเดินที่ถัดไปให้เป็น path_skip เพื่อทำการข้ามการวาดแผนผังทางเดินในช่วง นี้ เนื่องจากอยู่ในระหว่างการเลี้ยวโดยจะทำทั้งการเลี้ยวซ้าย และขวา

3.2 ทำการตรวจสอบรูปแบบทางเดินก่อนถึงช่วงวินาทีที่ทำการเลี้ยว เพื่อนำรูปการ เลี้ยวด้านนั้น ออกจากรูปที่อยู่ก่อนหน้าการเลี้ยวเป็นจำนวน 3 วินาที เช่น เกิดการเลี้ยวในวินาทีที่ 7 ดังนั้น รูปแบบทางเดินในช่วงวินาที 5-7 จะสามารถตรวจจับเส้นแกนแนวตั้งทางด้านซ้ายได้ เพื่อทำ การเกิดทางแยกเพียง 1 ครั้งจึงจำเป็นต้องลบรูปแบบการเลี้ยวซ้ายออกจากรูปแบบทางเดินในช่วง วินาทีที่ 5-6 ออกโดยการทำ AND Bit Operation กับรูปแบบทางเดินในวินาทีเหล่านั้นโดย AND Bit Operation ด้วย 0111 เมื่อมีการเลี้ยวขวา และ 1011 เมื่อมีการเลี้ยวซ้าย

3.3 ทำการตรวจสอบรูปแบบทางเดินที่ตรวจจับทางเลี้ยวได้ 3 รูปติดต่อกัน ซึ่งแสดงให้เห็น ว่ามีทางเลี้ยวอยู่ด้านซ้ายหรือขวานั้นจริง ซึ่งถือว่าทางเลี้ยวด้านนั้นๆ เป็นทางเลี้ยวเดียวกัน จึงทำ การ AND Bit Operation กับรูปแบบทางเดินของ 2 วินาทีก่อนหน้าเพื่อให้เหลือเพียงทางเลี้ยว เดียว ซึ่งในข้อนี้ จะแตกต่างกับ 3.2 ตรงจะทำกับตรวจสอบระหว่างการเดินตรงเท่านั้น

3.4 ทำการเพิ่มจำนวนก้าวในช่วงที่ทำการเลี้ยว 1 ก้าว

3.5 ทำการสลับตำแหน่งของการตรวจจับทางเลี้ยวซ้ายและขวาเพื่อนำทางเลี้ยวมาอยู่ในวินาทีสุดท้ายที่เจอทางเลี้ยวกับรูปแบบของ 2 วินาทีก่อนหน้านี้

4. การวาดแผนผังการเดิน

เมื่อทำการจัดรูปแบบข้อมูลเรียบร้อยแล้ว ขั้นตอนสุดท้ายคือการสร้างแผนผังการเดิน โดยมี Flow Chart การทำงานตามรูปที่ 3.13 และจะเริ่มจากการวาดแผนผังจากบริเวณกลางภาพด้านล่างและทิศทางเริ่มต้นจะเป็นการเดินจากด้านล่างขึ้นด้านบน ซึ่งทิศทางการวาดแผนผังนี้จะปรับเปลี่ยนเมื่อพบการเลี้ยวขึ้นในข้อมูลการเลี้ยวรายละเอียดการทำงานดังนี้

4.1 เพิ่มรูปแบบการเดินโดยการดูที่ค่าการเลี้ยวในวินาทีนั้น ถ้าเป็นการเดินตรงจะทำการ OR Bit Operation ด้วย 0001 หรือ หรือเป็นการเลี้ยวซ้ายหรือขวา จะทำการ OR Bit Operation ด้วย 0100 หรือ 1000 ตามลำดับ

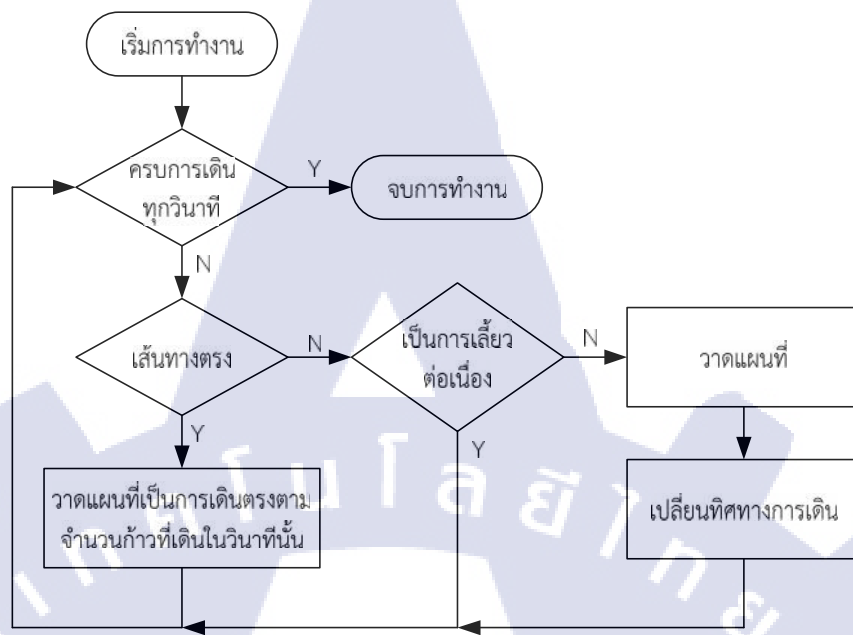
4.2 เช็ครูปแบบทางเดินว่ามีทางเลี้ยวซ้ายหรือขวาเกิดในภาพหรือไม่ ซึ่งทางเลี้ยวนี้ไม่จำเป็นต้องเกิดการเลี้ยวจริง

4.2.1 ถ้าไม่มีทางเลี้ยว จะทำการวาดทางตรงตามจำนวนก้าวที่เกิดขึ้นในวินาทีนั้น

4.2.2 ถ้ามีทางเลี้ยว ไม่ว่าจะเป็นทางเลี้ยวซ้ายหรือขวา จะทำการวาดขึ้นส่วนทางเดินลงไป แต่จะวางเพียงขึ้นเดียวโดยไม่สนจำนวนก้าวที่เกิดขึ้น

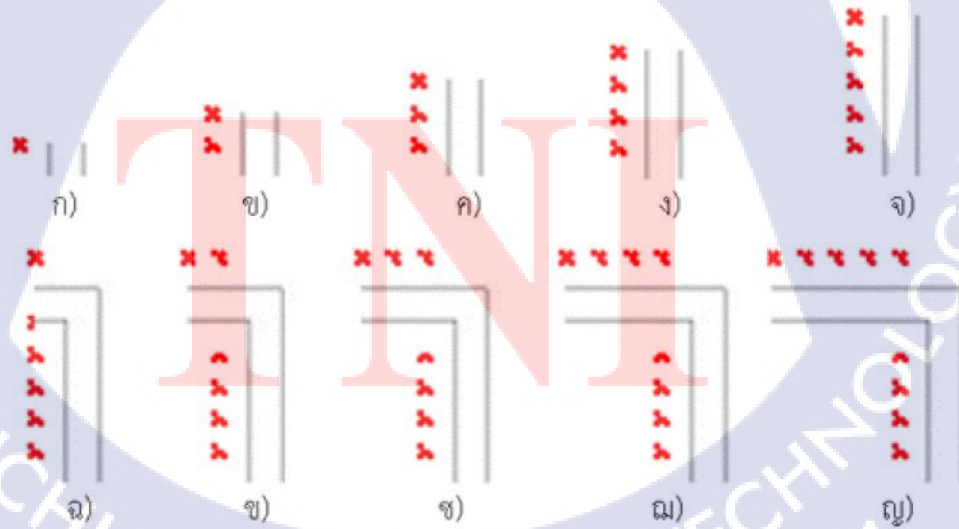
4.3 ตรวจสอบการเลี้ยวในวินาทีนั้นว่ามีการเลี้ยวเกิดขึ้นหรือไม่ ถ้ามีจะทำการปรับทิศทางการวาดแผนผัง โดยหมุนไปทางทิศทวนเข็มนาฬิกา 90 องศาถ้าเกิดการเลี้ยวซ้าย หรือ หมุนไปทางทิศตามเข็มนาฬิกา 90 องศา ถ้าเกิดการเลี้ยวขวาและทำการหมุนรูปทางเดินทั้งหมดตามทิศทางที่หมุนไปข้างต้นด้วย

TNI



รูปที่ 3.13 Flow Chart การสร้างแผนผังการเดิน

ในขณะที่ทำการสร้างแผนผังการเดินเป็นลักษณะการต่อขึ้นส่วนทางเดินไปจนกว่าจะครบทุกวินาทีของการเดินตามรูปที่ 3.14 เมื่อทำการสร้างแผนผังได้ผลลัพธ์แผนผังการเดินตามรูปที่ 3.15 เป็นการจบการทำงานสำหรับการเดิน 1 ครั้ง



รูปที่ 3.14 ก) ถึง ง) รูปแผนผังการเดินที่ทำการเพิ่มเติมไปในแต่ละช่วงวินาทีของการเดิน



รูปที่ 3.15 ผลลัพธ์สุดท้ายของการสร้างแผนผังการเดิน

TNI

THAI

NICHI INSTITUTE OF TECHNOLOGY

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

บทที่ 4

ผลการทดลอง

ในส่วนของการทดลองแบ่งออกเป็น 3 ส่วน คือ ส่วนของการเก็บข้อมูลและทดสอบสมมุติฐานในการเดินส่วนการเก็บข้อมูลการทดลอง และส่วนของการบันทึกผลการทดลอง

การเก็บข้อมูลและทดสอบสมมุติฐานต่างๆ ในการเดิน

ในการวิจัยนี้เกี่ยวข้องกับจำนวนก้าวในการเดินและระยะการก้าวของผู้ใช้งานเพื่อนำมาคำนวณเป็นระยะทางการเดิน ดังนั้นจึงทำการทดสอบสมมุติฐานที่เกี่ยวข้องกับสองหัวข้อนี้ โดยระยะทางการเดินรวมเกิดขึ้นจากจำนวนก้าวที่เกิดขึ้นในการเดินคูณด้วยระยะการก้าวในแต่ละก้าวซึ่งจะเป็นค่าที่ได้จากการคำนวณ ดังนั้นการทดสอบสมมุติฐานเกี่ยวกับการเดินนั้นก็จะมี 2 ส่วนคือ การทดสอบเรื่องการคำนวณระยะการก้าวและการทดสอบความแม่นยำของ API Step Detector

การทดสอบแรกจะทำการทดสอบระยะการก้าว โดยอ้างอิงถึงสูตรการคำนวณระยะก้าวในสมการที่ 2.2 โดยที่ค่าคงที่จะเป็น 0.413 ถ้าเป็นเพศหญิง และ 0.415 ถ้าเป็นเพศชาย

การทดสอบโดยทำการเดินเป็นระยะ 6-7 ก้าวเป็นจำนวน 11 ครั้ง รวมทั้งสิ้น 35 ก้าวซึ่งไม่นับการเดินในช่วง 3 ก้าวแรกที่เป็นช่วงเริ่มต้นเพราะการก้าวที่ยังไม่คงที่แล้วจึงทำการวัดระยะการก้าวของแต่ละก้าวและทำการจดบันทึก โดยผู้ทำการทดสอบนั้นเป็นผู้ชายมีความสูง 180 ซม. และได้ผลการทดสอบตามตารางที่ 4.1

ตารางที่ 4.1 ระยะและจำนวนการก้าวในการทดสอบ

ระยะก้าว (ซม.)	69	70	71	72	73	74	75	76	77	78	79	80
ความถี่	1	6	3	3	4	3	4	5	1	2	2	1

จากการเดินข้างต้นจะได้ค่าเฉลี่ยของระยะก้าวอยู่ที่ 73.8 ซม. ซึ่งถ้านำไปเปรียบเทียบกับสูตรคำนวณโดยผู้ชายที่มีความสูง 180 ซม. จะมีระยะก้าวประมาณ 74.7 ซม. ดังนั้นจะมีระยะทางที่ห่างกันประมาณ 0.9 ซม. ต่อการก้าว 1 ก้าวหรือคิดเป็น 1.2% เมื่อเทียบกับระยะก้าวจริง

การทดสอบต่อมาคือการทดสอบ APIstep Detector ของ Android โดยการทดสอบนั้นจะทำการเดินเป็นเส้นตรงในสถานที่ 3 แห่งที่ระยะการเดินที่แตกต่างกันตามตารางที่ 4.2

ตารางที่ 4.2 จำนวนก้าวจริงและจำนวนก้าวที่ได้จากระบบ

การเดินที่	1	2	3	4	5	6	7	8	9
จำนวนก้าวจริง	268	70	54	63	79	61	96	62	97
จำนวนก้าวจากระบบ	268	70	54	62	78	62	96	62	97
แตกต่าง	0	0	0	1	1	1	0	0	0

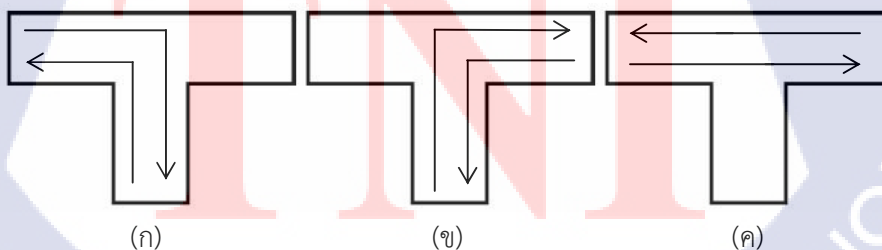
จากผลการทดลองการนับก้าวของ API จาก Android มีผลการนับก้าวคลาดเคลื่อน 3 ครั้ง จากการเดินทั้งหมด 9 ครั้ง ซึ่งการนับที่คลาดเคลื่อนไปนั้นอยู่ที่ 1 ก้าวเท่านั้นนับว่าเซ็นเซอร์สามารถทำงานได้อย่างถูกต้อง

การเก็บข้อมูลเพื่อนำมาใช้ในการทดลอง

การคัดเลือกสถานที่ที่นำมาใช้ในการเก็บข้อมูลการทดลองนั้นมีเงื่อนไขดังนี้

1. ทางเดินเป็นแนวระนาบ ไม่มีการขึ้นหรือลงบันไดระหว่างการเดิน
2. ทางเดินเป็นเส้นตรงที่เห็นกำแพงอยู่ภายในรูปทั้ง 2 ด้าน หรือน้อยหนึ่งด้าน
3. ทางเลี้ยวซ้ายและขวาดังฉากกับทางเดินตรง
4. มีแสงสว่างเพียงพอที่เก็บภาพแล้วสามารถนำไปประมวลผลทาง Image Processing

เงื่อนไขและวิธีการเก็บข้อมูลที่ได้กล่าวมาแล้วนั้น จึงได้เลือกสถานที่ทั้งหมด 4 แห่ง ซึ่งแต่ละแห่งจะทำการเดินในบริเวณทางแยกเดียวกันแต่เดินในเส้นทางที่ต่างกันจำนวน 3 เส้นทาง และอีกแห่งเป็นการเดินในเส้นทางที่มีการเลี้ยวสองครั้งภายในการเดินครั้งเดียว เป็นตัวเลือกในการเก็บข้อมูลการเดินซึ่งแต่ละเส้นทางจะเก็บข้อมูลการเดินทั้งขาไปและขากลับซึ่งจะนับเป็นการเดินคนละครั้งตามรูปที่ 4.1 และเนื่องจากการเดินไปกลับนั้นมีระยะทางจริงเท่ากันแต่นับการเดินไปเป็น 1 ครั้งและกลับ 1 ครั้ง



