# ระบบค้นหาเส้นทางและจำลองภาพวิดีโอเส้นทาง

# NAVIGATION SYSTEM AND ROUTE VIDEO SIMULATOR

ธนภูมิ ประกิจวินิจพันธ์
ธนันท์ชัย ธาราศิริวัฒน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

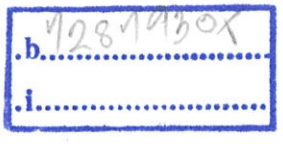
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# ระบบค้นหาเส้นทางและจำลองภาพวิดีโอเส้นทาง

# NAVIGATION SYSTEM AND ROUTE VIDEO SIMULATOR

ธนภูมิ ประกิจวินิจพันธ์

ธนันท์ชัย ธาราศิริวัฒน์




ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

ปริญญานิพนธ์ปีการศึกษา 2558

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบค้นหาเส้นทางและจำลองภาพวิดีโอเส้นทาง

NAVIGATION SYSTEM AND ROUTE SIMULATOR

ผู้จัดทำ

1. นายธนภูมิ ประกิจวินิจพันธ์ รหัสนักศึกษา 55010509
2. นายธนันท์ชัย ธาราศิริวัฒน์ รหัสนักศึกษา 55010528

อรฉัตร จิต

_____ อาจารย์ที่ปรึกษา

(รองศาสตราจารย์ ดร. อรฉัตร จิตต์โสภักตร์)

# ระบบค้นหาเส้นทางและจำลองภาพวิดีโอเส้นทาง

นายธนภูมิ ประกิจวินิจพันธ์ 55010509
นายธนันท์ชัย ธาราศิริวัฒน์ 55010528
รศ.ดร.อรฉัตร จิตต์โสภักตร์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2558

## บทคัดย่อ

เนื่องจากในปัจจุบันการเดินทางเข้ามาภายในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังนั้นมีความสะดวกสบาย และหลากหลายมากกว่าอดีตมาก ถึงแม้ว่าในปัจจุบันจะมีแอพพลิเคชั่นที่ช่วยในการเดินทางให้มาถึงสถาบันมากมาย แต่พบว่ามีการหลงทางบ่อยครั้งซึ่งส่งผลให้เสียเวลาที่มีค่าไปโดยเปล่าประโยชน์ จากปัญหาข้อนี้ทางคณะผู้จัดทำได้เล็งเห็นถึงประโยชน์ในการพัฒนาระบบค้นหาเส้นทาง และจำลองภาพวิดีโอเส้นทางในสถาบันในรูปแบบของเว็บแอพพลิเคชั่นที่เป็นตัวช่วยในการนำทางภายในสถาบัน พร้อมกับแนะนำสถาบันด้วยไปในตัว โดยเว็บแอพพลิเคชั่นนี้สามารถยืนยันตำแหน่งของตนเองด้วยรูปอาคารที่ได้รับจากผู้ใช้งาน โดยเมื่อทางผู้ใช้งานส่งรูปถ่ายอาคารและพิกัดตำแหน่งเข้ามาในระบบแล้ว ทางเว็บแอพพลิเคชั่นจะนำรูปถ่ายมาค้นคืนกับรูปอาคารในบริเวณนั้นในฐานข้อมูล เพื่อทำการยืนยันตำแหน่งปัจจุบันที่ถูกต้องของผู้ใช้ จากนั้นระบบจะให้ผู้ใช้งานเลือกสถานที่ปลายทางต่อไป โดยอาจจะเลือกจากชื่อสถานที่ หน่วยงาน หรือ พิกัดของสถานที่นั้นได้ และยังสามารถเลือกรูปแบบการเดินทางโดยรถยนต์ หรือ การเดิน หลังจากนั้นเว็บแอพพลิเคชั่นจะทำการส่งคำขอไปทำการหาเส้นทางที่ดีที่สุดภายในเซิร์ฟเวอร์ แล้วแสดงผ่านหน้าเว็บแอพพลิเคชั่นให้แก่ผู้ใช้งาน ซึ่งภายในเส้นทางนั้นจะประกอบด้วย เส้นทางจากตำแหน่งปัจจุบัน ไปยัง สถานที่ปลายทาง พร้อมกับวิดีโอแนะนำเส้นทางและรูปภาพพร้อมรายละเอียดของสถานที่ที่เส้นทางนั้น นอกจากจะแนะนำเส้นทางได้แล้วเว็บแอพพลิเคชั่นสามารถเข้ามาดูรายละเอียดของอาคาร หน่วยงาน หรือสถานที่สำคัญ เช่น ธนาคาร ATM ร้านอาหาร ลานจอดรถ และจุดจอดรถสาธารณะ ภายในสถาบันได้ รวมไปถึงการแนะนำการเดินทางด้วยขนส่งสาธารณะจากภายนอกเข้ามาภายในสถาบัน และยังสามารถเรียกดูแผนที่ภาพรวมทั้งหมดภายในสถาบันได้เช่นกัน

# Navigation system and Route video simulator

Mr. Thanapum Prakitwinitphan 55010509

Mr. Thananchai Tharasirivat 55010528

Assoc.Prof.Dr Orachat Chitsobhuk Advisor

Academic Year 2015

## Abstract

Nowadays, there are several convenient ways to come to the King Mongkut's Institute of Technology Ladkrabang (KMITL) compared to the past. Since people may come for the first time or are not familiar with the buildings or places, knowing only the name of building may not be enough. Therefore, web based navigation system called "KMITL NAVi" is developed to navigate and introduce our campus at the same time. The application can resolve the problems by combining navigation system and route video simulator. The main feature is to navigate users to the destination from where they are. They can register their position using building image from their cameras, building name, or GPS co-ordinate, then the application will search through the images in the database around submitted GPS location, identify the correct position, and use it as the origin place. Then, users are asked to choose their destination in the form of building name, division or GPS co-ordinate. Navigation modes available are driving or walking mode. After selecting the navigation mode, the application will present the direction information together with a navigation video. Moreover, our web application introduces the building and division information, the public transportation, and the overview map of our KMITL campus.

# กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้จะไม่สามารถสำเร็จลุล่วงได้สมบูรณ์เลย หากไม่ได้รับความกรุณาเป็นอย่างสูงจากอาจารย์ที่ปรึกษา คือ รศ.ดร.อรฉัตร จิตต์โสภักตร์ ที่ช่วยคอยให้คำปรึกษาช่วยเหลือคอยให้คำชี้แนะตลอดการทำงาน ไม่ว่าจะเป็นเรื่องงานหรือเรื่องส่วนตัวท่านก็คอยช่วยเหลือเราอยู่ตลอด รวมไปถึงอาจารย์อีกหลายๆท่านในภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ได้ให้ความรู้ตลอดการศึกษา และคำแนะนำต่างๆ ทำให้นำความรู้ และคำชี้แนะเหล่านั้น ทำให้ก่อเกิดเป็นปริญญานิพนธ์ฉบับนี้ได้อย่างสมบูรณ์

ขอขอบคุณเพื่อนๆ พี่ๆ และน้องๆ ทุกคนภายในสาขาวิศวกรรมคอมพิวเตอร์ที่ได้คอยให้กำลังใจ ที่ร่วมทุกข์ร่วมสุขกันมาตลอดจวบจนถึงวันนี้ รวมไปถึงคำแนะนำที่ดีที่คอยแนะนำกันตลอดมา

ขอขอบพระคุณ บิดา มารดา และครอบครัว ที่ได้ให้การสนับสนุนในเรื่องต่างๆ คอยเป็นกำลังใจ เป็นที่ปรึกษา เป็นบุคคลที่พร้อมจะอยู่เคียงข้างทุกเมื่อ ถ้าหากขาดพวกท่านไป ปริญญานิพนธ์ฉบับนี้คงไม่สามารถเสร็จสมบูรณ์ได้

สุดท้ายนี้คุณความดีประโยชน์ และคุณค่าที่เกิดจากปริญญานิพนธ์ฉบับนี้ ขอให้ส่งไปถึงบิดา มารดา ครูบาอาจารย์ทุกท่าน ตลอดจนผู้มีพระคุณแก่คณะผู้จัดทำทุกท่าน

ธนภูมิ ประกิจวินิจพันธ์
ธนันท์ชัย ธาราศิริวัฒน์

# สารบัญ

หน้า


บทคัดย่อภาษาไทย ..... I

บทคัดย่อภาษาอังกฤษ ..... II

กิตติกรรมประกาศ ..... III

สารบัญ ..... IV

สารบัญตาราง ..... VI

สารบัญภาพ ..... VII

บทที่ 1 บทนำ ..... 1

1.1 ความสำคัญและที่มาของโครงงาน ..... 1

1.2 วัตถุประสงค์ของโครงงาน ..... 2

1.3 ขอบเขตของโครงงาน ..... 2

1.4 วิธีการดำเนินการ ..... 3

1.5 ประโยชน์ที่คาดว่าจะได้รับ ..... 3

1.6 ส่วนประกอบของปริญญานิพนธ์ ..... 4

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง ..... 5

2.1 ระบบกำหนดตำแหน่งบนโลกหรือจีพีเอส (Global Position System: GPS) ..... 5

2.2 กูเกิลแมพจาวาสคริปต์เอพีไอ (Google maps Javascript API) ..... 6

2.3 กระบวนการประมวลผลภาพ (Image Processing) ..... 12

บทที่ 3 การออกแบบ และพัฒนา ..... 31

3.1 Conceptual Design ..... 33

3.2 ความต้องการของระบบ (System Requirement) ..... 35

3.3 การทำงานในแต่ละฟังก์ชัน ..... 37

3.4 แผนภาพการไหลของข้อมูลของระบบ (Data Flow Diagram: DFD) ..... 41

3.5 การออกแบบส่วนติดต่อผู้ใช้งาน ..... 45

3.6 การออกแบบฐานข้อมูล ..... 54

3.7 การทำงานในส่วนเปรียบเทียบรูปภาพ ..... 57

# สารบัญ (ต่อ)

หน้า

3.8 การทำงานในส่วนของการแนะนำเส้นทาง ..... 59

บทที่ 4 การทดลองและผลการทดลอง ..... 62

4.1 ผลการทดลองของโครงงาน ..... 62

บทที่ 5 บทสรุป ..... 84

5.1 ผลที่ได้รับจากโครงงาน ..... 84

5.2 ปัญหาและอุปสรรค ..... 85

5.3 แนวทางในการแก้ไข ..... 86

เอกสารอ้างอิง ..... 87

# สารบัญตาราง


ตาราง หน้า

2.1 Affine Transformation ..... 18

4.1 ตัวอย่างการเปรียบเทียบระยะทางที่ใช้จากสถานที่ต้นทางไปยังสถานที่ปลายทางโดยรถยนต์ที่ได้จากระบบแนะนำและระยะทางที่กูเกิลแนะนำ ..... 71

4.2 ตัวอย่างการเปรียบเทียบระยะทางที่ใช้จากสถานที่ต้นทางไปยังสถานที่ปลายทางโดยการเดินที่ได้จากระบบแนะนำและระยะทางที่กูเกิลแนะนำ ..... 72

4.3 การเปรียบเทียบเมื่อเปลี่ยนพารามิเตอร์ Tilt ..... 77

4.4 จำนวนคุณลักษณะเด่นที่จับคู่กันในการตั้งค่าพารามิเตอร์ Tilt ..... 78

4.5 จำนวนคุณลักษณะเด่นที่จับคู่กันในการตั้งค่าพารามิเตอร์ Tilt ในจับคู่ภาพมุมมองอื่นๆ ..... 80

4.6 เวลาเฉลี่ยต่อการเปรียบเทียบภาพถ่าย 1 ครั้ง ..... 82

# สารบัญรูป

รูป หน้า

2.1 เทคนิคการระบุตำแหน่ง ..... 5
2.2 แผนผังการทำงานของไดเรกชั่นเซอร์วิส ..... 6
2.3 แผนผังการทำงานของ Distance Matrix Service ..... 10
2.4 ตัวอย่างจาก Google Street View API ..... 12
2.5 จตุภาคในระบบพิกัดฉาก ..... 13
2.6 ระบบพิกัดเชิงขั้ว ..... 14
2.7 การสะท้อนของจุดในระบบพิกัดเชิงขั้ว ..... 14
2.8 การมีได้หลายจุดในพิกัดเชิงขั้ว ..... 15
2.9 ความสัมพันธ์ระหว่างพิกัดฉากกับพิกัดเชิงขั้ว ..... 15
2.10 ระบบพิกัดลอกโพลา ..... 16
2.11 ตัวอย่างที่ได้จากการแปลงตำแหน่งภาพ ..... 17
2.12 การทำ Inverse Mapping ..... 20
2.13 การทำ Inverse Mapping แบบ First-order ..... 20
2.14 การใช้ $\sigma$ ที่แตกต่างกันไป ..... 22
2.15 การหา Difference of Gaussian ในแต่ละขั้นของ Octave ..... 23
2.16 ผลลัพธ์จากการทำ Difference of Gaussian ในแต่ละขั้นของ Octave ..... 24
2.17 จุด X ที่เป็นจุดอ้างอิงเปรียบเทียบกับจุดรอบข้าง 26 จุด ..... 25
2.18 จุดสีแดงที่เป็นจุดสูงสุดหรือต่ำสุดที่เป็น pixel แต่จุดที่สูงสุด จุดสีเขียว ซึ่งเป็น subpixel ..... 25
2.19 ผลลัพธ์จากการทำ Locate maxima/minima กับ 4 DoG images ..... 26
2.20 ผลลัพธ์ที่ได้หลังจากทำการหา keypoint จากการทดสอบ Edge และ Contrast ที่เหมาะสม ..... 27
2.21 การขนาดและทิศทางของ keypoint ..... 28

# สารบัญรูป (ต่อ)

รูป หน้า

2.22 ผลลัพธ์ของการหาตำแหน่งและทิศทางของการวางตัวของกล้องจาก
หนังสือที่ต้องการ(เส้นขอบสีน้ำเงิน) ....30
2.23 ขั้นตอนการหมุนภาพด้วย Affine Camera Model ....32
3.1 ภาพรวมของระบบ ....33
3.2 แผนภาพยูสเคสไดอะแกรม (Use Case Diagram) ....36
3.3 แผนผังการทำงานของฟังก์ชันนำทางภายในสถาบัน – สถาบัน ด้วยรถยนต์ และการเดิน ....37
3.4 แผนผังการทำงานของฟังก์ชันนำทางภายในสถาบัน – สถาบัน ด้วยรถยนต์
และการเดิน โดยยืนยันตำแหน่งพิกัดปัจจุบันด้วยชื่อ ....38
3.5 แผนผังการทำงานของฟังก์ชันนำทางจากภายนอกสถาบันมาสู่ภายใน ....39
3.6 แผนผังการทำงานของฟังก์ชันเรียกดูข้อมูลสถานที่ หรือหน่วยงานภายในสถาบัน ....40
3.7 แผนผังการทำงานฟังก์ชันอื่นๆ ....41
3.8 Context Diagram ....42
3.9 Data Flow Diagram Level 0 ....43
3.10 Data Flow Diagram Level 1 ....44
3.11 หน้าเริ่มต้นการใช้งาน ....45
3.12 การนำทางจาก ภายในสถาบัน-ภายในสถาบัน ด้วยการขับรถ และ
แสดงสถานที่ระหว่างทางที่ผ่าน ....46
3.13 การนำทางจาก ภายในสถาบัน-ภายในสถาบัน ด้วยการเดิน ....47
3.14 การนำทางจาก ภายนอกสถาบัน-ภายในสถาบัน ด้วยการขับรถ ....48
3.15 หน้าประวัติสถาบัน ....49
3.16 หน้าองค์กรและหน่วยงาน ....50
3.17 หน้ารายละเอียดของตึกเมื่อกดดูในส่วนของรายละเอียด หรือ ในเมนูรายละเอียดตึก ....51

# สารบัญรูป (ต่อ)

รูป หน้า

3.18 วิธีการเดินทางมายังสถาบัน ..... 52
3.19 แผนผังทั้งหมดของสถาบัน ..... 53
3.20 ER-Diagram ..... 54
3.21 ตัวอย่างการเก็บข้อมูลใน Videos table ..... 56
3.22 ตัวอย่างวิดีโอ https://www.youtube.com/watch?v=yPfycEY_Qb4 ..... 56
3.23 ภาพรวมการทำงานของการเปรียบเทียบรูปภาพ ..... 58
3.24 ขั้นตอนการทำงานของอัลกอริทึมเอสตาร์รอบที่ 1 ..... 60
3.25 ขั้นตอนการทำงานของอัลกอริทึมเอสตาร์รอบที่ 2 ..... 60
4.1 หน้าหลักของส่วนการค้นหาเส้นทาง ..... 62
4.2 เมื่อผู้ใช้ยืนยันตำแหน่งด้วยรูปภาพถูกต้อง ..... 63
4.3 เมื่อผู้ใช้ยืนยันตำแหน่งด้วยรูปภาพไม่สำเร็จ ..... 64
4.4 แนะนำภาพอาคารใกล้เคียง เมื่อผู้ใช้ยืนยันตำแหน่งไม่สำเร็จ ..... 64
4.5 การเลือกสถานที่ปลายทาง ..... 65
4.6 เลือกประเภทการเดินทางด้วยรถยนต์ ..... 65
4.7 เลือกประเภทการเดินทางด้วยการเดิน ..... 66
4.8 การนำทางจาก ภายในสถาบัน-ภายในสถาบัน ด้วยการขับรถ
หมุดรูปสถานที่แสดงถึงสถานที่สำคัญระหว่างทางที่ผ่าน ..... 66
4.9 การนำทางจาก ภายในสถาบัน-ภายในสถาบัน ด้วยการเดิน
หมุดรูปสถานที่แสดงถึงสถานที่สำคัญระหว่างทางที่ผ่าน ..... 67
4.10 รายละเอียดเมื่อเลือกที่หมุดรูปสถานที่ ..... 67
4.11 วีดีโอนำทางเมื่อเลือกที่หมุดสีเหลือง ..... 68
4.12 การนำทางจาก ภายนอกสถาบัน-ภายในสถาบัน ..... 68
4.13 ตัวอย่างการเก็บข้อมูลสถานที่ต่างๆ ..... 69
4.14 ตัวอย่างข้อมูลในตารางฐานข้อมูล ..... 69