รูปที่ 4.1 รูปแบบการเดินที่ใช้ในการทดลอง (ก) เส้นทางที่ 1 (ข) เส้นทางที่ 2 (ค) เส้นทางที่ 3 (ง) เส้นทางที่ 4

เมื่อทำการบันทึกข้อมูลการเดินทางบนสมาร์ทโฟนและทำการคัดลอกข้อมูลทั้งหมดจากบนสมาร์ทโฟนไว้ที่คอมพิวเตอร์ต่อจากนั้นใช้เครื่องมือ MATLAB เพื่อเปิดชุดคำสั่งที่พัฒนาขึ้นและทำการเปลี่ยนสถานที่บันทึกข้อมูลการเดินทางที่ต้องการบนเครื่องคอมพิวเตอร์แล้วทำการรันชุดคำสั่งรอผลลัพธ์แผนผังการเดินทางขึ้นบนหน้าจอถือว่าสิ้นสุดการทำงาน

ขั้นตอนที่เกิดขึ้นในขณะที่ชุดคำสั่งกำลังทำงานนั้นจะมีขั้นตอนดังต่อไปนี้

1. นำข้อมูลวิดีโอถึงข้อมูลภาพวินาทีละ 1 เฟรมตามเฟรมเรทของวิดีโอและทำการตัดรูปภาพครึ่งบนทิ้งไปพร้อมกับบันทึกภาพทุกวินาทีไว้ที่เดียวกับไฟล์วิดีโอผลลัพธ์ตามรูปที่ 4-2 ถึงรูปที่ 4.5
2. นำภาพการเดินทั้งหมดเปลี่ยนจากภาพ RGB เป็น Grayscale ผลลัพธ์ตามรูปที่ 4.6 ถึงรูปที่ 4.9
3. นำภาพ Grayscale หาเส้นขอบด้วยกระบวนการ Canny Edge Detection ผลลัพธ์ตามรูปที่ 4.10 ถึงรูปที่ 4.13
4. นำเส้นขอบเข้ากระบวนการ Hough Transform เพื่อหาเส้นตรงและทำการบันทึกภาพต้นฉบับที่ระบุเส้นตรงที่ตรวจพบลงในสถานที่เดียวกับข้อมูลวิดีโอผลลัพธ์ตามรูปที่ 4.14 ถึงรูปที่ 4.17
5. นำเส้นตรงที่ได้ คัดแยกกลุ่มของเส้นด้วยองศา Theta ที่ได้จากกระบวนการ Hough Transform และจัดรูปแบบให้อยู่ในตัวเลข Pattern Code 4 ดิจิตผลลัพธ์ตามรูปที่ 4.18 และรูปที่ 4.19
6. นำข้อมูล Pattern Code ข้อมูลการก้าว ข้อมูลการเลี้ยว จัดรูปแบบในรูปแบบ Array โดยที่ Array แต่ละค่าจะแทนวินาทีที่เกิด Pattern Code จำนวนการก้าว และข้อมูลการเลี้ยวที่เกิดขึ้นและทำการจัดรูปแบบข้อมูลเพื่อสะดวกต่อการวาดแผนผังการเดินในขั้นตอนต่อไปได้ผลลัพธ์ตามรูปที่ 4.20 และรูปที่ 4.21 โดยที่ตารางที่ซ้อนกันด้านซ้ายสุดคือ Pattern Code ตารางตรงกลางคือข้อมูลการเลี้ยว และตารางด้านขวา คือ ตารางจำนวนก้าว
7. สร้างแผนผังการเดินและได้ผลลัพธ์ตามรูปที่ 4.22

TNI

THAI-NICHI INSTITUTE OF TECHNOLOGY



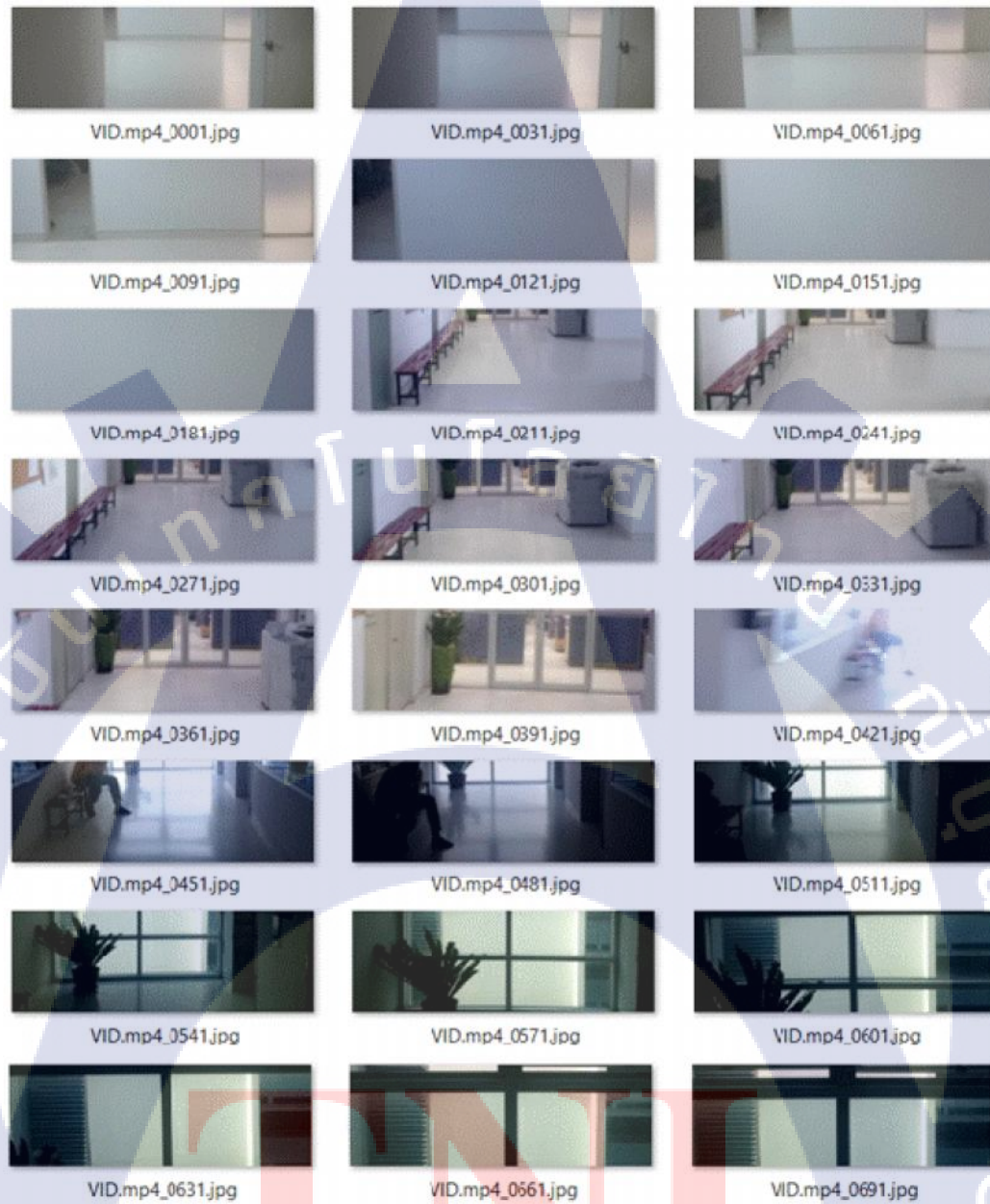
รูปที่ 4.2 การเดินที่ Sino Thai ชั้น 10



รูปที่ 4.3 การเดินที่ LPN อาคาร A ชั้น 9



รูปที่ 4.4 การเดินที่สถาบันเทคโนโลยีไทย – ญี่ปุ่น อาคาร A ชั้น 6



รูปที่ 4.5 การเดินที่สถาบันเทคโนโลยีไทย – ญี่ปุ่น อาคาร A ชั้น 6



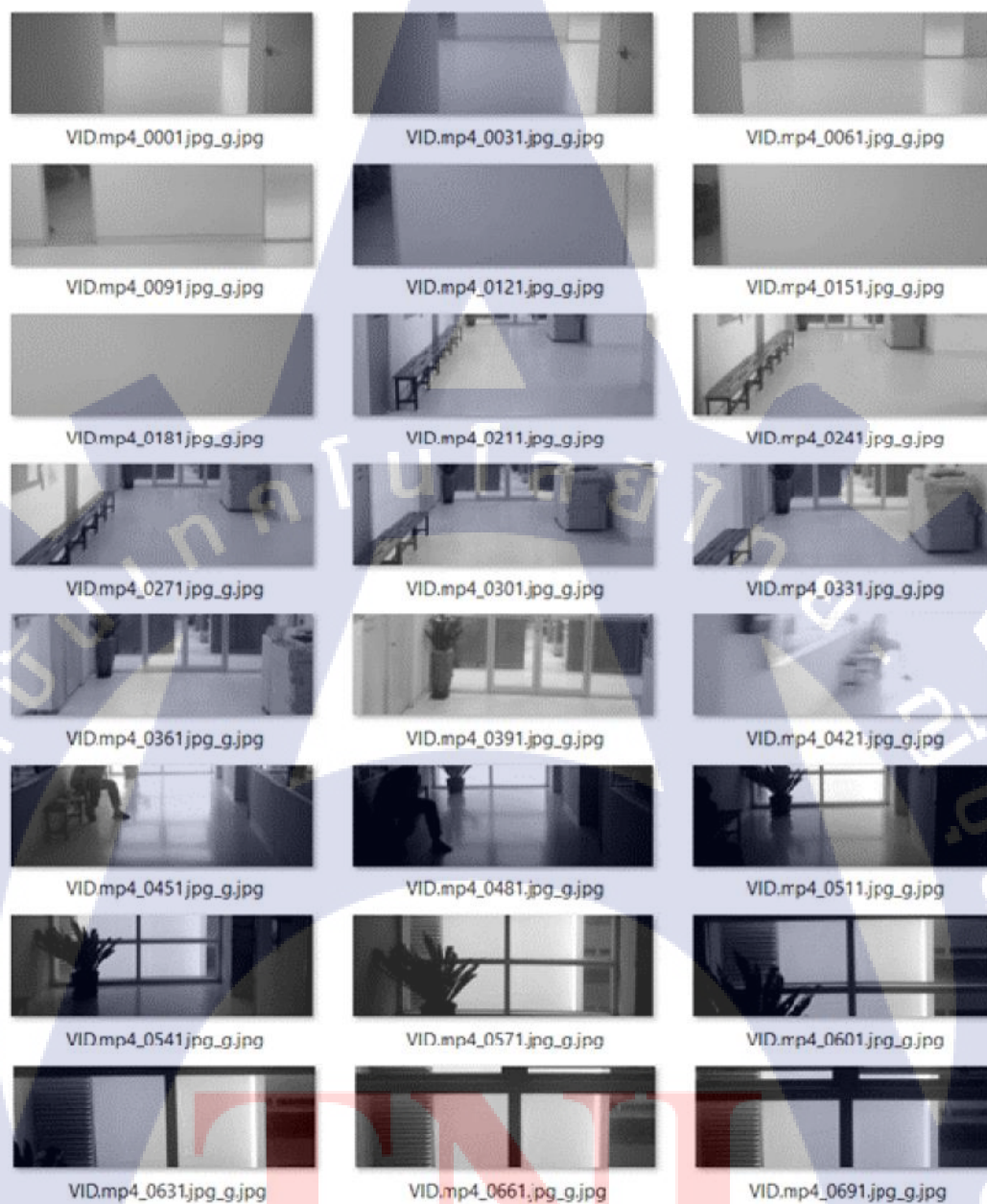
รูปที่ 4.6 ผลลัพธ์ Grayscale ของการเดินที่ Sino Thai ชั้น 10



รูปที่ 4.7 ผลลัพธ์ Grayscale ของการเดินที่ LPN อาคาร A ชั้น 9



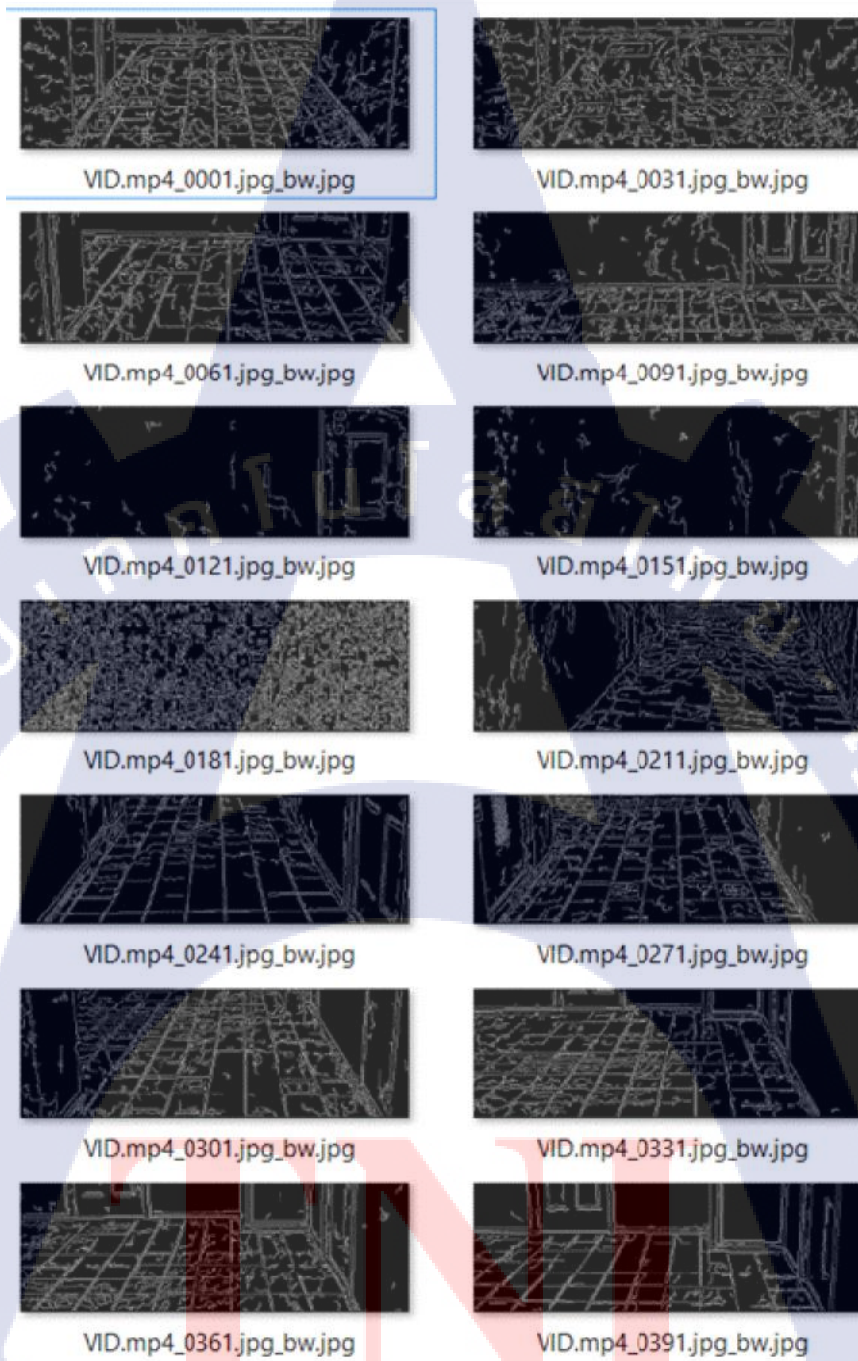
รูปที่ 4.8 ผลลัพธ์ Grayscale ของการเดินที่สถาบันเทคโนโลยีไทย - ญี่ปุ่น อาคาร A ชั้น 6