# สารบัญรูป (ต่อ)

รูป หน้า

4.15 ตัวอย่างการเก็บข้อมูลภาพถ่ายอาคารต่างๆ ..... 69
4.16 ตัวอย่างข้อมูลของภาพถ่ายที่เก็บอยู่ในฐานข้อมูล ..... 70
4.17 ตัวอย่างข้อมูลพิกัดที่ใช้แนะนำเส้นทางที่ทำการเก็บข้อมูล ..... 70
4.18 ตัวอย่างข้อมูลพิกัดที่ใช้แนะนำเส้นทางในฐานข้อมูล ..... 70
4.19 เส้นทางจากอาคารปฏิบัติการรวมวิศวกรรมศาสตร์ 2 ไปยัง
ภาควิชาวิศวกรรมวัดคุมโดยรถยนต์พร้อมแสดงระยะทางที่ใช้จากระบบ ..... 71
4.20 เส้นทางจากอาคารปฏิบัติการรวมวิศวกรรมศาสตร์ 2 ไปยัง
ภาควิชาวิศวกรรมวัดคุมโดยรถยนต์พร้อมแสดงระยะทางที่ใช้จากกูเกิล ..... 72
4.21 เส้นทางจากอาคารเรียนรวมสมเด็จพระเทพไปยังตึกวิศวกรรมศาสตร์ (ตึก A) โดยการเดิน ... 73
4.22 วิดีโอตัวอย่างที่ทำการร้องขอมาจาก Google Streetview API ..... 73
4.23 จำลองการค้นหาอาคารใกล้เคียงในระยะ 50 เมตรที่พิกัด 13.7282473,100.7759738 ..... 74
4.24 ชุดภาพในระยะ 50 เมตร จากพิกัด 13.7282473,100.7759738 ..... 75
4.25 ผลลัพธ์ของการเปรียบเทียบโดยใช้เทคนิค [4] ..... 76
4.26 การเปรียบเทียบภาพโดยใช้เทคนิค [4] แล้วค่ามีความผิดพลาด ..... 76
4.27 อินพุตจากผู้ใช้งาน ..... 78
4.28 ตัวอย่างที่ใช้ในการเปรียบเทียบจากฐานข้อมูล ..... 78
4.29 ผลการเปรียบเทียบระหว่างภาพอินพุตกับตัวอย่างภาพจากอาคาร โดย Tilt เท่ากับ 6 ..... 79
4.30 ผลการเปรียบเทียบระหว่างภาพอินพุตกับตัวอย่างภาพจากอาคาร โดย Tilt เท่ากับ 7 ..... 79
4.31 ตัวอย่างจากฐานข้อมูลในมุมอื่นๆ ..... 80
4.32 ผลการเปรียบเทียบระหว่างภาพอินพุตกับตัวอย่างภาพจากอาคารในมุมอื่น
โดย Tilt เท่ากับ 5 ..... 81
4.33 ผลการเปรียบเทียบระหว่างภาพอินพุตกับตัวอย่างภาพจากอาคารในมุมอื่น
โดย Tilt เท่ากับ 7 ..... 81

# สารบัญรูป (ต่อ)

**รูป** **หน้า**

4.34 ผลการเปรียบเทียบในกรณีที่อาคารมีลักษณะใกล้เคียงกัน .....82

4.35 ผลการเปรียบเทียบในกรณีที่อาคารเป็นคนละอาคารกัน .....83

# บทที่ 1
# บทนำ

## 1.1 ความสำคัญและที่มาของโครงงาน

ในปัจจุบันสถานที่ในประเทศไทยสามารถค้นหาเส้นทางได้จากอินเตอร์เน็ตทั่วไป โดยเฉพาะในกูเกิล (Google) ซึ่งสามารถค้นหาเส้นทางในการเดินทางไปได้เกือบทุกสถานที่ทั้งในกรุงเทพมหานครฯ และต่างจังหวัด โดยใช้บริการค้นหาเส้นทางผ่านกูเกิลแมพ (Google Map) ระบุสถานที่ต้นทาง และปลายทางที่จะต้องการค้นหาเส้นทางจะได้ผลลัพธ์ที่แสดงออกมาคือเส้นทางที่ต้องการ

ผู้วิจัยเล็งเห็นประโยชน์ของกูเกิลแมพบนแอพพลิเคชันจึงนำมาพัฒนาต่อเพื่ออำนวยความสะดวกแก่ผู้ใช้งานในการค้นหาเส้นทางภายในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โดยการค้นหาจากตำแหน่งปัจจุบันของผู้ใช้งานไปยังปลายทางที่ต้องการ และหากต้องการทราบข้อมูล ภายในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทั้งข้อมูลสถานที่ และข้อมูลที่เกี่ยวข้องสามารถใช้งานระบบนี้ได้เช่นกัน ระบบสามารถแสดงข้อมูลประกอบเส้นทางที่ค้นหาในรูปแบบของภาพหรือวีดีโอ โดยที่ผู้ใช้สามารถดูภาพหรือวิดีโอได้ตลอดเส้นทางที่ค้นหา เพื่อให้ผู้ใช้งานเกิดความคุ้นเคยกับสถานที่ได้อย่างชัดเจนและไปถึงจุดหมายได้อย่างถูกต้อง อย่างไรก็ดี เนื่องจากความคลาดเคลื่อนของการระบุตำแหน่งพิกัดของจีพีเอส (GPS) มีความคลาดเคลื่อนจากความเป็นจริง ดังนั้นเพื่อให้ระบบสามารถระบุตำแหน่งผู้ใช้ที่ถูกต้องได้อย่างมีประสิทธิภาพ ระบบจึงเปิดให้ผู้ใช้สามารถส่งภาพสถานที่ที่อยู่ปัจจุบันของผู้ใช้งานมาเทียบกับภาพในฐานข้อมูล เพื่อให้ทราบตำแหน่งปัจจุบันที่แท้จริงของผู้ใช้งาน

ระบบค้นหาเส้นทางและจำลองภาพวีดีโอเส้นทาง เป็นระบบที่อำนวยความสะดวกภายในสถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง ซึ่งให้บริการแนะนำข้อมูลที่เกี่ยวข้องกับสถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง และสามารถค้นหาเส้นทางภายในของสถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบังจากตำแหน่งที่อยู่ปัจจุบันของผู้ใช้งานไปยังสถานที่ปลายทางที่ต้องการ โดยมีการแสดงผลการค้นหาเป็นภาพวีดีโอจำลองเส้นทาง ผ่านเว็บแอพลิเคชันทั้งบนเครื่องพีซี และบนระบบปฏิบัติการบนมือถือ ซึ่งผู้ใช้งานสามารถเข้าถึงได้ทุกที่ทุกเวลาเมื่อเชื่อมต่อกับระบบอินเทอร์เน็ต

ฟีเจอร์สำคัญที่จะเป็นประโยชน์แก่ผู้ใช้งาน ได้แก่ การค้นหาตำแหน่ง การขอเส้นทาง นอกจากนี้ยังมีแมพเอพีไอ (Map API) เป็นชุดเครื่องมือฟรีที่ใช้จาวาสคริป (Java Script) ช่วยให้สามารถนำกูเกิลแมพไว้บนเว็บไซต์อื่น แผนที่ตัวนี้จะสามารถทำงานแบบอินเทอแอคทีฟ (Interactive) ได้อย่างสมบูรณ์ ทั้งยังรองรับการปรับปรุงตามต้องการ ทำให้โปรแกรมคอมพิวเตอร์

แบบกำหนดเองสามารถโต้ตอบกับระบบกูเกิลแมพซึ่งเป็นการส่งเสริมนวัตกรรมและการให้ความช่วยเหลือแก่นักพัฒนาโมบายแอพพลิเคชั่น (Mobile Application) ซึ่งกำลังเป็นที่นิยมในปัจจุบัน เนื่องจากทำให้ผู้ใช้งานเข้าถึงการให้บริการได้ง่ายขึ้น

ระบบนี้จึงเป็นทางเลือกหนึ่งให้กับผู้ใช้งานที่ต้องการเข้าถึงสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังผ่านทางแอพพลิเคชันซึ่งง่ายต่อการเข้าถึง นอกจากนี้เป็นการนำเสนอสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังแก่สังคมเพื่อเพิ่มโอกาสให้บุคคลภายนอกได้รู้จักสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังมากขึ้น และยังถือเป็นการพัฒนาด้านการศึกษาให้ก้าวทันเทคโนโลยีในระดับนานาชาติ

## 1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อพัฒนาระบบนำทางพร้อมวิดีโอแนะนำเส้นทาง เพื่อช่วยนำทางภายใน สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
2) เพื่อค้นหาตำแหน่งปัจจุบันที่ถูกต้องของตนเองภายในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังด้วยภาพถ่าย และพิกัดจีพีเอสได้
3) แนะนำสถานที่ภายในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
4) เพื่อให้บุคคลภายนอกเข้าถึงสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังได้ง่ายขึ้น