รูปที่ 4.9 ผลลัพธ์ Grayscale ของการเดินที่สถาบันเทคโนโลยีไทย – ญี่ปุ่น อาคาร A ชั้น 5



รูปที่ 4.10 ผลลัพธ์ Canny Edge Detection จากการเดินที่ Sino Thai ชั้น 10



รูปที่ 4.11 ผลลัพธ์ Canny Edge Detection จากการเดินที่ LPN อาคาร A ชั้น 9



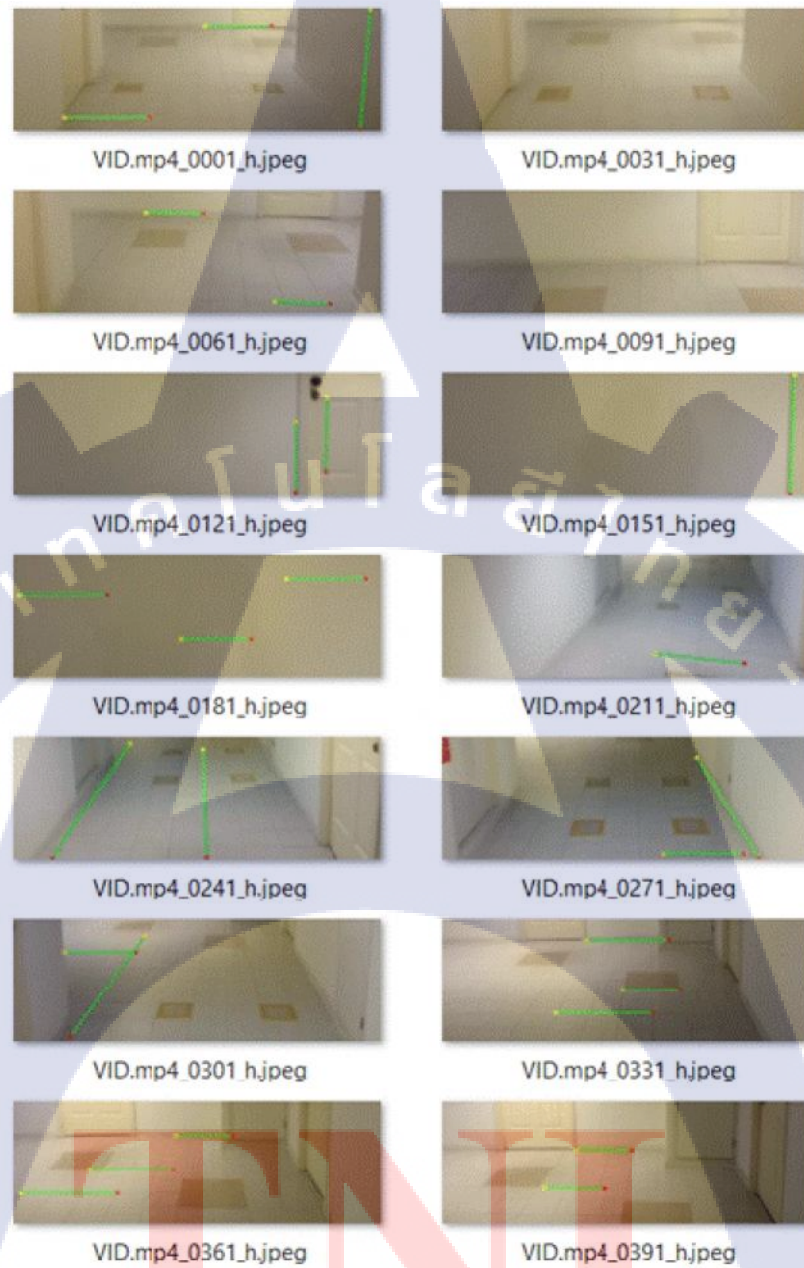
รูปที่ 4.12 ผลลัพธ์ Canny Edge Detection จากการเดินที่สถาบันเทคโนโลยีไทย – ญี่ปุ่น อาคาร A
ชั้น 6



รูปที่ 4.13 ผลลัพธ์ Canny Edge Detection จากการเดินที่สถาบันเทคโนโลยีไทย – ญี่ปุ่น อาคาร A
ชั้น 5



รูปที่ 4.14 ผลลัพธ์ Hough Transform จากการเดินที่ Sino Thai ชั้น 10



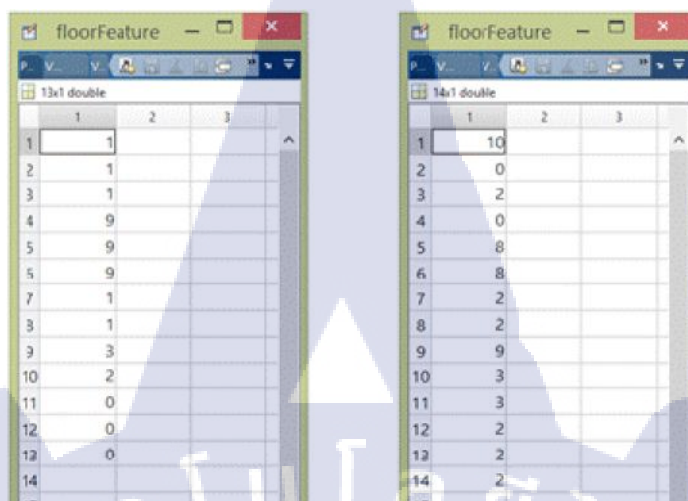
รูปที่ 4.15 ผลลัพธ์ Hough Transform จากการเดินที่LPN อาคาร A ชั้น 9



รูปที่ 4.16 ผลลัพธ์ Hough Transform จากการเดินที่สถาบันเทคโนโลยีไทย - ภูเก็ต อาคาร A ชั้น 6



รูปที่ 4.17 ผลลัพธ์ Hough Transform จากการเดินที่สถาบันเทคโนโลยีไทย – ญี่ปุ่น อาคาร A ชั้น 5

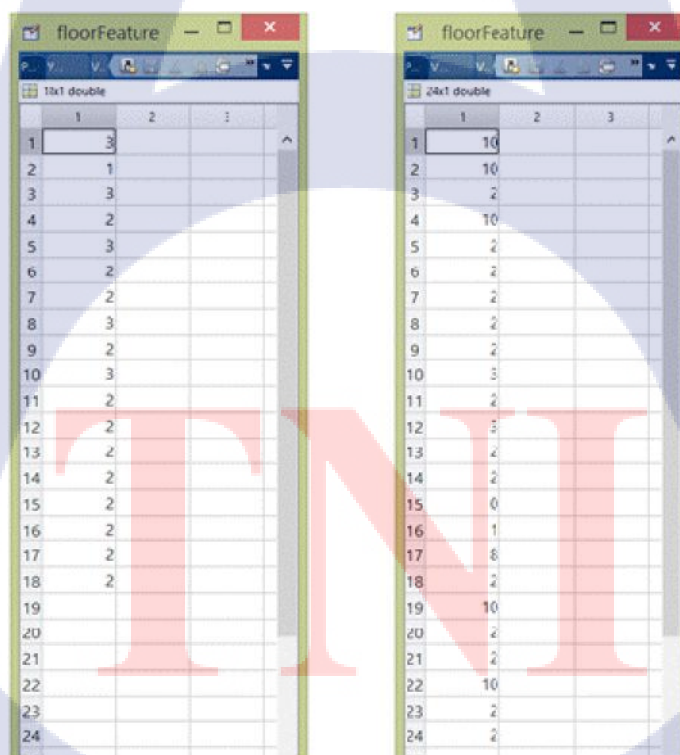


The image shows two side-by-side screenshots of a software application titled 'floorFeature'. Each window displays a table with 3 columns and 14 rows. The left window is titled '13x1 double' and the right window is titled '14x1 double'. Both tables contain numerical data representing floor features.

	1	2	3
1	1		
2	1		
3	1		
4	9		
5	9		
6	9		
7	1		
8	1		
9	3		
10	2		
11	0		
12	0		
13	0		
14			

	1	2	3
1	10		
2	0		
3	2		
4	0		
5	8		
6	8		
7	2		
8	2		
9	9		
10	3		
11	3		
12	2		
13	2		
14	2		

รูปที่ 4.18 ข้อมูล Pattern ที่ได้จากการวิเคราะห์เส้นตรงภาพซ้ายที่อาคาร Sino Thai และภาพขวาอาคาร LPN



The image shows two side-by-side screenshots of a software application titled 'floorFeature'. Each window displays a table with 3 columns and 24 rows. The left window is titled '13x1 double' and the right window is titled '24x1 double'. Both tables contain numerical data representing floor features.

	1	2	3
1	3		
2	1		
3	3		
4	2		
5	3		
6	2		
7	2		
8	3		
9	2		
10	3		
11	2		
12	2		
13	2		
14	2		
15	2		
16	2		
17	2		
18	2		
19			
20			
21			
22			
23			
24			

	1	2	3
1	10		
2	10		
3	2		
4	10		
5	2		
6	2		
7	2		
8	2		
9	2		
10	2		
11	2		
12	2		
13	2		
14	2		
15	0		
16	1		
17	8		
18	2		
19	10		
20	2		
21	2		
22	10		
23	2		
24	2		

รูปที่ 4.19 ข้อมูล Pattern ที่ได้จากการวิเคราะห์เส้นตรงภาพซ้ายที่อาคาร TNI ชั้น 6 และภาพขวาที่อาคาร TNI ชั้น 5

floorFe.	turnD	stepCount	floorFe.	turnD	stepCount
13x1 double	13x1 double	13x1 double	14x1 double	14x1 double	14x1 double
1	1	1	1	1	1
2	1	0	2	1	0
3	1	0	3	1	0
4	1	0	4	1	0
5	1	0	5	1	0
6	1	0	6	1	0
7	1	0	7	1	0
8	1	0	8	1	0
9	1	0	9	1	0
10	1	0	10	1	0
11	1	0	11	1	0
12	1	0	12	1	0
13	1	0	13	1	0

(ก)

(ข)

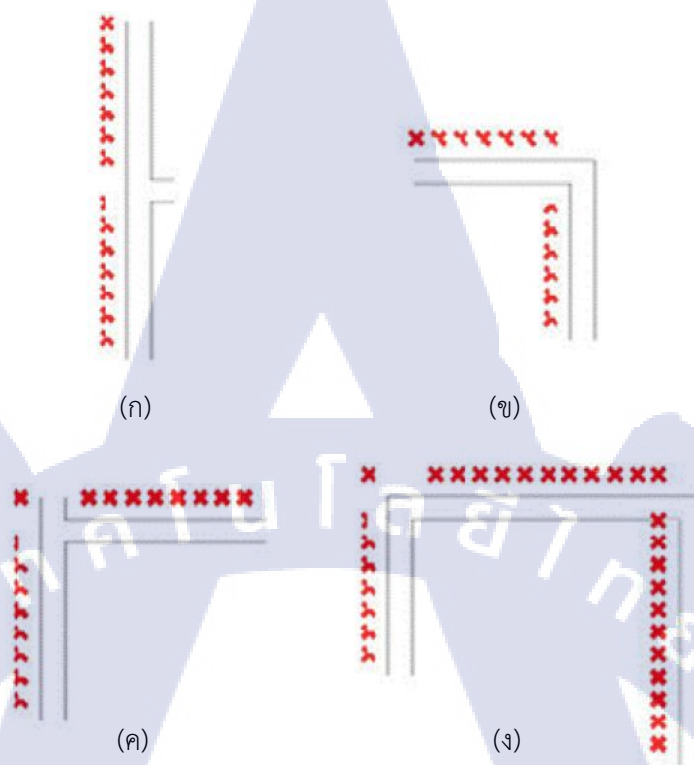
รูปที่ 4.20 ข้อมูล Pattern ที่ได้จากการวิเคราะห์เส้นตรงที่ (ก) อาคาร Sino Thai และ (ข) อาคาร LPN

floorFe.	turnD	stepCount	floorFe.	turnD	stepCount
18x1 double	18x1 double	18x1 double	24x1 double	24x1 double	24x1 double
1	1	1	1	1	1
2	1	0	2	1	0
3	3	0	3	1	0
4	2	0	4	1	0
5	3	0	5	1	0
6	2	0	6	1	0
7	3	0	7	1	0
8	2	0	8	1	0
9	2	0	9	1	0
10	3	0	10	1	0
11	2	0	11	1	0
12	2	0	12	1	0
13	2	0	13	1	0
14	2	0	14	1	0
15	2	0	15	1	0
16	2	0	16	1	0
17	2	0	17	1	0
18	2	0	18	1	0

(ก)

(ข)

รูปที่ 4.21 ข้อมูล Pattern ที่ได้จากการวิเคราะห์เส้นตรงที่อาคาร TNI (ก) ชั้น 6 และ (ข) ชั้น 5



รูปที่ 4.22 ผลลัพธ์สุดท้ายของแผนผังการเดิน

การบันทึกผลการทดลอง

หลังจากรวบรวมข้อมูลและประมวลผลสำหรับการทดลองใน 3 สถานที่ สถานที่ละ 3 เส้นทาง และเส้นทางที่เป็นการเลี้ยวสองครั้งอีก 1 เส้นทาง รวมทั้งหมด 58 ครั้ง ได้ผลลัพธ์แผนผังการเดินทั้งหมดตามรูปในภาคผนวก ค.และข้อมูลระยะทางการเดินตามตารางที่ 4.3 โดยในแต่ละช่องมีที่มาดังนี้

1. Total Distance คือระยะทางจริงของเส้นทางแต่ละเส้น โดยที่ระยะทางนี้จะเท่ากันทั้งขาไปและกลับมีหน่วยเป็นเซนติเมตร
2. System Count คือจำนวนก้าวเฉลี่ยของแต่ละเส้นทางที่สามารถตรวจจับได้จากระบบของการเดินในเส้นทางนั้นๆ มีหน่วยเป็นจำนวนก้าว
3. Calculated Distance คือระยะทางเฉลี่ยที่คำนวณได้จากการเดินในเส้นทางนั้น คำนวณได้จากการนำจำนวนก้าวในช่อง System Count มาคูณกับระยะก้าว ซึ่งการเดินครั้งนี้ผู้ใช้งานสูง 180 เซนติเมตร จึงทำให้คำนวณระยะก้าวได้เท่ากับ 75 เซนติเมตรต่อหนึ่งก้าว มีหน่วยเป็นเซนติเมตร

4. Distance Error คือเปอร์เซ็นต์ที่คลาดเคลื่อนของระยะทางที่คำนวณได้ (Calculated Distance) เทียบกับระยะทางจริง (Total Distance)

ตารางที่ 4.3 ผลการทดลองจากการเดินในเส้นทาง 10 เส้นทางใน 4 สถานที่

สถานที่ที่ 1 : Sino Thai ชั้น 10

	Total Distance (cm.)	System Count (Step)	Calculated Distance (cm.)	Distance Error (%)
เส้นทางที่ 1	1320	17.0	1275	3.41
เส้นทางที่ 2	1170	15.0	1125	3.85
เส้นทางที่ 3	1050	13.0	975	7.14