## 1.3 ขอบเขตของโครงงาน

1) เว็บแอพพลิเคชันสามารถดูแผนที่ภายในสถาบันได้
2) เว็บแอพพลิเคชันสามารถดูรายละเอียดหน่วยงานของตึกภายในสถาบัน
3) เว็บแอพพลิเคชันสามารถดูสถานที่และสิ่งอำนวยความสะดวกเช่น โรงอาหาร ร้านกาแฟ สนามกีฬา จุดให้บริการรถสาธารณะ ATM ที่จอดรถ
4) เว็บแอพพลิเคชันสามารถถ่ายภาพสถานที่ ในสถาบันเพื่อระบุตำแหน่งปัจจุบันของผู้ใช้งาน
5) เว็บแอพพลิเคชันสามารถดูรายละเอียดภายในหน่วยงานได้
6) เว็บแอพพลิเคชันสามารถดูตารางการเดินรถสาธารณะที่ผ่านสถาบัน เช่น ตารางรถไฟ ตารางแอร์พอตลิงค์ และรถสายสาธารณะ
7) เว็บแอพพลิเคชันสามารถค้นหาเส้นทางภายในสถาบัน และแสดงเส้นทางเป็นวิดีโอเส้นทางได้
8) การถ่ายภาพเพื่อระบุตำแหน่งปัจจุบัน สามารถถ่ายได้เฉพาะตอนกลางวันเท่านั้น

9) การส่งภาพถ่ายของผู้ใช้งานเข้ามาเปรียบเทียบต้องส่งมาพร้อมกับพิกัดเสมอ
10) ไม่สามารถเปรียบเทียบภาพถ่ายที่มีการย้อนแสงได้
11) รองรับ เว็บเบราเซอร์เฉพาะ Google chrome, Safari และเบราเซอร์ของ Android
12) สามารถร้องขอเส้นทางได้เพียง 25,000 เส้นทางต่อวันเท่านั้น อ้างอิงจาก Standard API ของ Google

## 1.4 วิธีการดำเนินการ

1) ศึกษาปัญหาที่ควรพัฒนา
2) ระบุความต้องการและปัญหาโครงงาน
3) ศึกษาข้อมูลเกี่ยวกับทฤษฎีที่จำเป็นในการทำโครงงาน
   - เรื่องการสร้างเว็บ (Web)
   - เรื่องจีพีเอส (Global Positioning System: GPS)
   - เรื่องกระบวนการประมวลผลภาพ (Image Processing)
   - เรื่องกูเกิลแมพเอพีไอ (Google Map Javacript API)
4) เขียนผังการทำงานของระบบโดยรวม (Conceptual Design)
5) รวบรวมข้อมูลภายในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
   - รวบรวมข้อมูลภาพถ่ายของสถานที่ โดยการดึงภาพถ่ายจากกูเกิลเอพีไอ
   - รวบรวมตำแหน่งและพิกัดของสถานที่
   - รวบรวมรายละเอียดของสถานที่
   - รวบรวมรายละเอียดของหน่วยงานในสถาบัน
6) ออกแบบรายละเอียดของระบบ
7) ศึกษาวิธีการแก้ปัญหา และความเป็นไปได้ทั้งหมดของโครงงาน
8) เขียนตารางการทำงานทั้งหมด และแบ่งหน้าที่การทางาน
9) พัฒนาระบบ
10) จัดทำเอกสาร

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ช่วยค้นหาเส้นทางให้กับผู้ที่ใช้งานแอพพลิเคชันโดยมีวิดีโอแนะนำเส้นทางการเดินประกอบด้วย เพื่อให้ผู้ใช้งานเกิดความคุ้นเคยกับเส้นทางที่ต้องการเดินทาง
2) เป็นการแนะนำสถานที่ ภายในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

3) เส้นทางที่ได้จากการใช้งานแอพพลิเคชันจะมีบางเส้นทางที่สามารถเดินทางได้สั้นกว่าเมื่อเทียบกับกูเกิลเอพีไอ รวมไปถึงสามารถดูวิดีโอแนะนำเส้นทางได้
4) ในการระบุตำแหน่งพิกัดปัจจุบันตัวกูเกิลเอพีไอมีความคลาดเคลื่อนอยู่ในบางจุดซึ่งเว็บแอพพลิเคชันนี้สามารถยืนยันตำแหน่งพิกัดปัจจุบันที่ถูกต้อง ให้กับผู้ใช้งานได้

## 1.6 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 บทนำ กล่าวถึงความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง จะกล่าวถึงทฤษฎีเกี่ยวกับ จีพีเอส(Global Positioning System: GPS) กระบวนการประมวลผลภาพ (Image Processing) กูเกิลแมพจาวาสคริปต์เอพีไอ (Google Map Javascript API) ASIFT Algorithm และ SIFT Algorithm

บทที่ 3 การออกแบบและการพัฒนาระบบ จะประกอบด้วย 5 ส่วน คือ ภาพรวมของระบบ (Conceptual Design) ความต้องการของระบบ (System Requirement) ยูสเคสไดอะแกรม (Use Case Diagram) การทำงานในแต่ละฟังก์ชันของเว็บแอพพลิเคชัน แผนภาพการไหลของข้อมูลของระบบ (Data Flow Diagram: DFD) และการออกแบบส่วนติดต่อผู้ใช้งาน (User Interface) รวมไปถึงตัวอัลกอริทึม ASIFT [3] ที่ใช้สำหรับในการเปรียบเทียบภาพถ่าย และอัลกอริทึม A*[4] ที่ใช้สำหรับค้นหาเส้นทางที่สั้นที่สุด

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึงการเตรียมการทดลองทั้งการจัดเตรียมส่วนฮาร์ดแวร์ ส่วนซอฟต์แวร์ สภาวะแวดล้อมในการทำการทดลอง ข้อมูลทดลองหรือทดสอบ การทำงานหรือการจำลองการทำงานของระบบ ผลการทดลอง ค่าสมรรถนะของระบบ การวัดประสิทธิภาพของระบบ และการวิเคราะห์ผลการทดลองหรือผลการทำงาน

บทที่ 5 บทสรุป กล่าวถึงบทสรุปของโครงงาน วิจารณ์สิ่งที่ได้รับจากโครงงาน ข้อจำกัด รวมถึงปัญหาอุปสรรคต่าง ของโครงงาน และข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนาต่อ

# บทที่ 2
# ทฤษฎีที่เกี่ยวข้อง

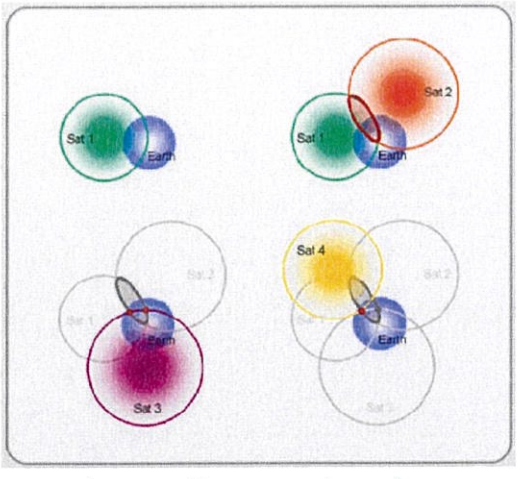
## 2.1 ระบบกำหนดตำแหน่งบนโลกหรือจีพีเอส (Global Position System: GPS)

GPS (Global Positioning System) หมายถึง ระบบที่บ่งบอกถึงตำแหน่งบนพื้นผิวโลก โดยจะใช้การคำนวนพิกัดจากสัญญาณนาฬิกา ซึ่งสัญญาณนาฬิกานั้นดึงมาจากดาวเทียม ที่โคจรอยู่รอบโลก โดยเราเรียกการคำนวนแบบนี้ว่า การคำนวนแบบ UTM ซึ่งจะทำให้เราได้ตำแหน่งที่แน่นอน ระบบ GPS สามารถบอกตำแหน่ง ณ จุดที่สามารถรับสัญญาณได้ทั่วโลก โดยเครื่องรับสัญญาณ GPS จะสามารถคำนวณความเร็ว และทิศทาง มาใช้ร่วมกับโปรแกรมแผนที่ เพื่อใช้ในการนำทางได้

สำหรับเทคนิคของการหาตำแหน่งของพิกัดนั้น เราจะสามารถคำนวณได้จาก ระยะทางระหว่างดาวเทียม กับเครื่อง GPS ของเรา ซึ่งเราจำเป็นต้องใช้ระยะทางที่ได้รับจากดาวเทียมอย่างน้อย 3 ดวง เพื่อจะนำมาคำนวนให้ได้ตำแหน่งที่แน่นอนที่สุด เมื่อเครื่อง GPS รับสัญญาณ จากดาวเทียมเรียบร้อยแล้ว จึงจะนำสัญญาณที่ได้มาทำการคำนวณหา ระยะทางระหว่างดาวเทียม กับตัวเครื่อง โดยใช้สมการ 2.1 คือ

$$\text{ความเร็ว} \times \text{เวลา} = \text{ระยะทาง} \tag{2.1}$$

ดาวเทียมทั้ง 3 ดวงจะส่งสัญญาณที่เหมือนกันมายังเครื่องรับด้วยความเร็วแสง (186,000 ไมล์ต่อวินาที) แต่ระยะเวลาที่ใช้รับสัญญาณจากดาวเทียมแต่ละดวงนั้นจะไม่เท่ากัน เนื่องจากระยะทางไม่เท่ากัน สามารถอธิบายได้จากรูป 2.1




**รูป 2.1 เทคนิคการระบุตำแหน่ง [5]**

จากรูป 2.1 อธิบายได้ว่าในการระบุพิกัด GPS นั้นจะยิ่งมีความแม่นยำมากขึ้นก็ต่อเมื่อมีการรับสัญญาณจากดาวเทียมหลายดวง โดยพิกัด GPS ที่แม่นยำนั้นจะถูกระบุในตำแหน่งที่วงโคจรของดาวเทียมตัดกันตามรูป 2.1

## 2.2 กูเกิลแมพจาวาสคริปต์เอพีไอ (Google maps Javascript API)

สำหรับบริการกูเกิลแมพจาวาสคริปต์เอพีไอนั้นเป็นไลบรารี่ที่ออกแบบมาเพื่อให้ผู้ใช้บริการสามารถกระทำการกับแผนที่ที่ทางกูเกิลให้บริการอยู่ผ่านทางภาษา HTML และจาวาสคริปต์ ซึ่งมีบริการที่สามารถใช้ได้ดังต่อไปนี้

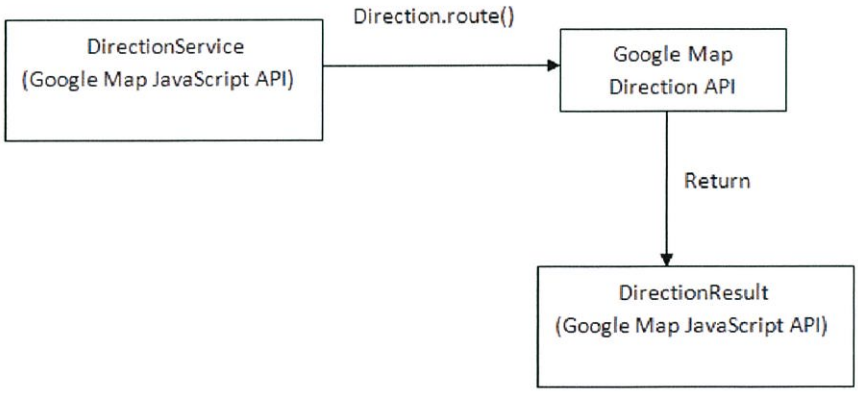
### 2.2.1 Web Service

ในเว็บแอพพลิเคชั่นของโครงงานนี้ได้มีการเรียกใช้บริการดังต่อไปนี้ ได้แก่

#### 2.2.1.1 ไดเรกชั่นเซอร์วิส (Direction service)

ไดเรกชั่นเซอร์วิส เป็นการเรียกใช้บริการค้นหาเส้นทางของกูเกิลสามารถทำได้โดยใช้ไดเรกชั่นเซอร์วิสออบเจ็กต์ (Direction Service Object) โดยต้องออบเจ็กต์ประเภทนี้จะถูกส่งไปยัง Google Maps API Direction Service และจะคืนค่ากลับมาเป็นเส้นทางที่ทำการค้นหา โดยค่าที่คืนกลับมาจะมีทั้ง เส้นทาง สถานะของคำร้องขอใน ด้วยออบเจ็กต์ประเภท DirectionResult และสามารถแสดงผลผ่านส่วน UI ได้ด้วยออพเจ็กต์ประเภท DirectionRender

ในการหาระยะทางนั้นสามารถค้นหาได้หลากหลายรูปแบบโดยอาจจะกำหนดจุดเริ่มต้น ปลายทาง หรือ ทางผ่าน(waypoints) โดยใช้เป็นข้อมูลประเภทอักขระ (Stirng) หรือจะกำหนดเป็นตำแหน่งละติจูด ลองจิจูด ย่อมได้เช่นกัน ซึ่งการทำงานของไดเรกชั่นเซอร์วิสนั้นสามารถเขียนแผนผังออกมาได้เป็น




**รูป 2.2 แผนผังการทำงานของไดเรกชั่นเซอร์วิส**

จากรูป 2.2 เป็นขั้นตอนการทำงานของไดเรกชั่นเซอร์วิส ซึ่งในการขอเส้นทางนั้นต้องส่งคำร้องขอผ่านไปยังไดเรกชั่นเซอร์วิสก่อน ซึ่งตัวไดเรกชั่นเซอร์วิสนั้นจะทำการส่งคำขอผ่านฟังก์ชั่น route() เพื่อส่งไปยังกูเกิลแมพไดเรกชันเอพีไออีกครั้ง และจะคืนค่าผลลัพธ์ที่ได้ออกมาทาง DirectionResult

**2.1.1.1.1 Direction Request**

ในการใช้บริการคำขอเส้นทางในกูเกิลแมพจาวาสคริปต์เอพีไอต้องทำการสร้างออบเจ็กต์ประเภทไดเรกชั่นเซอร์วิส และเรียกใช้ผ่านทาง DirectionsService.route() เพื่อเป็นการสร้างคำขอไปยังไดเรกรันเซอร์วิส และทำการส่งผ่านไปด้วยออบเจ็กต์ DirectionRequest โดยในออบเจ็กต์ประเภทนี้ เราสามารถกำหนดค่าต่างได้ดังต่อไปนี้

1) Origin: พิกัด (LatLng) | ชื่อ (String) | google.maps.Place (required) เป็นสถานที่ต้นทางโดยสามารถกำหนดค่าได้เป็นที่อยู่ หรือ ตำแหน่งพิกัด
2) Destination: พิกัด (LatLng) | ชื่อ (String) | google.maps.Place (required) เป็นสถานที่ต้นทางโดยสามารถกำหนดค่าได้เป็น ที่อยู่ (String) ออบเจ็กต์ google.maps.Place หรือ ตำแหน่งพิกัด (LatLng)
3) Travelmode: TravelMode เป็นรูปแบบวิธีที่จะเดินทางสามารถกำหนดได้โดย
   - google.maps.TravelMode.DRIVING (Default): เป็นการเดินทางโดยรถยนต์
   - google.maps.TravelMode.BICYCLING: เป็นการเดินทางโดยจักรยาน
   - google.maps.TravelMode.TRANSIT: เป็นการเดินทางโดยใช้ขนส่งสาธารณะ
   - google.mapsTravelMode.WALKING: เป็นการเดินทางด้วยการเดิน
4) transitOptions: TransitOptions (optional) เป็นการกำหนดค่าเพิ่มเติมในการขอเส้นทาง อาจจะเป็นเวลาที่จะออกเดินทาง รูปแบบของขนส่งสาธารณะที่ต้องการ โดยพารามิเตอร์นี้สามารถใช้ได้เฉพาะโหมด Transit เท่านั้น
5) unitSystem: UnitSystem เป็นตัวแปรที่กำหนดหน่วยวัดของระยะทางโดยสามารถ กำหนดได้เป็น
   - metric: แสดงผลเป็นหน่วยกิโลเมตร และเมตร

- imperial: แสดงผลเป็นหน่วยไมล์ และฟุต

6) durationInTraffic: Boolean (optional): เป็นพารามิเตอร์สำหรับลูกค้าของกูเกิลแมพเอพีไอเท่านั้น ซึ่งจะบอกถึงระยะเวลาที่ใช้ในการเดินทางโดยคำนวณจากสภาพการจราจรขณะนั้น
7) waypoints[]: DirectionsWaypoint (optional): เป็นพารามิเตอร์ที่สามารถกำหนดจุดที่เป็นทาง ผ่านได้โดยแต่ละจุดจะถูกเก็บไว้ในออบเจ็กต์ DirectionsWaypoints โดยในแต่ละออบเจ็กต์จะประกอบไปด้วย
   - location: เป็นสถานที่ที่ต้องการเก็บไว้ กำหนดได้เป็น ชื่อ(String) google.maps.Place หรือ ตำแหน่ง(LatLng)
   - stopover: มีค่าเป็นบูลีนใช้ในกรณีที่ต้องการให้หยุดเมื่อถึงจุด waypoints แต่ละจุด
8) optimizeWaypoint: Boolean (optional): เป็นพารามิเตอร์ที่ใช้กับพารามิเตอร์ waypoints โดยจะทำการลดเส้นทางให้สั้นที่สุด ในขณะที่ผ่านจุด waypoints ทั้งหมดถ้าตั้งค่าให้เป็น true
9) provideRouteAlternatives: Boolean (optional): ถ้าตั้งค่าพารามิเตอร์ตัวนี้เป็น true เส้นทางที่คืนค่ามาจะมีมากกว่าหนึ่งเส้นทางที่เป็นไปได้
10) avoidHighways: Boolean (optional): ถ้าตั้งค่าเป็น true เส้นทางที่ได้จะหลีกเลี่ยงการใช้เส้นทางหลวง
11) avoidTolls: Boolean (optional): ถ้าตั้งค่าเป็น true เส้นทางที่จะได้จะหลีกเลี่ยงทางด่วน
12) region: String (optional): ใช้ระบบ cctLD (“top-level domain”) เป็นตัวอักขระสองตัวใช้ในการกำหนดสัญชาติเช่น ไทย(th) ญี่ปุ่น(jp) เป็นต้น