สถานที่ที่ 2 : LPN ทำพระ อาคาร A ชั้น 9

	Total Distance (cm.)	System Count (Step)	Calculated Distance (cm.)	Distance Error (%)
เส้นทางที่ 1	960	12.2	915.00	4.69
เส้นทางที่ 2	1050	14.6	1092.86	4.08
เส้นทางที่ 3	930	13.3	996.43	7.14

สถานที่ที่ 3 : สถาบันเทคโนโลยีไทย – ญี่ปุ่น อาคาร A ชั้น 6

	Total Distance (cm.)	System Count (Step)	Calculated Distance (cm.)	Distance Error (%)
เส้นทางที่ 1	1170	17.6	1317.86	12.64
เส้นทางที่ 2	900	14.0	1050.00	16.67
เส้นทางที่ 3	930	14.4	1080.00	16.13

ตารางที่ 4.3 ผลการทดลองจากการเดินในเส้นทาง 10 เส้นทางใน 4 สถานที่ (ต่อ)

สถานที่ที่ 4 : สถาบันเทคโนโลยีไทย – ญี่ปุ่น อาคาร A ชั้น 5 การเดินแบบเลี้ยวสองครั้ง

	Total Distance (cm.)	System Count (Step)	Calculated Distance (cm.)	Distance Error (%)
เส้นทางที่ 4	2130	27.6	2070	2.82

จากผลการทดลองพบว่าผลลัพธ์ของแผนผังการเดินในเส้นทางเดียวกันจะมีลักษณะใกล้เคียงกันทั้งขาไปและขากลับโดยความแตกต่างของการนับจำนวนก้าวในแต่ละเส้นทางซึ่งนับได้จากสัญลักษณ์กากบาทที่แทนการก้าว 1 ก้าว แตกต่างกันไม่เกิน 2 ช่องหรือ 2 ก้าว

ความถูกต้องของข้อมูลทิศทางการเดินอยู่ที่ 100% คือสามารถระบุการเดินในทางตรงหรือการเลี้ยวซ้ายหรือเลี้ยวขวาได้อย่างถูกต้องซึ่งเป็นผลของข้อมูลจาก Gyroscope โดยพบว่าการเลี้ยวที่สมบูรณ์จะใช้เวลาประมาณ 2 วินาทีต่อการเลี้ยว 1 ครั้ง

ความถูกต้องของการระบุรูปแบบพื้นทางเดินซึ่งแบ่งหมวดของการระบุพื้นทางเดินได้เป็น 2 กลุ่ม ซึ่งไม่นับการเดินทางตรงที่ไม่มีทางแยกได้เป็น

1. การตรวจพบทางแยกระหว่างการเดินในทางตรง พบทางแยก 1 ครั้งจากการเดิน 14 ครั้ง คิดเป็น 7.14%

2. การตรวจพบทางแยกในช่วงการเลี้ยว ในส่วนนี้แบ่งได้อีก 2 ส่วน คือ

2.1 พบทางเดินตรงที่ถัดจากทางแยกพบทางเดินตรง 19 ครั้ง การจากเดิน 26 ครั้ง คิดเป็น 73.08%

2.2 การระบุทางแยกอีกด้านหนึ่งที่ไม่อยู่ในเส้นทางเดิน พบทางแยก 1 ครั้งจากการเดิน 24 ครั้ง คิดเป็น 4.17%

สำหรับการคำนวณระยะทางการเดิน จากตารางที่ 4.3 เมื่อนำค่า Distance Error ที่เป็นค่าความคลาดเคลื่อนจากระยะทางการเดินที่คำนวณได้เปรียบเทียบกับระยะทางการเดินจริงมาเฉลี่ยในทุกการเดิน ได้ค่าคลาดเคลื่อนเฉลี่ยอยู่ที่ 7.86% หรือถ้าเทียบเป็นระยะทางที่คลาดเคลื่อนเฉลี่ยที่ 75.19 ซม. ซึ่งถ้านำเลขจำนวนนี้ไปหาจำนวนก้าวที่เกิดขึ้นกับผู้ใช้จะได้ค่าคลาดเคลื่อนเฉลี่ยเป็นจำนวน 1 ก้าว

บทที่ 5

การวิเคราะห์ผลการทดลอง สรุปผลงานวิจัย และข้อเสนอแนะ

วิเคราะห์ผลการทดลอง และสรุปผลงานวิจัย

การวิเคราะห์ผลการทดลองแบ่งออกเป็น 4 ด้าน คือ วิเคราะห์การเก็บข้อมูลการเดิน วิเคราะห์การระบุรูปแบบพื้นทางเดิน วิเคราะห์การสร้างแผนผังการเดิน และวิเคราะห์การคำนวณระยะทางการเดิน

ด้านการเก็บข้อมูลการเดินพบว่าระบบทำงานได้อย่างมีประสิทธิภาพและสามารถเก็บข้อมูลได้อย่างต่อเนื่อง ภาพที่ได้มาจากการถ่ายวิดีโออย่างเพียงพอและสามารถทำให้เห็นภาพรวมของภาพที่เห็นในระหว่างการเดินได้ ถึงแม้ว่าจะใช้งานภาพเพียง 1 ภาพต่อ 1 วินาทีก็ตาม รวมถึงเซ็นเซอร์ที่นำมาใช้งานคือ Step Detector และ Gyroscope สามารถทำงานได้อย่างมีประสิทธิภาพ โดยที่ความคลาดเคลื่อนที่เกิดขึ้นจาก Step Detector อยู่ที่ประมาณ 1 ก้าวจากการทดสอบในบทที่ 4 และความคลาดเคลื่อนที่เกิดขึ้นในแผนผังการเดินที่เป็นผลลัพธ์อยู่ที่ประมาณ 2 ก้าว โดยความคลาดเคลื่อนที่น่าจะเกิดบริเวณทางแยกที่มีการเปลี่ยนทิศทางการเดินและจุดสิ้นสุดซึ่งทั้งสองจุดมีการสับเท้าที่อยู่ด้านหลังไปบรรจบกับเท้าด้านหน้าก่อนทำการเลี้ยวหรือทำการหยุดการเดินหรือการเดินที่ไม่เต็มก้าวที่บริเวณดังกล่าวเพื่อไม่ให้เดินเกินเส้นทางการเดินที่กำหนดไว้ จึงทำให้บางครั้งจำนวนก้าวคลาดเคลื่อนไป ส่วนการทำงานของ Gyroscope ปกติเซ็นเซอร์ชนิดนี้จะทำการส่งค่าให้กับผู้ใช้งานตลอดเวลา ดังนั้นประสิทธิภาพของ Gyroscope จึงอยู่ที่การตั้งค่า Threshold เป็นหลักโดยที่งานวิจัยนี้ใช้ค่า 1.0 เป็นค่า Threshold เพื่อกรองข้อมูลการหมุนเพียงเล็กน้อยที่อยู่ระหว่าง -1.0 ถึง 1.0 rad/s ออกและไม่ทำการบันทึกข้อมูล ซึ่งสำหรับการทดลองในสถานที่อื่นหรือผู้ทำการทดลองท่านอื่นอาจต้องทำการทดสอบเพื่อทำการหาค่า Threshold ที่เหมาะสำหรับการทดลองนั้นเพื่อให้ระบบบันทึกผลได้อย่างถูกต้องเพื่อให้การเดินทางตรงไม่ตรวจจับเจอการเลี้ยวและในช่วงการเลี้ยวระบบต้องตรวจจับการเลี้ยวอย่างต่อเนื่อง

ด้านการระบุรูปแบบพื้นทางเดินซึ่งเป็นส่วนที่วิเคราะห์ลักษณะของภาพทางเดินจากมุมมอง Perspective เพื่อทำการจับคู่กับรูปแผนผังทางเดินที่เป็นมุมมอง Top View ซึ่งได้นำกระบวนการทาง Image Processing ในการค้นหาและวิเคราะห์ลักษณะเส้นจากภาพแต่ในงานวิจัยนี้ยังมีความแม่นยำที่ได้ไม่สูงนักสำหรับการตรวจจับทางเดินตรงที่ถัดไปจากทางแยกและความแม่นยำต่ำในการตรวจจับทางเลี้ยวด้านอื่นนอกเส้นแนวการเดิน โดยน่าจะมีสาเหตุหลายประการ เช่น ภาพถ่ายในอาคารที่ได้แสงไม่เพียงพอต่อการแยกแยะเส้นขอบของกำแพงที่เป็นทางเลี้ยว การเดินที่ทำให้ภาพถ่ายจากวิดีโอบางภาพเบลอจึงทำให้เส้นขอบที่ได้ไม่ชัดเจนและยากสำหรับการตรวจหาเส้นตรงหรือลักษณะของ

กำแพงที่ทางแยกซึ่งเป็นกำแพงสีเดียวกันแต่มีความแตกต่างในด้านความลึกจึงแยกแยะและหาเส้นขอบในแนวตั้งได้ยากเมื่อประมวลผลผ่านทาง Image Processing แต่สามารถระบุได้ง่ายโดยมนุษย์

ด้านการสร้างแผนผังการเดินทางกระบวนกรจัดรูปแบบข้อมูลก่อนการสร้างแผนผังมีความสำคัญเป็นอย่างมาก เนื่องจากข้อมูลที่ได้รับมาจากสมาร์โฟนไม่สามารถใช้งานได้ทันที แต่ถ้าจัดข้อมูลอย่างมีระเบียบแบบแผนจะทำให้การสร้างแผนผังการเดินทางเป็นไปอย่างมีระเบียบและทำการปรับปรุงแก้ไขกระบวนกรการสร้างแผนผังการเดินทางได้ง่ายอีกด้วย

ด้านการคำนวณระยะทางการเดิน ที่คำนวณจากจำนวนก้าวและระยะก้าวนั้นเมื่อนำมาเปรียบเทียบกับระยะทางการเดินจริงแล้วพบว่า คลาดเคลื่อนที่ประมาณ 75 ซม หรือประมาณเท่ากับ 1 ก้าวของการเดินปกติ ซึ่งระยะทางที่เกิดขึ้นไม่ส่งผลต่อความเข้าใจในแผนผังการเดินทางที่เป็นผลลัพธ์ เพราะลักษณะการใช้งานแผนที่ในอาคารมักใช้ในการบอกตำแหน่งและระยะทางโดยคร่าวเพื่อให้ผู้อ่านเข้าใจทางเดินในอาคารและสามารถเดินไปยังจุดหมายที่ต้องการได้ โดยที่ไม่ต้องการความแม่นยำของระยะทางในแต่ละช่วงของการเดินมากนัก เพียงแค่คงสัดส่วนของระยะทางให้อยู่ในสัดส่วนเดียวกัน ก็ทำให้สามารถใช้แผนที่ในอาคารขั้นนั้นได้

สำหรับการทดลองที่ TNI อาคาร A ชั้น 6 ทั้ง 3 เส้นทางที่มีค่าความคลาดเคลื่อนผิดไปมากจากการเดินในสถานที่อื่นอย่างมีนัยสำคัญจะมีสาเหตุจากการเดินที่ผิดปกติไปจากการเดินทำการทดลองวันอื่นเพราะการเดินของมนุษย์นั้นไม่สามารถกำหนดระยะการเดินได้แม่นยำในทุกก้าวจึงมีความเป็นไปได้ที่จะได้ระยะก้าวที่มากกว่าหรือน้อยกว่าค่าที่ได้จากการคำนวณถึงแม้ว่าเราจะไม่สามารถกำหนดระยะก้าวให้แม่นยำได้ในทุกครั้ง แต่โดยทั่วไปแล้วการเดินในแบบปกติของช่วงเวลาเดียวกันมักคงที่ซึ่งจากการสังเกตแผนผังการเดินทางที่เป็นผลลัพธ์นั้นยังมีสัดส่วนระยะทางการเดินสอดคล้องกันแสดงให้เห็นว่าถึงแม้ว่าการเดินในสถานที่ที่ 3 จะผิดปกติไปจากการเดินครั้งอื่น แต่ถ้าทำให้การเดินในแต่ละช่วงเวลาสม่ำเสมอได้แล้วนั้นถึงแม้ว่าระยะทางจริงจะมีความคลาดเคลื่อนไปแต่ภาพผลลัพธ์ของแผนผังการเดินทางมีสัดส่วนเดิมและเมื่อนำมาเฉลี่ยกับการเดินในทุกช่วงเวลาแล้วความคลาดเคลื่อนอยู่ที่ประมาณ 7% หรือประมาณ 1 ก้าวตามที่สรุปไว้สรุปของการทดลอง ซึ่งสามารถสรุปได้ว่าถึงแม้ว่าจะมีการเดินที่ไม่เป็นปกติในบางครั้งก็ไม่ส่งผลทำให้แผนผังการเดินทางผิดสัดส่วนไปจากการเดินจริง

ดังนั้นจากผลการทดลองและการวิเคราะห์ผลการทดลองสามารถสรุปผลงานวิจัยได้ว่าระบบที่พัฒนาขึ้นนี้สามารถทำการเก็บข้อมูลการเดินทางในอาคารด้วยสมาร์โฟนแล้วทำการส่งต่อข้อมูลการเดินทางประมวลผลไปยังคอมพิวเตอร์เพื่อสร้างแผนผังการเดินทางได้โดยที่ผลลัพธ์แผนผังการเดินทางที่ได้เป็นไปตามข้อมูลที่ทำกรจัดเก็บมีทิศทางการเดินที่ถูกต้องและสัดส่วนระยะทางของแผนผังที่เป็นผลลัพธ์ใกล้เคียงกับการเดินจริง โดยไม่จำเป็นต้องใช้ๆ ข้อมูลเพิ่มเติมจากอุปกรณ์อื่นได้

ข้อเสนอแนะ

ข้อเสนอแนะสำหรับงานวิจัยนี้มีด้วยกัน 5 ประเด็น คือ ประเด็นด้านความแม่นยำของการคำนวณระยะทางในการเดิน ประเด็นด้านความสามารถในการวิเคราะห์รูปแบบพื้นทางเดินประเด็นด้านการนำไปใช้งานร่วมกับระบบที่เกี่ยวข้องประเด็นด้านการนำข้อมูลที่เก็บได้ไปใช้ประโยชน์ในด้านอื่นและการปรับปรุงระบบให้สามารถใช้งานบนอุปกรณ์สวมใส่

ประเด็นด้านความแม่นยำของการคำนวณระยะทางการเดินนั้น เนื่องจากงานวิจัยนี้ใช้ระยะก้าวการเดินซึ่งมักมีระยะการเดินแต่ละก้าวไม่เท่ากัน จึงมีผลกระทบต่อการคำนวณเพราะการได้มาของระยะทางการเดินนั้น เกิดจากการนำจำนวนการก้าวที่เกิดขึ้นมาคูณกับระยะก้าวของผู้ใช้ แต่สิ่งที่สามารถพัฒนาขีดความสามารถในการทำงานได้คือเซ็นเซอร์ที่ใช้ในการตรวจจับการเดิน ซึ่งถ้าเซ็นเซอร์ยังมีความแม่นยำในการตรวจจับมากเท่าใด ระยะทางที่คลาดเคลื่อนก็น้อยลงเท่านั้น สำหรับเรื่องระยะการก้าวของผู้ใช้งาน ยังคงจำเป็นต้องใช้การควบคุมและฝึกฝนการเดินของผู้ใช้เพื่อให้ระยะก้าวของผู้ใช้สม่ำเสมอและคงที่ใกล้เคียงกับการคำนวณ หรือทำการพัฒนาวิธีการวัดระยะทางการเดินด้วยวิธีอื่นแทนการใช้ระยะก้าว และจำนวนก้าวในการคำนวณ

ประเด็นด้านความสามารถในการวิเคราะห์รูปแบบพื้นทางเดิน เนื่องจากงานวิจัยนี้เน้นการสร้างแผนผังทางเดิน ดังนั้นทางเดินจะอยู่ในรูปแบบทางเดินที่ต้องมีกำแพงอยู่ทั้ง 2 ด้าน และกำแพงเป็นเส้นตรง จึงเกิดข้อจำกัดในการนำระบบไปใช้งานจริง ฉะนั้นถ้าสามารถวิเคราะห์รูปแบบพื้นทางเดินเพิ่มเติม เพิ่มความสามารถในการแยกและคัดเลือกสิ่งรบกวนหรือ Noise ออกจากภาพก่อนการประมวลผล และปรับปรุงการสร้างแผนผังทางเดิน จะทำให้ระบบสามารถนำไปใช้งานได้หลากหลายพื้นที่มากขึ้น เช่น บริเวณที่เป็นโถงกว้าง หรือ บริเวณทางแยกที่ไม่เป็นมุมฉากกับทางแยก หรือทางเดินที่เป็นวงกลม เป็นต้น

ประเด็นด้านการนำไปใช้งานร่วมกับระบบที่เกี่ยวข้อง ที่กล่าวถึงในบทนำมีงานวิจัยมากมายที่เนื้อหาเกี่ยวกับระบบนำทางๆ หรือระบบแนะนำเส้นทางในอาคาร แต่ยังไม่มียุคสร้างแผนผังในอาคาร ดังนั้นมีการพัฒนาระบบนี้และนำไปต่อเชื่อมกับระบบนำทาง หรือ ระบบแนะนำเส้นทาง และสามารถเชื่อมต่อข้อมูลผู้ใช้หลายคนเข้าไว้ด้วยกันได้ ก็จะสามารถแบ่งปันฐานข้อมูลการเดินร่วมกันได้

ประเด็นด้านการนำข้อมูลที่เก็บได้ไปใช้ประโยชน์ด้านอื่น งานวิจัยนี้มีพื้นฐานจากการสร้างแผนผังเพื่อนำไปใช้งานกับระบบนำทางในอาคาร หรือระบบแนะนำเส้นทางในอาคาร ดังนั้น ภาพที่ได้จากการถ่ายวิดีโออันถือเป็นข้อมูลสำคัญที่ช่วยในการระบุสถานที่ภายในอาคารได้เป็นอย่างดี เช่น ป้ายบอกทาง ชื่อสถานที่ที่อยู่ในภาพ หรือ วัตถุที่มีลักษณะเด่น และนำมาใช้ในการค้นหาได้

ประเด็นด้านการปรับปรุงระบบสามารถใช้งานบนอุปกรณ์สวมใส่ได้ทั้งระบบยกตัวอย่าง เช่น Google Glass หรืออุปกรณ์สวมใส่ที่บริเวณสายตาอื่นๆ ซึ่งนอกจากจะทำให้การสร้างแผนผัง การเดินง่ายขึ้นเนื่องจากไม่จำเป็นต้องขนย้ายข้อมูลไปที่เครื่องคอมพิวเตอร์เพราะสามารถสร้างแผนผัง ได้ด้วยตัวเองแล้ว ยังเอื้อต่อการสร้างแผนผังการเดินแบบเรียบใหม่อีกด้วย

ซึ่งจะเป็นผลให้การนำไปต่อเชื่อมกับระบบอื่น เช่น ระบบช่วยเหลือการเดิน ระบบนำ ทางการเดิน เป็นไปได้อย่างมีประสิทธิภาพอีกด้วย



TNIT

บรรณานุกรม

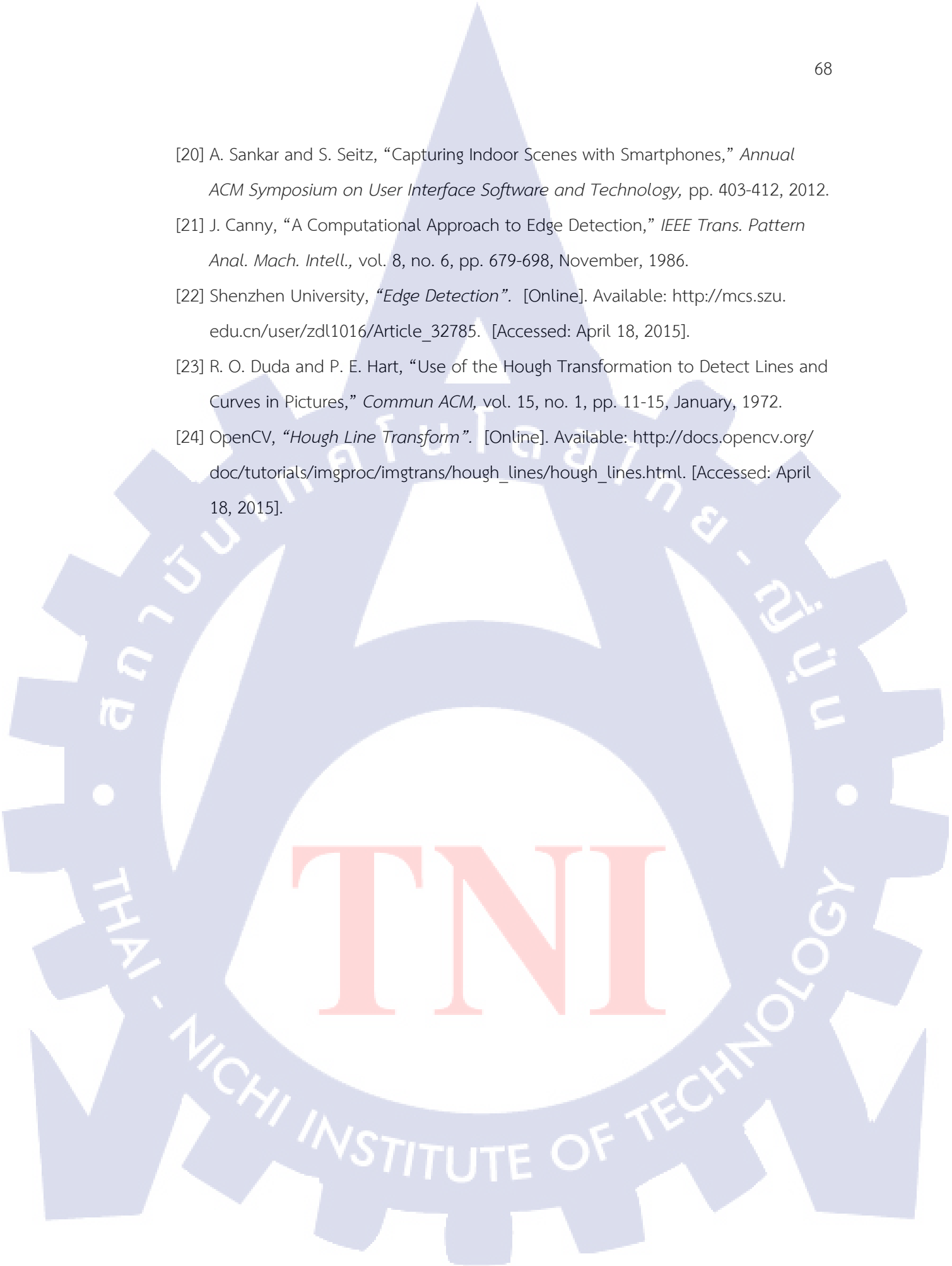
TNI

บรรณานุกรม

- [1] A. C. Madrigal, “How Google Builds Its Maps-and What It Means for the Future of Everything”. [Online]. Available: <http://www.theatlantic.com/technology/archive/2012/09/how-google-builds-its-maps-and-what-it-means-for-the-future-of-everything/261913/>. [Accessed: January 2, 2015].
- [2] Google, “Google Maps”. [Online]. Available: <https://www.google.co.th/maps/@13.6053584,100.4133405,12z?hl=en>. [Accessed: January 2, 2015].
- [3] H. Rao and W. Fu, “A General Framework for a Collaborative Mobile Indoor Navigation Assistance System”. [Online]. Available: http://www.dfki.de/LAMDa/accepted/13_HuamingRao.pdf. [Accessed: July 17, 2014].
- [4] L. Ran et al, “Drishti: An Integrated Indoor/Outdoor Blind Navigation System and Service,” *IEEE Annual Conference on Pervasive Computing and Communications*, pp. 23–30, 2004.
- [5] E. Wise et al, “Indoor Navigation for the Blind and Vision Impaired: Where are we and Where are we Going?,” *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1-7, 2012.
- [6] N. Fallah et al, “The User As a Sensor: Navigating Users with Visual Impairments in Indoor Spaces Using Tactile Landmarks,” *SIGCHI Conference on Human Factors in Computing Systems*, pp. 425–432, 2012.
- [7] M. Nozawa et al, “Indoor Human Navigation System on Smartphones Using View-based navigation,” *International Conference on Control, Automation and Systems (ICCAS)*, pp. 1916–1919, 2012.
- [8] M. Werner et al, “Indoor Positioning Using Smartphone Camera,” *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1-6, 2011.
- [9] H. Kawaji et al, “Image-Based Indoor Positioning System : Fast Image Matching Using Omnidirectional Panoramic Images,” *ACM International Workshop on Multimodal Pervasive Video Analysis*, pp. 1–4, 2010.

- [10] Google, “*Introduction to Android | Android Developers*”. [Online]. Available: <http://developer.android.com/guide/index.html>. [Accessed : January 3, 2015].
- [11] Google, “*Android KitKat | Android Developers*”. [Online]. Available: <https://developer.android.com/about/versions/kitkat.html>. [Accessed: December 28, 2014].
- [12] Google, “*Sensors Overview*”. [Online]. Available: http://developer.android.com/guide/topics/sensors/sensors_overview.html. [Accessed: April 18, 2015].
- [13] Y. Li and S. T. Birchfield, “Image-Based Segmentation of Indoor Corridor Floors for a Mobile Robot,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 837-843, 2010.
- [14] G. C. Barcelo et al, “Image-Based Floor Segmentation in Visual Inertial Navigation,” *Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1402-1407, 2013.
- [15] GPS.gov, “*GPS.gov: GPS Accuracy*”. [Online]. Available: <http://www.gps.gov/systems/gps/performance/accuracy/>. [Accessed: December 28, 2014].
- [16] W.-Y. Shih et al, “Estimating Walking Distance with a Smart Phone,” *International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*, pp. 166-171, 2012.
- [17] J.-S. Wang et al, “Walking Pattern Classification and Walking Distance Estimation Algorithms Using Gait Phase Information,” *IEEE Trans. Biomed. Eng.*, vol. 59, no. 10, pp. 2884-2892, October, 2012.
- [18] J. Goodwin, “*Touch & Movement: Palpation and Kinesiology for Massage Therapists. Cengage Learning*”. [Online]. Available: <https://books.google.co.th/books?id=SRwKAAAOBAJ&pg=PA121&lpg=PA121#v=onepage&q&f=false>. [Accessed: February 1, 2014].
- [19] H. Shin et al, “Unsupervised Construction of an Indoor Floor Plan Using a Smartphone,” *IEEE Trans. Syst. Man Cybern.*, vol. 42, no. 6, pp. 889-898, November, 2012.

- [20] A. Sankar and S. Seitz, "Capturing Indoor Scenes with Smartphones," *Annual ACM Symposium on User Interface Software and Technology*, pp. 403-412, 2012.
- [21] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679-698, November, 1986.
- [22] Shenzhen University, "Edge Detection". [Online]. Available: http://mcs.szu.edu.cn/user/zdl1016/Article_32785. [Accessed: April 18, 2015].
- [23] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Commun ACM*, vol. 15, no. 1, pp. 11-15, January, 1972.
- [24] OpenCV, "Hough Line Transform". [Online]. Available: http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html. [Accessed: April 18, 2015].



TNI



ภาคผนวก

ภาคผนวก ก.

ซอร์สโค้ดโปรแกรมบนสมาร์ทโฟน Android

TNI

ซอร์สโค้ดโปรแกรมบนสมาร์ทโฟน Android

ส่วนของซอร์สโค้ดโปรแกรมบนสมาร์ทโฟน Android มีโครงสร้าง ดังนี้



รูปที่ ก.1 โครงสร้าง Code ของระบบ

1. CounterActivity.java ซอร์สโค้ดหลักในการทำงานของโปรแกรมบน Android

```
package com.starboardland.pedometer;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.GregorianCalendar;
import java.util.StringTokenizer;
import android.app.Activity;
```



```

import android.app.AlertDialog;
import android.content.Context;
import android.content.pm.ActivityInfo;
import android.hardware.*;
import android.hardware.Camera.PreviewCallback;
import android.media.CamcorderProfile;
import android.media.MediaRecorder;
import android.media.MediaScannerConnection;
import android.os.Bundle;
import android.os.Environment;
import android.transition.Visibility;
import android.view.View;
import android.widget.Button;
import android.widget.FrameLayout;
import android.widget.TextView;
import android.widget.Toast;

public class CounterActivity extends Activity implements SensorEventListener {
    private SensorManager sensorManager;
    private TextView step,turn, distance, stepLength, gyro;
    private Button pause;
    boolean activityRunning;
    private int stepCount = 0;
    private File currentSavePath;
    String newLineString = "\r\n";
    SimpleDateFormat s = new SimpleDateFormat("yyyyMMdd-HH:mm:ss");
    StringBuffer sbStep, sbTurn;
    private Camera mCamera;
    private CameraPreview mPreview;
    public static final int MEDIA_TYPE_VIDEO = 2;
    final int LANDSCAPE_GYRO_VALUE = 0;
    final int PORTRAIT_GYRO_VALUE = 1;
    private boolean isRecording = false;
    private MediaRecorder mMediaRecorder;

```

```

private boolean isPause = false;
private File mVideo;

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_USER_LANDSCAPE);
    setContentView(R.layout.main);

    stepLength = (TextView) findViewById(R.id.stepLength);
    step = (TextView) findViewById(R.id.step);
    turn = (TextView) findViewById(R.id.turn);
    distance = (TextView) findViewById(R.id.distance);
    gyro = (TextView) findViewById(R.id.gyro);
    pause = (Button) findViewById(R.id.pause);

    sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);

    sbStep = new StringBuffer();
    sbTurn = new StringBuffer();

    preparePreview();
}

@Override
protected void onResume() {
    super.onResume();
    registerSensorServices();
}

/*****
* Button Events
*****/

public void onStart(View view){

```

```

registerSensorServices();

currentSavePath = new File(Environment.getExternalStorageDirectory() +
"/FPC/"+s.format(new Date()));

// /storage/emulated/0/Download/FPC
if (!currentSavePath.exists()){
    if(!currentSavePath.mkdirs()){
        System.out.println("folder does not created");
    }else {
        System.out.println("created folder");
    }
}

System.out.println(currentSavePath.getPath());
startVideoRecord();
sbTurn.append("=",s.format(new Date()));
sbTurn.append("|");
sbStep.append("=,=",s.format(new Date()));
sbStep.append("|");
}

public void onPause(View view){

    if (isPause){
        pause.setText("Pau");
        registerSensorServices();
        isPause = false;
    }else{
        pause.setText("Resu");
        unregisterSensorServices();
        isPause = true;
    }
}

public void onStop(View view){