**2.2.1.1.2 DirectionsResult Object**

หลังจากที่ได้ทำการร้องขอผ่านบริการไดเรกชั่นเซอร์วิส ไปแล้วทางไดเรกชั่นเอพีไอจะทำการคืนค่าออกมาด้วยออบเจ็กต์ที่ชื่อ DirectionsResult โดยใน DirectionsResult จะเก็บค่าสถานะของคำขอ และเส้นทางที่ร้องขอไว้ตอนต้นโดยใน DirectionResult จะประกอบไปด้วยข้อมูลต่อไปนี้

1) geocoded_waypoints[]: เป็นข้อมูลชนิดอาเรย์มีออบเจ็กต์ชื่อ DirectionGeocodedWaypointโดยในแต่ละออบเจ็กต์จะประกอบไปด้วยรหัสพิกัดของสถานที่ต้นทาง ปลายทาง และ Waypoints

2) routes []: เป็นข้อมูลอาเรย์เก็บออบเจ็กต์ชื่อ DirectionRoute โดย ในแต่ละออบเจ็กต์ประกอบไปด้วยเส้นทางจากสถานที่ต้นทางไปยังต้นทาง โดยจะมีหนึ่งเส้นทางต่อหนึ่งออบเจ็กต์ แต่ถ้าหากคำร้องขอมีการตั้งค่าพารามิเตอร์ provideRouteAlternatives เป็น true จะมีเส้นทางมากกว่าหนึ่งเส้นทางในออบเจ็กต์

**2.2.1.2 Distance Matrix Service**

เป็นบริการคำนวณระยะทาง และเวลาที่ใช้ในการเดินทางระหว่างสถานที่ต้นทางไปยังสถานที่ปลายทางตามรูปแบบการเดินทางที่เลือกไว้

**2.2.1.2.1 Distance Matrix Requests**

ในการติดต่อร้องขอกับ Distance Matrix Service เป็นบริการที่ สามารถติดต่อผ่านด้วยออบเจ็กต์ประเภท Google.maps.DistanceMatrixService โดยจะใช้ฟังก์ชั่น getDistanceMatrix() ในการกำหนดค่าเริ่มต้น เช่น สถานที่ต้นทาง สถานที่ปลายทาง รูปแบบการเดินทางไปยัง Service ที่ให้บริการซึ่งพารามิเตอร์ตั้งต้นมีดังต่อไปนี้

1) Origin: origin (required): เป็นตัวแปรอาร์เรย์เก็บสถานที่ต้นทาง โดยจะเป็น ตำแหน่งพิกัด (LatLng) หรือ ที่อยู่ (String)
2) Destinations: destination (required): เป็นตัวแปรอาร์เรย์เก็บสถานที่ปลายทางโดยจะเป็น ตำแหน่งพิกัด (LatLng) หรือ ที่อยู่ (String)
3) travelMode: TravelMode (optional): เป็นรูปแบบวิธีการเดินทางกำหนดได้โดย

- google.maps.TravelMode.DRIVING(Default): เดินทางด้วยรถยนต์
- google.maps.TravelMode.BICYCLING: เดินทางด้วยจักรยาน
- google.maps.TravelMode.TRANSIT: เดินทางด้วยขนส่งสาธารณะ
- google.maps.TravelMode.WALKING: เดินทางด้วยการเดิน

4) transitOptions: TransitOption (optional): เป็นการกำหนดค่าเพิ่มเติมในการขอเส้นทาง อาจจะเป็นเวลาที่จะออกเดินทาง รูปแบบของขนส่งสาธารณะที่ต้องการ โดยพารามิเตอร์นี้สามารถใช้ได้เฉพาะโหมด Transit เท่านั้น
5) unitSystem: UnitSystem เป็นตัวแปรที่กำหนดหน่วยวัดของระยะทาง โดยสามารถ กำหนดได้เป็น

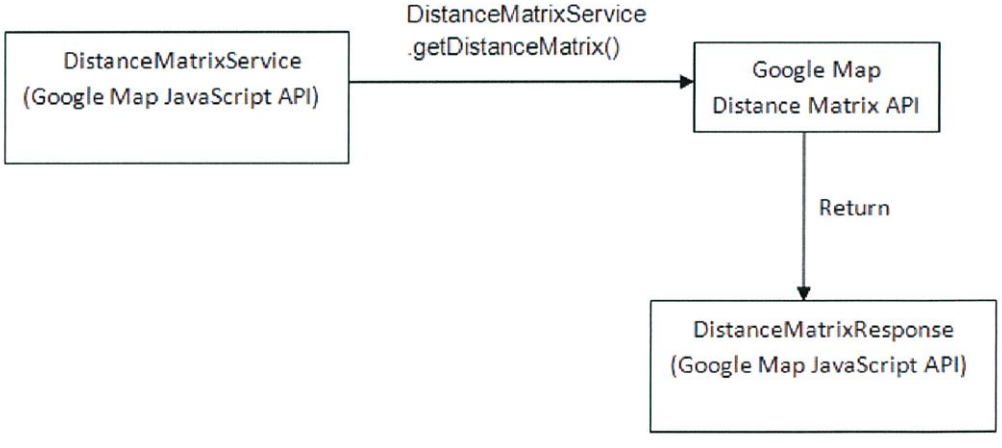
- metric: แสดงผลเป็นหน่วยกิโลเมตร และเมตร
- imperial: แสดงผลเป็นหน่วยไมล์ และฟุต

6) durationInTraffic: Boolean (optional): เป็นพารามิเตอร์สำหรับลูกค้าของกูเกิลแมพเอพีไอเท่านั้น ซึ่งจะบอกถึงระยะเวลาที่ใช้ในการเดินทางโดยคำนวณจากสภาพการจราจรขณะนั้น
7) avoidHighways: Boolean (optional): ถ้าตั้งค่าเป็น true เส้นทางที่ได้จะหลีกเลี่ยงการใช้เส้นทางหลวง
8) avoidTolls: Boolean (optional): ถ้าตั้งค่าเป็น true เส้นทางที่จะได้จะหลีกเลี่ยงทางด่วน

**2.2.1.2.2 Distance Matrix Responses**

เมื่อทำการส่ง Request ไปแล้ว Distance Matrix Service จะทำการคืนออบเจ็กต์ DistanceMatrixResponse โดยในแต่ละออบเจ็กต์จะมี สถานะคำร้องขอ ค่าระยะทาง และเวลาที่ใช้ในการเดินทาง โดยจะเก็บค่าสถานที่ต้นทาง (origin) ซึ่งใน DistanceMatrixResponse จะมีการเก็บข้อมูลประเภทอาเรย์ชื่อ rows โดยในแต่ละ rows จะมีอาเรย์ชื่อ element โดยในแต่ละ element จะมีค่าสถานะคำขอ ระยะทาง รวมไปถึงเวลาที่ใช้ในการเดินทางอยู่ภายใน rows

สำหรับแผนผังการทำงานของ Distance Matrix Service นั้นสามารถ อธิบายได้ดังรูป 2.3




**รูป 2.3 แผนผังการทำงานของ Distance Matrix Service**

จากรูป 2.3 การทำงานของ Distance Matrix Service หลังจากเรากำหนดค่าพารามิเตอร์ตั้งต้นผ่าน DistanceMatrixService.getDistanceMatrix() แล้วตัว javascript จะเป็นทำการเชื่อมต่อไปยัง Google Map Distance Matrix API แล้ว Google Map Distance Matrix API นี้จะคืนค่ากลับมาเป็นออบเจ็กต์ DistanceMatrixResponse

**2.2.1.2.3 ข้อจำกัดในการใช้ Distance Matrix Service**

1) ในแต่ละคำร้องขอสามารถใส่สถานที่ต้นทาง (origin) และสถานที่ปลายทาง (destination) ได้สูงสุดแค่ 25 สถานที่เท่านั้น
2) พารามิเตอร์ต่อการร้องขอหนึ่งครั้งห้ามเกิน 100 elements

**2.2.1.3 Google Street View Image API**

Google Street View Image API เป็นบริการของกูเกิลที่สามารถใช้รูปจาก Google Street View ได้ไม่ว่าจะเป็นรูปภาพแบบพาโนราม่า จวบจนถึงรูปภาพจากมุมต่าง โดยสามารถร้องขอได้ด้วย URL ต่อไปนี้

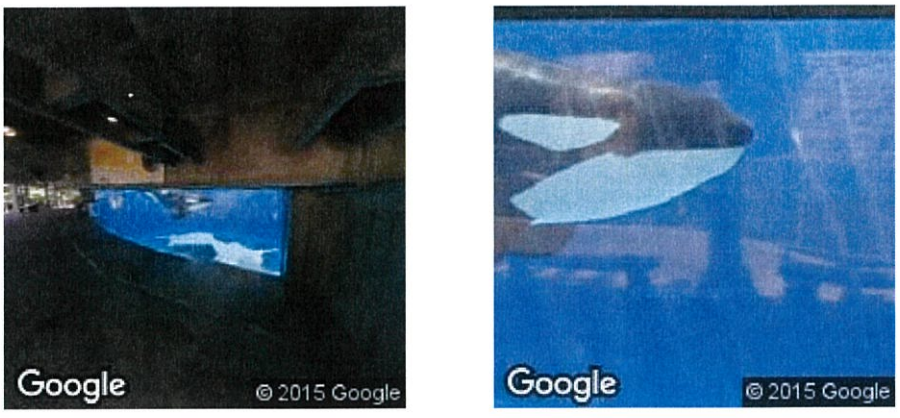
https://maps.googleapis.com/maps/api/streetview?parameters

ในลิ้งค์คำขอ 2.2 โดยในแต่ละคำขอจะแบ่งพารามิเตอร์ ออกจากกันด้วยตัวอักษร "&" ซึ่งในแต่ละคำขอนั้นจะมีพารามิเตอร์หลักที่ต้องมีทั้งหมดสองตัวด้วยกัน คือ

1) สถานที่ (Location): ใช้เป็นตัวอักษร (Chagrin Falls, OH) หรือ ตำแหน่งพิกัดละจิจูด ลองจิจูด (40.457375,-80.009353) โดย Google Street View Image API จะทำการแสดงรูปภาพพาโนราม่าที่เลือก หรือ pano: เป็น panorama id ของสถานที่นั้น
2) size – เป็นขนาดของรูปที่ต้องการ โดยกำหนดเป็น {ความกว้าง} x {ความสูง} เช่น size = 600x400 เป็นต้น
3) key – เป็นรหัสที่ทำให้สามารถเชื่อมต่อกับ API ได้

ในส่วนของพารามิเตอร์เสริมมีรายละเอียดดังต่อไปนี้

1) signature – เป็นลายเซ็นดิจิตอลที่ลงทะเบียนไว้กับทางกูเกิล ใช้สำหรับยืนยันสิทธิ์ในการใช้
2) heading – เป็นทิศของกล้องมีค่าตั้งแต่ 0 – 360 องศา โดยที่ 0 และ 360 คือทิศเหนือ 90 คือทิศตะวันออก 180 คือทิศใต้ และ 270 คือทิศตะวันตก ถ้าไม่ได้กำหนดพารามิเตอร์ชนิดนี้ไว้ ค่าจะถูกตั้งต้นเป็นด้านหน้าของถนนเส้นนั้น
3) fov (Field of View) – เป็นระยะการมองเห็นของรูป มีค่าได้สูงสุดถึง 120 องศา ใช้สำหรับการขยายหรือลดขนาดของรูป ยิ่งค่าน้อยจะได้รูปที่มีการขยายเยอะ มีค่าตั้งต้นที่ 90 องศา




ก) ข)

**รูป 2.4 ตัวอย่างจาก Google Street View API**

ก) ค่า fov เท่ากับ 120

ข) ค่า fov เท่ากับ 20

4) pitch – เป็นมุมก้มหรือมุมเงยจากกล้อง ถ้ามีค่าเป็นบวกแสดงว่าจะเป็นมุมเงยขึ้น ถ้าค่าเป็นลบ จะถือว่าเป็นมุมก้ม มีค่าเริ่มต้นเป็น 0

**2.2.1.3.1 ข้อจำกัดของ Google Street View Image API**

1) สามารถใช้งานได้ฟรีจนกระทั่งมีคำขอมากกว่า 25,000 ครั้งภายในหนึ่งวัน เป็นระยะเวลาติดต่อกันทั้งหมด 90 วัน
2) ขนาดของรูปภาพสูงสุดที่ขอได้คือ 640x640

## 2.3 กระบวนการประมวลผลภาพ (Image Processing)

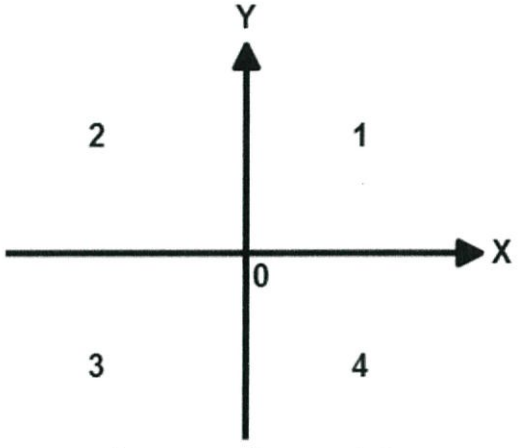
กระบวนการประมวลผลภาพคือกระบวนการที่เรานำภาพที่ต้องการมาทำการประมวลผล หรือทำการคิดคำนวณ หลังจากนั้นข้อมูลที่ได้หลังจากการประมวลผลภาพ จะออกมาเป็น 2 แบบ คือ เชิงคุณภาพ และเชิงปริมาณ โดยในเว็บแอพพลิเคชั่นของเรานั้นได้ทำการนำภาพถ่ายอาคารที่ผู้ใช้งานได้ทำการถ่าย มาทำการเปรียบเทียบกับภาพถ่ายอาคารของเราที่เราเก็บไว้อยู่ในฐานข้อมูล เพื่อทำการคำนวนให้ได้พิกัดตำแหน่งปัจจุบันของผู้ใช้ ตามทฤษฎีที่จะอ้างอิงถึงดังต่อไปนี้

**2.3.1 ระบบพิกัดในระนาบ**

**2.3.1.1 ระบบพิกัดฉาก (Cartesian Coordinates or Rectangular Coordinates)**

ระบบพิกัดฉากเป็นระบบพื้นฐานทางด้านเรขาคณิตวิเคราะห์ ซึ่งเรานำมาใช้เพื่อจับคู่แบบหนึ่งต่อหนึ่งระหว่างคู่อันดับกับจุดบนระนาบ

ระบบพิกัดฉากสามารถแบ่งได้เป็นสองมิติ และสามมิติ โดยระบบพิกัดฉากแบบสองมิติจะประกอบด้วยเส้นจำนวนในแนวนอน และแนวตั้ง อย่างละหนึ่งเส้น ตัดกันเป็นมุมฉากที่จุดกำเนิด (Origin) โดยที่ แกน X คือเส้นจำนวนในแนวแกนนอน และ แกน Y คือเส้นจำนวนในแนวตั้งว่า ซึ่งจุดบนแกน X ที่อยู่ทางด้านขวาของแกน Y จะมีค่าเป็นบวก และจุดที่อยู่ทางด้านซ้ายของแกน Y จะมีค่าเป็นลบ เช่นเดียวกันกับแกน Y จุดของแกน Y ถ้าอยู่ทางด้านบนของแกน X จะมีค่าเป็นบวก และจุดที่อยู่ทางด้านล่างของแกน X จะมีค่าเป็นลบ จากลักษณะดังกล่าว เราสามารถแบ่งระนาบออกได้เป็น 4 ส่วนด้วยกัน โดยแต่ละส่วนเราจะเรียกว่า “จตุภาค (Quadrant)” ตามรูป 2.5




**รูป 2.5 จตุภาคในระบบพิกัดฉาก**

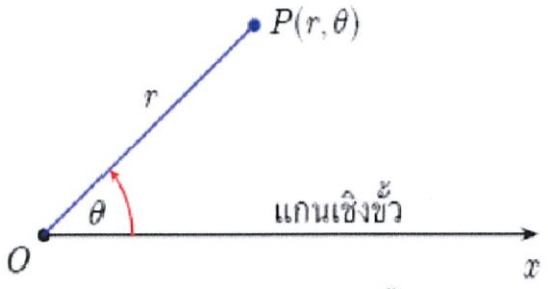
**2.3.2 ระบบพิกัดเชิงขั้ว (Polar Coordinates or Polar Grid)**

เรานำระบบพิกัดเชิงขั้วมาทำการพิจารณาเส้นโค้งนั้นเพราะว่าระบบพิกัดเชิงขั้วสามารถอธิบายเส้นโค้งบางประเภทที่มีสมการซับซ้อน ได้ดีกว่าระบบพิกัดฉาก ถ้าพิจารณาเส้นโค้งนั้นด้วยระบบพิกัดเชิงขั้ว

ระบบพิกัดเชิงขั้วประกอบไปด้วยจุดกำเนิดที่เรียกว่า Pole หรือ Origin ในระนาบ และลากเส้นจากจุด O ไปยังจุดใด เราจะเรียกเส้นดังกล่าวว่า “แกนเชิงขั้ว (Polar Axis)”