```

```

unregisterSensorServices();
stopVideoRecord();

saveTurnInformation();
saveStepInformation();
}
public void onButtonReset(View view){
    resetValue();
}
public void onButtonHide(View view){
    mPreview.setVisibility(Visibility.MODE_OUT);
}

/*****
 * Sensors Registration & Un-registration
 *****/

private void registerSensorServices() {

    activityRunning = true;
    Sensor stepDetector = sensorManager.getDefaultSensor(Sensor.
TYPE_STEP_DETECTOR);
    Sensor gyro = sensorManager.getDefaultSensor(Sensor.TYPE_GYROSCOPE);

    if (stepDetector != null) {
        sensorManager.registerListener(this, stepDetector, Sensor
Manager.SENSOR_DELAY_FASTEST);
    }

    if (gyro != null) {
        sensorManager.registerListener(this, gyro, SensorManager.
SENSOR_DELAY_FASTEST);
    }

    if (stepDetector == null || gyro == null) {

```

```

        Toast.makeText(this, "some sensors not
        available!",Toast.LENGTH_LONG).show();
    }
}

private void unregisterSensorServices(){
    activityRunning = false;

    Sensor stepDetector = sensorManager.getDefaultSensor
    (Sensor.TYPE_STEP_DETECTOR);

    Sensor gyro = sensorManager.getDefaultSensor(Sensor.TYPE_GYROSCOPE);

    if (stepDetector != null) {
        sensorManager.unregisterListener(this, stepDetector);
    }

    if (gyro != null){
        sensorManager.unregisterListener(this, gyro);
    }
}

/*****
 * Sensor Override Functions
 *****/

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) { }

@Override
public void onSensorChanged(SensorEvent event) {
    if (activityRunning) {
        if(event.sensor.getType() == Sensor.TYPE_STEP_DETECTOR){
            stepCount++;
            step.setText(String.valueOf(stepCount));

            int walkdistance = Integer.parseInt(String.valueOf(stepLength.getText()))
            *Integer.parseInt(String.valueOf(step.getText()));

```



```

distance.setText(String.valueOf(walkdistance));
if (isRecording){
    sbStep.append(stepCount);
    sbStep.append(",");
    sbStep.append(walkdistance);
    sbStep.append(",");
    sbStep.append(s.format(new Date()));
    sbStep.append("||");
}
}else if(event.sensor.getType() == Sensor.TYPE_GYROSCOPE){
    float threshold = 0;
    try{
        threshold = Float.parseFloat(String.valueOf
(gyro.getText()));
    } catch (NumberFormatException ex){
        // do nothing
    }

    float minusThreshold = 0-threshold;
    if (event.values[LANDSCAPE_GYRO_VALUE]<minusThreshold){
        turn.setText("RIGHT");

        if (isRecording){
            sbTurn.append("-1,");
            sbTurn.append(s.format(new Date()));
            sbTurn.append("||");
        }
    }else if (event.values[LANDSCAPE_GYRO_VALUE]>threshold){
        turn.setText("LEFT");

        if (isRecording){
            sbTurn.append("1,");
            sbTurn.append(s.format(new Date()));
            sbTurn.append("||");
        }
    }
}
}

```

```

        }else{
            turn.setText("");
        }
    }

}

}

}

/*****
* General Functions
*****/

private void resetValue(){
    step.setText("0");
    stepCount = 0;
    distance.setText("0");
    sbStep = new StringBuffer();
    sbTurn = new StringBuffer();
}

private void refreshMTPFile(File file){
    MediaScannerConnection.scanFile(this, new String[] { file.getAbsolutePath() }, null, null);
}

private File getOutputMediaFile(int type){
    File mediaFile;
    if(type == MEDIA_TYPE_VIDEO) {
        mediaFile = new File(currentSavePath, "VID.mp4");
    } else {
        return null;
    }

    System.out.println("output file at : " + mediaFile);
    refreshMTPFile(mediaFile);

    return mediaFile;
}

```

```

private void preparePreview(){
// Create an instance of Camera
    mCamera = getCameraInstance();

    // Create our Preview view and set it as the content of our activity.
    mPreview = new CameraPreview(this, mCamera);
    FrameLayout preview = (FrameLayout) findViewById(R.id.myFrame);
    preview.addView(mPreview);
    System.out.println("step:set preview object");
}
private void saveTurnInformation(){
    System.out.println(currentSavePath.getPath());
    FileOutputStream fos,fosr;
    StringTokenizer st;

    // Save step information into file
    File fTurn = new File(currentSavePath,"turn.txt");
    File fTurnRaw = new File(currentSavePath,"turn_raw.txt");

    StringBuffer sbt = new StringBuffer();
    StringBuffer sbtr = new StringBuffer();

    // Save turn information into file
    try {
        fos = new FileOutputStream(fTurn);
        fosr = new FileOutputStream(fTurnRaw);

        st = new StringTokenizer(sbTurn.toString(), "|");

        // Get filter and revise information to easy to import by MATLAB's
        // dmlread function
        sbt = filterTurnInformation(sbTurn.toString());

        // Put real information for backup and replace || with \r\n to easy to
        read
    }

```

```

        while (st.hasMoreTokens()){
            sbtr.append(st.nextToken());
            sbtr.append(newLineString);
        }

        fos.write(sbt.toString().getBytes());
        fosr.write(sbtr.toString().getBytes());

        fos.close();
        fosr.close();

        refreshMTPFile(fTurn);
        refreshMTPFile(fTurnRaw);
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

private void saveStepInformation(){
    System.out.println(currentSavePath.getPath());
    FileOutputStream fos, fosr;

    // Save step information into file
    File fStep = new File(currentSavePath,"step.txt");
    File fStepRaw = new File (currentSavePath,"step_raw.txt");

    StringBuffer sbs = new StringBuffer();
    StringBuffer sbsr = new StringBuffer();

    StringTokenizer st = new StringTokenizer(sbStep.toString(), "|");

    try {
        fos = new FileOutputStream(fStep);
        fosr = new FileOutputStream(fStepRaw);

```

```

function
    // Get filter and revise information to easy to import by MATLAB's dmlread

    sbs = filterStepInformation(sbStep.toString());

    while (st.hasMoreTokens()){
        sbsr.append(st.nextToken());
        sbsr.append(newLineString);
    }

    fos.write(sbs.toString().getBytes());
    fosr.write(sbsr.toString().getBytes());
    fos.close();
    fosr.close();

    refreshMTPFile(fStep);
    refreshMTPFile(fStepRaw);
    } catch (FileNotFoundException e) { e.printStackTrace();
    } catch (IOException e) {e.printStackTrace();
    }

    }

    // For filter and revise turn information and time from raw data to second count start from
    application recording
    private StringBuffer filterTurnInformation(String input){
        StringBuffer result = new StringBuffer();
        Date startTime = new Date();
        String []previous = {"", ""};
        StringTokenizer st = new StringTokenizer(input, "|");
        while (st.hasMoreTokens()){
            String []line = st.nextToken().split(",");
            int h = Integer.parseInt(line[1].substring(9,11));
            int m = Integer.parseInt(line[1].substring(11,13));
            int s = Integer.parseInt(line[1].substring(13,15));

```



```

GregorianCalendar g = new GregorianCalendar(1,1,1,h,m,s);

if (line[0].compareToIgnoreCase("=")==0){
    System.out.println(h+": "+m+": "+s);
    startTime = new Date(g.getTimeInMillis());

    result.append("0,0");
    result.append(System.lineSeparator());
}else{
    Date compare = new Date(g.getTimeInMillis());
    int diff = (int)(compare.getTime() - startTime.getTime()) / 1000;

    if (!line[0].equalsIgnoreCase(previous[0])||
        !line[1].equalsIgnoreCase(previous[1])){

        result.append(line[0]);
        result.append(",");
        result.append(diff);
        result.append(System.lineSeparator());

        previous[0] = line[0];
        previous[1] = line[1];
    }
}

return result;
}

// For filter and revise step information and time from raw data to second count start from
application recording
private StringBuffer filterStepInformation(String input){
    StringBuffer result = new StringBuffer();

    Date startTime = new Date();
    String []previous = {"", "", ""};
    StringTokenizer st = new StringTokenizer(input, "|");
    while (st.hasMoreTokens()){
        String []line = st.nextToken().split(",");

```

```

int h = Integer.parseInt(line[2].substring(9,11));
int m = Integer.parseInt(line[2].substring(11,13));
int s = Integer.parseInt(line[2].substring(13,15));
GregorianCalendar g = new GregorianCalendar(1,1,1,h,m,s);

if (line[0].compareToIgnoreCase("=")==0){
    System.out.println(h+":"+m+":"+s);
    startTime = new Date(g.getTimeInMillis());

    result.append("0,0,0");
    result.append(System.lineSeparator());
}
else{
    Date compare = new Date(g.getTimeInMillis());
    int diff = (int)(compare.getTime() - startTime.getTime()) / 1000;

    if (!line[0].equalsIgnoreCase(previous[0])
        || !line[1].equalsIgnoreCase(previous[1])
        || !line[2].equalsIgnoreCase(previous[2])){
        result.append(line[0]);
        result.append(",");
        result.append(line[1]);
        result.append(",");
        result.append(diff);
        result.append(System.lineSeparator());

        previous[0] = line[0];
        previous[1] = line[1];
        previous[2] = line[2];
    }
}
return result;
}

```

```

/*****
 * Camera Functions
 *****/

public static Camera getCameraInstance(){
    Camera c = null;
    try {
        c = Camera.open(); // attempt to get a Camera instance
        Camera.Parameters params = c.getParameters();
        params.setRotation(90);
        c.setParameters(params);
    }
    catch (Exception e){
        // Camera is not available (in use or does not exist)
    }
    return c; // returns null if camera is unavailable
}

private boolean prepareVideoRecorder(){
    mMediaRecorder = new MediaRecorder();
    System.out.println("Step 1: Unlock and set camera to MediaRecorder");
    mCamera.unlock();
    mMediaRecorder.setCamera(mCamera);

    System.out.println("Step 2: Set sources");
    mMediaRecorder.setAudioSource(MediaRecorder.AudioSource.CAMCORDER);
    mMediaRecorder.setVideoSource(MediaRecorder.VideoSource.CAMERA);

    System.out.println("Step 3: Set a CamcorderProfile (requires API Level 8 or higher)");
    mMediaRecorder.setProfile(CamcorderProfile.get(CamcorderProfile.QUALITY_480P));

    // Step 4: Set output file
    mVideo = getOutputMediaFile(MEDIA_TYPE_VIDEO);
    mMediaRecorder.setOutputFile(mVideo.toString());
}

```

```

// Step 5: Set the preview output
mMediaRecorder.setPreviewDisplay(mPreview.getHolder().getSurface());
System.out.println("All camera setting before prepare");
// Step 6: Prepare configured MediaRecorder
try {
    mMediaRecorder.prepare();
} catch (IllegalStateException e) {
    e.printStackTrace();
    releaseMediaRecorder();
    return false;
} catch (IOException e) {
    e.printStackTrace();
    releaseMediaRecorder();
    return false;
}
return true;
}

private void stopVideoRecord(){
    if (isRecording){
        // stop recording and release camera
        mMediaRecorder.stop(); // stop the recording
        releaseMediaRecorder(); // release the MediaRecorder object
        mCamera.lock();        // take camera access back from MediaRecorder
        refreshMTPFile(mVideo);
        // inform the user that recording has stopped
        isRecording = false;
    }
}

private void startVideoRecord(){
    // initialize video camera
    if (!isRecording && prepareVideoRecorder()) {
        resetValue();
        System.out.println("step:Prepare camera");
        // Camera is available and unlocked, MediaRecorder is prepared,
        // now you can start recording
    }
}

```

```

        mMediaRecorder.start();
        System.out.println("step:Start record");
        isRecording = true;
    } else {
        // prepare didn't work, release the camera
        releaseMediaRecorder();
    }
}

@Override
protected void onPause() {
    super.onPause();
    releaseMediaRecorder();    // if you are using MediaRecorder, release it first
    releaseCamera();           // release the camera immediately on pause event
}

private void releaseMediaRecorder(){
    if (mMediaRecorder != null) {
        mMediaRecorder.reset(); // clear recorder configuration
        mMediaRecorder.release(); // release the recorder object
        mMediaRecorder = null;
        mCamera.lock();           // lock camera for later use
    }
}

private void releaseCamera(){
    if (mCamera != null){
        mCamera.release();    // release the camera for other applications
        mCamera = null;
    }
}
}

```


2. CameraPreview.java ซอร์สโค้ดที่ทำงานร่วมกับ MyCameraSurfaceView.java เพื่อ
ทำการแสดงผลการถ่ายภาพของกล้องบนสมาร์ตโฟนบนหน้าจอของโปรแกรม

```
package com.starboardland.pedometer;
import java.io.IOException;
import android.content.Context;
import android.hardware.Camera;
import android.util.Log;
import android.view.SurfaceHolder;
import android.view.SurfaceView;
public class CameraPreview extends SurfaceView implements SurfaceHolder.Callback {
    private SurfaceHolder mHolder;
    private Camera mCamera;

    public CameraPreview(Context context, Camera camera) {
        super(context);
        mCamera = camera;

        // Install a SurfaceHolder.Callback so we get notified when the underlying surface is created
        and destroyed.
        mHolder = getHolder();
        mHolder.addCallback(this);
        // deprecated setting, but required on Android versions prior to 3.0
        mHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
    }

    public void surfaceCreated(SurfaceHolder holder) {
        // The Surface has been created, now tell the camera where to draw the preview.
        try {
            mCamera.setPreviewDisplay(holder);
            mCamera.startPreview();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```
public void surfaceDestroyed(SurfaceHolder holder) { }

public void surfaceChanged(SurfaceHolder holder, int format, int w, int h) {
    // If your preview can change or rotate, take care of those events here.
    // Make sure to stop the preview before resizing or reformatting it.

    if (mHolder.getSurface() == null){
        // preview surface does not exist
        return;
    }

    // stop preview before making changes
    try {
        mCamera.stopPreview();
    } catch (Exception e){ }

    // set preview size and make any resize, rotate or reformatting changes here start preview
    with new settings
    try {
        mCamera.setPreviewDisplay(mHolder);
        mCamera.startPreview();
    } catch (Exception e){
    }
}
```

3. MyCameraSurfaceView.java ซอร์สโค้ดที่ทำงานร่วมกับ CameraPreview.java เพื่อทำการแสดงผลการถ่ายภาพของกล้องบนสมาร์ตโฟนบนหน้าจอของโปรแกรม

```
package com.starboardland.pedometer;
import java.io.IOException;
import android.content.Context;
import android.hardware.Camera;
import android.view.SurfaceHolder;
import android.view.SurfaceView;

public class MyCameraSurfaceView extends SurfaceView implements
    SurfaceHolder.Callback {
    private SurfaceHolder mHolder;
    private Camera mCamera;
    public MyCameraSurfaceView(Context context, Camera camera) {
        super(context);
        mCamera = camera;

        // Install a SurfaceHolder.Callback so we get notified when the
        // underlying surface is created and destroyed.
        mHolder = getHolder();
        mHolder.addCallback(this);
    }

    @Override
    public void surfaceChanged(SurfaceHolder holder, int format, int weight,int height) {
        // If your preview can change or rotate, take care of those events here.
        // Make sure to stop the preview before resizing or reformatting it.

        if (mHolder.getSurface() == null) {
            // preview surface does not exist
            return;
        }
    }
}
```

```

// stop preview before making changes
try {
    mCamera.stopPreview();
} catch (Exception e) {
    // ignore: tried to stop a non-existent preview
    e.printStackTrace();
}

// make any resize, rotate or reformatting changes here
// start preview with new settings
try {
    mCamera.setPreviewDisplay(mHolder);
    mCamera.startPreview();
} catch (Exception e) {
    e.printStackTrace();
}
}

@Override
public void surfaceCreated(SurfaceHolder holder) {
    // The Surface has been created, now tell the camera where to draw the
    preview.
    try {
        mCamera.setPreviewDisplay(holder);
        mCamera.startPreview();
        mCamera.setDisplayOrientation(90);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

@Override
public void surfaceDestroyed(SurfaceHolder holder) {}
}

```

ภาคผนวก ข.

ซอร์สโค้ดโปรแกรมในเครื่องมือ MATLAB

TNI

ซอร์สโค้ดโปรแกรมในเครื่องมือ MATLAB

1. Main.m เป็นส่วนโค้ดหลักในการประมวลผลเพื่อสร้างแผนผังการเดิน และทำแยกฟังก์ชันบางส่วนเพื่อให้โค้ดเป็นระเบียบและง่ายต่อการทำความเข้าใจมากขึ้น

```
%-----
% Get information from mobile for
% [] Video that will get images to use in floor plan's generate process
% [] Steps count with time for reference in process (from accelerometer)
% [] Turn detection information for reference in process (from gyroscope)
%-----
inputPath = strcat('C:\Users\vameso\Desktop\Present\Data\Complex\20150124-152022');
outputVideolImagePath = strcat(inputPath,'\out');

if exist(outputVideolImagePath,'dir') == 0
    mkdir(outputVideolImagePath);
end

% Get video and convert into images and put into "InputImages" folder
video2pic(strcat(inputPath,'\VID.mp4'),outputVideolImagePath,'jpg');
% After capture image from video. function will create 'vid' folder and put
% all answer in it. So need to add 'vid' after path name
outputVideolImagePath = strcat(outputVideolImagePath,'vid');

%-----
% Start find floor boundaries and then find features and patterns of each
% picture
%-----
filelist = dir(fullfile(outputVideolImagePath,'*.jpg'));
duration = length(filelist);
floorFeature = zeros(duration,1);
temp = zeros(duration,1);

for i=1:duration
    floorFeature(i) = FindFeatures(outputVideolImagePath,strcat('\',filelist(i).name),inputPath);
```



```

    temp(i) = floorFeature(i);
end
%-----
% Start floor plan generation process
% [] Get gyroscope data
% [] Get walking data
% [] Create floor plan from above information
%-----
% Get Accelerometer's activities while walking
stepRawData = dlmread(strcat(inputPath, 'step.txt'), ',');
% Get gyroscope's activities while walking
turnRawData = dlmread(strcat(inputPath, 'turn.txt'), ',');
% Declare array for step and turn data with the same size of array with
% video duration. These two will use with floorFeature
stepCount = zeros(duration,1);
stepDistance = zeros(duration,1);
turnData = zeros(duration,1);
%get second of step into floorInputArray. Not all floorInputArray will have
%step info
size = length(stepRawData);
for i=1:size(1)
    %get second
    second = stepRawData(i,3);
    %put step info into the same row in second that get from step
    if second > 0
        if stepRawData(i,1) == 0
            stepCount(second) = stepRawData(i-1,1);
            stepDistance(second) = stepRawData(i-1,2);
        else
            stepCount(second) = stepRawData(i,1);
            stepDistance(second) = stepRawData(i,2);
        end
    end
end
end
%fill in zero step in some second

```

```

for i=length(stepCount):-1:2
    if stepCount(i) == 0
        if stepCount(i-1) == 0 && i-2 > 0
            stepCount(i) = stepCount(i-2);
        else
            stepCount(i) = stepCount(i-1);
        end
    end
end

%calculate number of steps for each second because some second will go more
%than 1 step
for i=length(stepCount):-1:2
    if stepCount(i) ~= 0 && stepCount(i-1) ~= 0
        stepCount(i) = stepCount(i) - stepCount(i-1);
    end
end

%get second of turn info into floorInputArray. Not all floorInputArray will
%have turn info
size = length(turnRawData);
if size > 1 && size ~= 2
    for i=1:size(1)
        %get second
        second = turnRawData(i,2);

        %put turn info into the same row in second that get
        %from turn
        if second > 0
            turnData(second) = turnRawData(i,1);
        end
    end
end

% merge turn into 1 point (turning will be 2-3 seconds)
for i=1:duration
    if (turnData(i) == 1 || turnData(i) == -1)
        if (turnData(i+1) == 1 || turnData(i+1) == -1)

```

```

        if floorFeature(i+1) ~= FeatureType.path_skip
            floorFeature(i) = bitor(floorFeature(i),floorFeature(i+1));
            floorFeature(i+1) = FeatureType.path_skip;
        end
    end

    if (turnData(i+2) == 1 || turnData(i+2) == -1)
        if floorFeature(i+2) ~= FeatureType.path_skip
            floorFeature(i) = bitor(floorFeature(i),floorFeature(i+2));
            floorFeature(i+2) = FeatureType.path_skip;
        end
    end
end
end

% first loop for revise map when detect any turn then 2 previous picture
% will turn side on map
for i=1:duration
    % check left turn, if turn then 3 previous pictures will remove feature
    % for their left ( not adjuscent picture with turn )
    if ( turnData(i) == 1 )
        if bitand(floorFeature(i-2),FeatureType.vertical_on_left,'int16') ==
FeatureType.vertical_on_left
            floorFeature(i-2) = floorFeature(i-2) - FeatureType.vertical_on_left;
        end

        if bitand(floorFeature(i-3),FeatureType.vertical_on_left,'int16') ==
FeatureType.vertical_on_left
            floorFeature(i-3) = floorFeature(i-3) - FeatureType.vertical_on_left;
        end
    end

    % check right turn, if turn then 3 previous pictures will remove feature
    % on their right ( not adjuscent picture with turn )
    if ( turnData(i) == -1)

```

```

        if bitand(floorFeature(i-2),FeatureType.vertical_on_right,'int16') ==
FeatureType.vertical_on_right
            floorFeature(i-2) = floorFeature(i-2) - FeatureType.vertical_on_right;
        end

        if bitand(floorFeature(i-3),FeatureType.vertical_on_right,'int16') ==
FeatureType.vertical_on_right
            floorFeature(i-3) = floorFeature(i-3) - FeatureType.vertical_on_right;
        end
    end
end

% second loop for check pattern within 3 picture that have vertical line in
% the same side, first picture will be remove this feature
for i=duration:-1:3
    %check continuous right vertical line
    if (bitand(floorFeature(i),FeatureType.vertical_on_right,'int16') == FeatureType.vertical_on_right)
        right_score = 0;
        if (i+1<=duration && bitand(floorFeature(i-1),FeatureType.vertical_on_right,'int16') ==
FeatureType.vertical_on_right)
            right_score = right_score+1;
        end
        if (i+2<=duration && bitand(floorFeature(i-2),FeatureType.vertical_on_right,'int16') ==
FeatureType.vertical_on_right)
            right_score = right_score+1;
        end

        if right_score > 1 && floorFeature(i) ~= FeatureType.path_skip
            floorFeature(i-1) = bitand(floorFeature(i-
1),bitcmp(FeatureType.vertical_on_right,'int16'),'int16');
            floorFeature(i-2) = bitand(floorFeature(i-
2),bitcmp(FeatureType.vertical_on_right,'int16'),'int16');
        else
            floorFeature(i) = floorFeature(i) - FeatureType.vertical_on_right;
        end
    end
end

```

```

% check continuous left vertical line
if (bitand(floorFeature(i),FeatureType.vertical_on_left,'int16') == FeatureType.vertical_on_left)
    left_score = 0;
    if (i+1<=duration && bitand(floorFeature(i-1),FeatureType.vertical_on_left,'int16') ==
FeatureType.vertical_on_left)
        left_score = left_score+1;
    end
    if (i+2<=duration && bitand(floorFeature(i-2),FeatureType.vertical_on_left,'int16') ==
FeatureType.vertical_on_left)
        left_score = left_score+1;
    end

    if left_score > 1 && floorFeature(i) ~= FeatureType.path_skip
        floorFeature(i-1) = bitand(floorFeature(i-1),bitcmp(FeatureType.vertical_on_
left,'int16'),'int16');
        floorFeature(i-2) = bitand(floorFeature(i-
2),bitcmp(FeatureType.vertical_on_left,'int16'),'int16');
        i = i+2;
    else
        floorFeature(i) = floorFeature(i) - FeatureType.vertical_on_left;
    end
end
end

%for update first 2 features to 1
floorFeature(1) = 1;
floorFeature(2) = 1;
%forth loop for increase step when found any turn
for i=1:duration
    if (turnData(i) == 1 || turnData(i) == -1 && floorFeature(i) ~= FeatureType.path_skip)
        stepCount(i-1) = stepCount(i-1) + 1;
    end
end
end

```

```

% fifth loop for decrease step from last non-zero step
for j=1:2
    for i=duration:-1:1
        if stepCount(i) ~= 0
            stepCount(i) = stepCount(i)-1;
            break;
        end
    end
end
% loop for get turn left or turn right to next 2 second
for i=duration:-1:3
    if ( (bitand(floorFeature(i-2),FeatureType.vertical_on_right,'int16') ==
        FeatureType.vertical_on_right || bitand(floorFeature(i-2),FeatureType.vertical_on_right,'int16') ==
        FeatureType.vertical_on_right ) && turnData(i-2) ~= 1 && turnData(i-2) ~= -1 )
        tmp = floorFeature(i);
        floorFeature(i) = floorFeature(i-2);
        floorFeature(i-2) = tmp;
    end
end
% start point
a_px = 250;
a_py = 400;
% background size is 1024 x 1024
imshow('D:\Dropbox\TNI\Thesis\IMATLAB Codes\My Code\FloorPiece\background.jpg');
% %load floor pieces
% feature:
p1l = imread('D:\Dropbox\TNI\Thesis\IMATLAB Codes\My Code\FloorPiece\1l.jpg');
% feature:1,3
p1s = imread('D:\Dropbox\TNI\Thesis\IMATLAB Codes\My Code\FloorPiece\1s.jpg');
% feature:
p1r = imread('D:\Dropbox\TNI\Thesis\IMATLAB Codes\My Code\FloorPiece\1r.jpg');
% feature:
p2lr = imread('D:\Dropbox\TNI\Thesis\IMATLAB Codes\My Code\FloorPiece\2lr.jpg');
% feature:

```



```

p2ls = imread('D:\Dropbox\TNI\Thesis\MATLAB Codes\My Code\FloorPiece\2ls.jpg');
% feature:9,11
p2sr = imread('D:\Dropbox\TNI\Thesis\MATLAB Codes\My Code\FloorPiece\2sr.jpg');
% feature:
p3lsr = imread('D:\Dropbox\TNI\Thesis\MATLAB Codes\My Code\FloorPiece\3.jpg');
% current direction that will goto
% 0=north, 1=east, 2=south, 3=west
a_direction = 0;

for i=1:duration
    a_feature = floorFeature(i);
    %for debug only, can comment when no need them
    a_turnData = turnData(i);
    a_stepCount = stepCount(i);
    %check for main walking path
    if (turnData(i) == 0)
        a_feature = bitor(a_feature,FeatureType.main_path);
    % turn left
    elseif (turnData(i) == 1 && a_feature ~= FeatureType.path_skip)
        a_feature = bitor(a_feature,FeatureType.vertical_on_left);
        if FeatureType.vertical_on_right == bitand(floorFeature(i-1),FeatureType.vertical_on_right)
            a_feature = bitor(a_feature,FeatureType.vertical_on_right);
        end
    % turn right
    elseif (turnData(i) == -1 && a_feature ~= FeatureType.path_skip)
        a_feature = bitor(a_feature,FeatureType.vertical_on_right);
        if FeatureType.vertical_on_left == bitand(floorFeature(i-
1),FeatureType.vertical_on_left)
            a_feature = bitor(a_feature,FeatureType.vertical_on_left);
        end
    end

    if a_feature == 1    piece = p1s;
    elseif a_feature == 2    piece = p1s;
    elseif a_feature == 3    piece = p1s;

```

```

elseif a_feature == 4    piece = p1l;
elseif a_feature == 5    piece = p2ls;
elseif a_feature == 6    piece = p1l;
elseif a_feature == 7    piece = p2ls;
elseif a_feature == 8    piece = p1r;
elseif a_feature == 9    piece = p2sr;
elseif a_feature == 10   piece = p1r;
elseif a_feature == 11   piece = p2sr;
elseif a_feature == 12   piece = p2lr;
elseif a_feature == 13   piece = p3lsr;
elseif a_feature == 14   piece = p2lr;
elseif a_feature == 15   piece = p3lsr;
end

if (turnData(i) == 1 || turnData(i) == -1)
    a_step = 1;
else
    a_step = stepCount(i);
end

for g=1:a_step
    %plot piece of floor and reference red point
    if a_feature ~= FeatureType.path_skip
        hold on;
        %if piece is not 1s then move 1 more step before paint
        if ( a_feature ~= 1 && a_feature ~= 2 && a_feature ~= 3 && g == a_step)
            if a_direction == 0
                a_py = a_py-10;
            elseif a_direction == 1
                %a_px = a_px+10;
            elseif a_direction == 2
                a_py = a_py+10;
            elseif a_direction == 3
                a_px = a_px-10;
            end
        end
    end
end

```

```

        image(a_px,a_py,piece);
    else
        image(a_px,a_py,p1s);
    end

    plot(a_px,a_py,'x','LineWidth',2,'Color','red');
    hold off;
end
%go straight
if (turnData(i) == 0)
    if (FeatureType.vertical_on_right == bitand(a_feature,FeatureType.vertical_on_right) ...
        || (FeatureType.vertical_on_left == bitand(a_feature,FeatureType.vertical_on_left) ))
        turnMultiply = 1;
    else
        turnMultiply = 1;
    end

    if a_direction == 0
        a_py = a_py-10*(turnMultiply);
    elseif a_direction == 1
        a_px = a_px+10*(turnMultiply);
    elseif a_direction == 2
        a_py = a_py+10*(turnMultiply);
    elseif a_direction == 3
        a_px = a_px-10*(turnMultiply);
    end
end
end

if (turnData(i) == 1 ...
    && turnData(i-1) == 0 ...
    && a_feature ~= FeatureType.path_skip)
    a_direction = mod((a_direction-1),4);
    p1l = imrotate(p1l,90);
    p1s = imrotate(p1s,90);
end

```

```

p1r = imrotate(p1r,90);
p2lr = imrotate(p2lr,90);
p2ls = imrotate(p2ls,90);
p2sr = imrotate(p2sr,90);

if a_direction == 0
    a_py = a_py-10;
elseif a_direction == 1
    a_px = a_px+30;
elseif a_direction == 2
    a_py = a_py+20;
elseif a_direction == 3
    a_px = a_px-10;
end

% turn right
elseif (turnData(i) == -1 ...
&& turnData(i-1) == 0 ...
&& a_feature ~= FeatureType.path_skip)
    a_direction = mod((a_direction+1),4);
    p1l = imrotate(p1l,-90);
    p1s = imrotate(p1s,-90);
    p1r = imrotate(p1r,-90);
    p2lr = imrotate(p2lr,-90);
    p2ls = imrotate(p2ls,-90);
    p2sr = imrotate(p2sr,-90);

    if a_direction == 0
        a_py = a_py-10;
    elseif a_direction == 1
        a_px = a_px+30;
    elseif a_direction == 2
        a_py = a_py+20;

```

```

elseif a_direction == 3
    a_px = a_px-30;
end
end
end
saveas(gcf,strcat(inputPath,'VIDPlan.jpg'));

```

2. FeatureType.m เป็นโค้ดส่วนของการกำหนดค่ารูปแบบของทางเดินในรูปแบบของบิตจุดประสงค์เพื่อเป็นต้นแบบและง่ายต่อการนำไปใช้งาน

```

classdef FeatureType
    properties (Constant)
        main_path      = 1;
        wall_ahead      = 2;
        vertical_on_left = 4;
        vertical_on_right = 8;
        dead_end = 1024;
        path_skip = 2048;
    end
end

```

3. FloorDetection.m เป็นฟังก์ชันโค้ดเพื่อการวิเคราะห์รูปแบบของพื้นทางเดินโดยประยุกต์มาจากโค้ดจาก <http://www.mathworks.com/help/images/ref/houghlines.html> โดยเพิ่มเติมในส่วนของการเตรียมภาพก่อนทำการใช้ฟังก์ชัน Hough Transform เพื่อทำการหาเส้นตรงภายในภาพ

```

function lines = FloorDetection(filePath, imFileName,output_path)
%line minimum length
minLength = 100;
%1.Read an image into the MATLAB? workspace.
im = imread(strcat(filePath,imFileName));
% change to gray scale image
gi = rgb2gray(im);

```

```

imwrite(gi, strcat('C:\Users\vameso\Desktop\Present\Data\Complex\20150124-
152022\',imFileName,'_g.jpg'));

% change to binary image
% bi = adaptivethreshold(gi,50,0.1,0);

%3.Find the edges in the image using the edge function.
BW = edge(gi,'canny');
imwrite(BW, strcat('C:\Users\vameso\Desktop\Present\Data\Complex\20150124-
152022\',imFileName,'_bw.jpg'));

%4.Compute the Hough transform of the image using the hough function.
[H,theta,rho] = hough(BW);

%5.Display the transform using the imshow function.
%figure, imshow(imadjust(mat2gray(H)),[],'XData',theta,'YData',rho,...
% 'InitialMagnification','fit');
%xlabel('\theta (degrees)'), ylabel('\rho');
%axis on, axis normal, hold on;
%colormap(hot)

%6.Find lines in the image using the houghlines function.
P = houghpeaks(H,3);%,'threshold',ceil(0.3*max(H(:)))));

%7.Superimpose a plot on the image of the transform that identifies the peaks.
% x = theta(P(:,2));
% y = rho(P(:,1));
% plot(x,y,'s','color','black');

%8.Find lines in the image using the houghlines function.
lines = houghlines(BW,theta,rho,P,'FillGap',10,'MinLength',minLength);

%9.Create a plot that superimposes the lines on the original image.
figure, imshow(im), hold on
max_len = 0;
for k = 1:length(lines)
    xy = [lines(k).point1; lines(k).point2];
    plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');
    % Plot beginnings and ends of lines
    plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
    plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');
    % Determine the endpoints of the longest line segment
    len = norm(lines(k).point1 - lines(k).point2);

```



```

        if ( len > max_len)
            max_len = len;
            xy_long = xy;
        end
    end
    imFileName = strrep(imFileName,'.jpg','_h.jpeg');
    saveas(gcf,strcat(output_path,'\',imFileName));
    close all;
end

```

4. FindFeature.m เป็นฟังก์ชันโค้ดที่ใช้ในการกำหนดรูปแบบของทางเดินโดยทำการวิเคราะห์จากเส้นตรงภายในภาพ

```

function feature = FindFeatures(file_path, input_image,output_path)
%input_image = 'D:\FPC\sample\videooutput\sino2\VID\VID.mp4_0064.jpg';
lines = FloorDetection(file_path,input_image,output_path);
%degree for identify which type of line
horizonMin = 84;
horizonMax = 90;
verticalMin = 0;
verticalMax = 6;
pathMin = 13;
pathMax = 50;
%position of middle of the picture on X-AXIS for specific line is on
%the left or on the right of the picture. unit is pixel
mid_of_pic = 360;
%group of line types
verticalLines = {};
horizonLines = {};
pathLines = {};
%result for this picture
feature = 0;
for(i=1:length(lines))
    theta = abs(lines(i).theta);

```

```

if (theta >= horizonMin && theta <= horizonMax)
    horizonLines{1,length(horizonLines)+1} = lines(i);
elseif (theta >= pathMin && theta <= pathMax)
    pathLines{1,length(pathLines)+1} = lines(i);
elseif (theta >= verticalMin && theta <= verticalMax)
    verticalLines{1,length(verticalLines)+1} = lines(i);
end
end
% check for there are lines that should be path lines in this picture
% or not. If yes, this picture should has main walk path evenif there
% are only 1 side of line's path
if (~isempty(pathLines))
    feature = bitor(feature,FeatureType.main_path);
end
% check this picture about horizontal lines in this picture to check
% path in front of walker. So if this picture can check horizontal
% lines, then this picture should see wall in front of walker.
if (~isempty(horizonLines))
    feature = bitor(feature,FeatureType.wall_ahead);
end
% check for there are any lines can detect as vertical line as there is
% any type of junction in this picture. If there are vertical lines
% then there are junction or some door that make floor plan know this
% walk path has left or right path user able to walk to.
% About nature of these lines
for i=1:length(verticalLines)
    %point1(2) and point2(2) is x
    if ( verticalLines{i}.point2(1) < mid_of_pic ) && ( verticalLines{i}.point1(1) > mid_of_pic )
        feature = bitor(feature,FeatureType.vertical_on_left);
    elseif ( verticalLines{i}.point2(1) > mid_of_pic ) && ( verticalLines{i}.point1(1) > mid_of_pic )
        feature = bitor(feature,FeatureType.vertical_on_right);
    end
end
end
end

```

5. Video2pic.m เป็นโค้ดที่ดัดแปลงจากโค้ดของ Shiquan Wang จาก url <http://www.mathworks.com/matlabcentral/fileexchange/25607-save-a-video-to-images/content/video2pic.m> เพื่อทำการนำภาพออกมาจากวิดีโอ โดยทำการดัดแปลงโค้ดเพื่อให้ฟังก์ชันทำการดึงภาพทุกวินาทีแทนการดึงภาพทุกเฟรมภายในวิดีโอ โดยใช้ค่าเฟรมเรทของวิดีโอในการคำนวณ

```
function video2pic( videoFile, outputPath, picFormat )
%VIDEO2PIC converts video frames into pictures.
% This function uses mmreader to read videos. Thus video format supported
% by system can be converted to pictures. Output file name follows the
% format: VideoName_FrameNumber.PicSuffix.
%
% VIDEO2PIC input parameters interactively.
% VIDEO2PIC( VIDEOFILE, OUTPUTPATH, PICFORMAT ) runs as a
% function.%
% videoFile: full path of video to convert.
% outputPath: pictures will be saved under outputPath/videoName/.
% picFormat: check 'help imwrite'
%
% Shiquan Wang @ CASIA
% sqwang@nlpr.ia.ac.cn
% http://www.cbsr.ia.ac.cn/users/sqwang/
% revised on 2009/10/19
%% Initialization
if nargin < 1
    videoFile = input('Input full video path:', 's');
    outputPath = input('Input video output path[Default is under the video path]:', 's');
    if isempty(outputPath)
        outputPath = videoFile(1, 1:find(videoFile=='\ ', 1, 'last')-1);
    end
    picFormat = input('Input output pic format[JPG by default]:');
    if isempty(picFormat)
        picFormat = 'JPG';
    end
end
end
```

```

%% Main part
videoObject = VideoReader(videoFile);
numFrames = get(videoObject, 'numberOfFrames')
info = mmfileinfo(videoFile);
duration = info.Duration
frameSkip = ceil(numFrames/duration)
fileName = get(videoObject, 'Name');
% bitsPerPix = get(videoObject, 'BitsPerPixel');
% H = get(videoObject, 'Height');
% W = get(videoObject, 'Width');
% bytesPerFrame = bitsPerPix*H*W/8;
outputPath = fullfile(outputPath, fileName(1, 1:find(fileName=='.', 1, 'last')-1));
numOrder = max(4, size(int2str(numFrames), 2));
mkdir(outputPath);

indexFrame = [1 1];
frameLimit = 1000; % This can be set a larger value with enough memory.
while indexFrame(1, 1) <= numFrames
    % memorynow = memory;
    % memoryMax = memorynow.MaxPossibleArrayBytes;
    % frameLimit = floor(memoryMax/bytesPerFrame*0.8);
    indexFrame(1, 2) = min(numFrames, indexFrame(1, 1) + frameLimit);
    frameAll = read(videoObject, indexFrame);

    for i = indexFrame(1, 1):frameSkip:indexFrame(1, 2) %save frames to pic
        imgFrame = frameAll(:, :, i - indexFrame(1, 1) + 1);
        saveFormat = strcat('%s\\%s_%0', int2str(numOrder), 'd.%s');
        picName = sprintf(saveFormat, outputPath, fileName, i, picFormat);
        imgFrame = imcrop(imgFrame, [0 240 720 240]);
        imwrite(imgFrame, picName);
        disp(picName); % This output can be turned off.
    end
    indexFrame(1, 1) = indexFrame(1, 2) + 1;
end
end

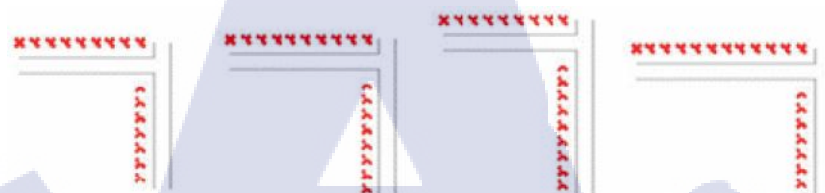
```

ภาคผนวก ค.
ภาพผลลัพธ์แผนผังการเดิน

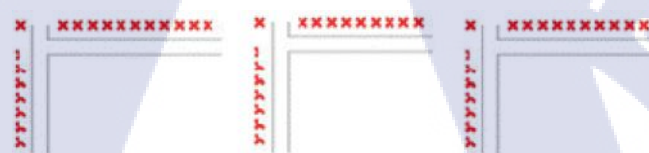
TNI

ภาพผลลัพธ์แผนผังการเดิน

สถานที่ : Sino Thai ชั้น 10



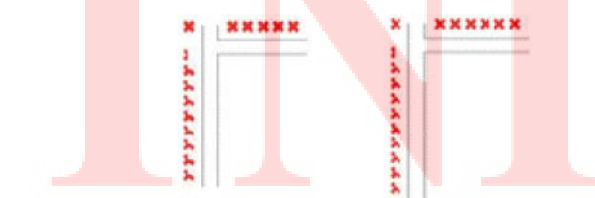
รูปที่ ค.1 แผนผังการเดินเส้นทางที่ 1 เลี้ยวซ้าย



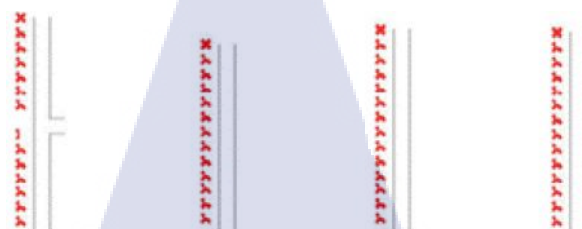
รูปที่ ค.2 แผนผังการเดินเส้นทางที่ 1 เลี้ยวขวา



รูปที่ ค.3 แผนผังการเดินเส้นทางที่ 2 เลี้ยวซ้าย



รูปที่ ค.4 แผนผังการเดินเส้นทางที่ 2 เลี้ยวขวา



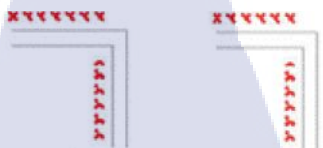
รูปที่ ค.5 แผนผังการเดินทางที่ 3 เติมนตรง

TNI

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

THAI - JAPANESE INSTITUTE OF TECHNOLOGY

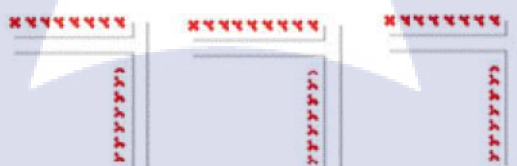
สถานที่ : LPN ท่าพระชั้น 9



รูปที่ ค.6 แผนผังการเดินเส้นทางที่ 1 เลี้ยวซ้าย



รูปที่ ค.7 แผนผังการเดินเส้นทางที่ 1 เลี้ยวขวา



รูปที่ ค.8 แผนผังการเดินเส้นทางที่ 2 เลี้ยวซ้าย

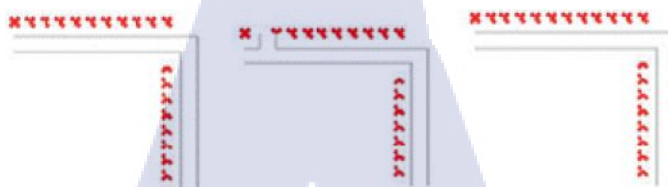


รูปที่ ค.9 แผนผังการเดินเส้นทางที่ 2 เลี้ยวขวา

รูปที่ ค.10 แผนผังการเดินทางที่ 3 เดินตรง

TNI

สถานที่ : TNI อาคาร A ชั้น 6



รูปที่ ค.11 แผนผังการเดินเส้นทางที่ 1 เลี้ยวซ้าย



รูปที่ ค.12 แผนผังการเดินเส้นทางที่ 1 เลี้ยวขวา



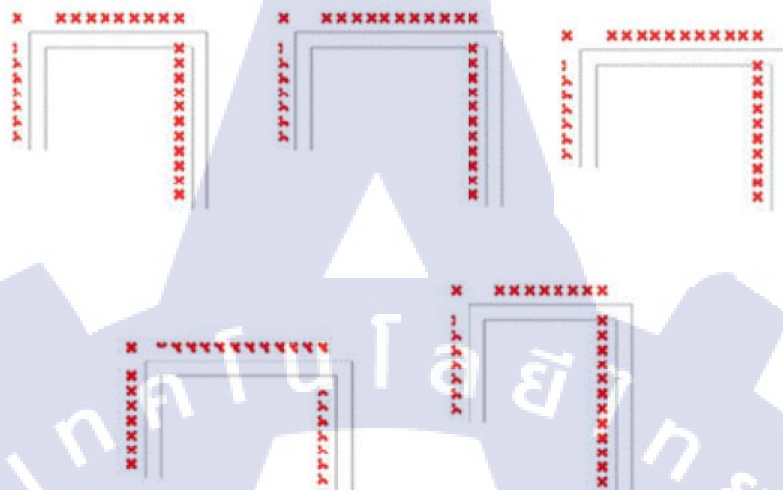
รูปที่ ค.13 แผนผังการเดินเส้นทางที่ 2 เลี้ยวซ้าย



รูปที่ ค.14 แผนผังการเดินเส้นทางที่ 2 เลี้ยวขวา

รูปที่ ค.15 แผนผังการเดินทางที่ 3 เดินตรง

สถานที่ : TNI อาคาร A ชั้น 5



รูปที่ ค.16 แผนผังการเดินทางที่ 4