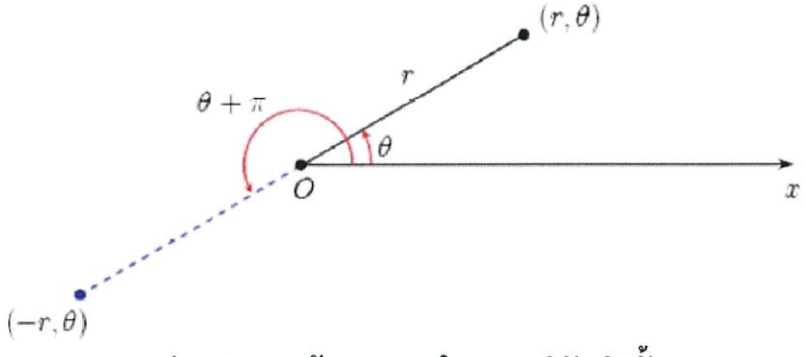
เมื่อทำการกำหนดให้ P เป็นจุดใดบนระนาบ และให้ r เป็นระยะจากจุด O ถึงจุด P ให้ $\theta$ เป็นมุมระหว่างแกนเชิงขั้วกับส่วนของเส้นตรง OP โดยมุม $\theta$ จะมีค่าเป็นบวกเมื่อวัดในทิศทวนเข็ม

นาฬิกา ดังรูป 2.6 จากนั้นจุด P จึงสามารถแทนได้ด้วยคู่ลำดับ (r, θ) และเรียก r กับ θ ว่า "พิกัดเชิงขั้ว (Polar Coordinates) ของจุด P" ในกรณีที่จุด P คือจุด O เราจะพบว่า r = 0 จะได้เป็น (0, θ) แทนขั้วสำหรับทุกค่าของ θ




**รูป 2.6 ระบบพิกัดเชิงขั้ว**

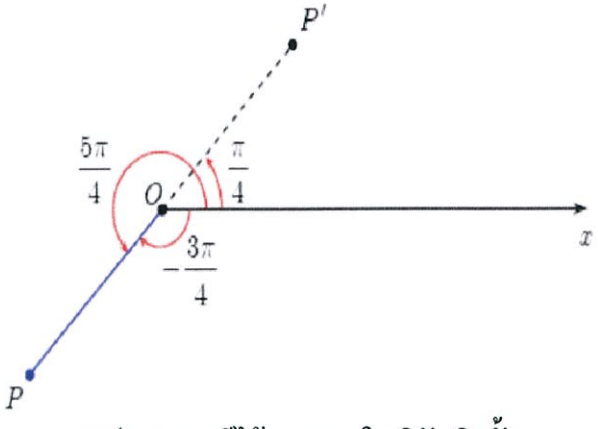
สามารถขยายความของระบบพิกัดเชิงขั้ว (r, θ) เมื่อ r < 0 โดยให้จุด (-r, θ) เป็นจุดที่ได้จากการสะท้อนจุด (r, θ) ผ่านจุด O โดยเส้นตรงที่เชื่อมจุดทั้งสองผ่านจุด O และจุดทั้งสองจุดอยู่ห่างจากจุด O ด้วยระยะ r เท่ากัน แต่อยู่กันคนละด้าน ดังรูป 2.7 สามารถสังเกตได้ว่าจุด (-r, θ) แทนจุดเดียวกับ (r, θ+π)




**รูป 2.7 การสะท้อนของจุดในระบบพิกัดเชิงขั้ว**

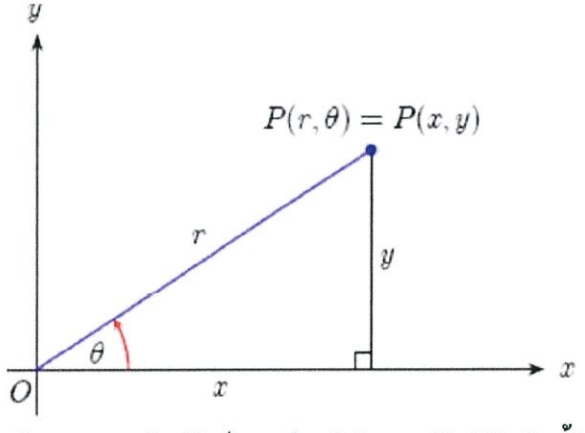
ในระบบพิกัดเชิงขั้วในจุดหนึ่ง สามารถมีจุดได้หลายพิกัดอย่างเช่นจุด P $(1,\frac{5\pi}{4})$ หรือแม้แต่ $(1,\frac{13\pi}{4})$ ก็ได้ นอกจากนั้นหากพิจารณาจุด P' $(1,\frac{\pi}{4})$ ซึ่งเป็นจุดที่สะท้อนของจุดผ่านจุด O จะเห็นว่าจุด P ยังสามารถแทนได้ด้วยพิกัด $(-1,\frac{\pi}{4})$ ได้ด้วย ซึ่งในระบบพิกัดฉากนั้นจุดทุกจัดจะมีพิกัดเดียวเท่านั้น

ในกรณีทั่วไปจุดที่แทนด้วยพิกัดเชิงขั้ว (r,θ) ยังสามารถแทนได้ด้วยพิกัด (r, θ+2nπ) หรือ (-r, θ+(2n+1) π) โดยที่ n เป็นจำนวนเต็ม




**รูป 2.8 การมีได้หลายจุดในพิกัดเชิงขั้ว**

โดยทั่วไปเราจะนิยมใช้มุมที่อยู่ระหว่าง 0 ถึง 2π เท่านั้น เนื่องจากค่ามุมที่เพิ่มเข้ามา 2nπ ทำให้จุดของพิกัดเชิงขั้วไม่เปลี่ยนแปลง และค่า r<0 ก็ทำให้จุด (r, θ) ไปซ้ำกับตำแหน่งอื่นได้




**รูป 2.9 ความสัมพันธ์ระหว่างพิกัดฉากกับพิกัดเชิงขั้ว**

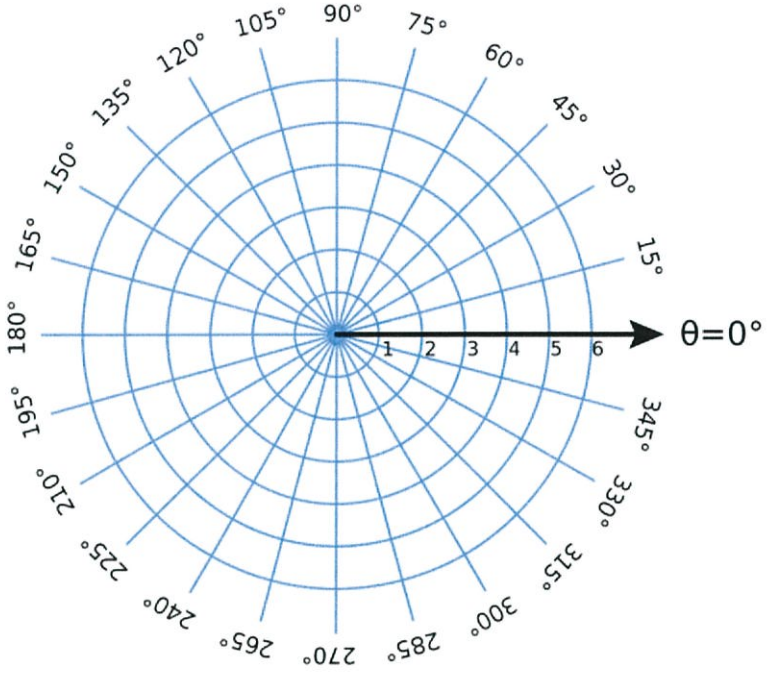
ความสัมพันธ์ระหว่างพิกัดฉากกับพิกัดเชิงขั้ว ดังรูป 2.9 เราสามารถทำการแปลง จุดจากระบบพิกัดฉากไปเป็นจุดในระบบพิกัดเชิงขั้วได้โดย ให้เรากำหนดจุด P ให้มีพิกัดเป็น (x,y) และในระบบพิกัดเชิงขั้วมีพิกัดเป็น (r, θ) จะได้รูปแบบสมการเป็น

$$x = r\cos\theta \text{ และ } y = r\sin\theta \qquad (2.2)$$

เนื่องจากค่าทุกค่าของ r และ θ จากสมการ 2.2 จะมีค่าเป็นจริงทำให้เราสามารถหาพิกัดฉากได้เมื่อทราบพิกัดเชิงขั้วในทางกลับกัน

$$r^2 = a^2 + b^2 \text{ และ } tan\theta = \frac{y}{x} \tag{2.3}$$

### 2.3.3 ระบบพิกัดลอกโพลา (Log Polar Coordinates or Log Polar Grid)




**รูป 2.10 ระบบพิกัดลอกโพลา**

ระบบพิกัดลอกโพลานั้นจะใกล้เคียงกับระบบพิกัดเชิงขั้ว โดยหลังการที่เปลี่ยนไปคือพิกัดเชิงขั้วในรัศมี (r) ถูกเปลี่ยนไปเป็น ใช้เป็นรูปแบบลอกการิทึมสเกล (Log Scale) ค่า θ นั้นจะเหมือนกับ θ ในระบบพิกัดเชิงขั้ว โดยสามารถเขียนคู่อันดับในระบบพิกัดลอกโพลาได้เป็น (ρ , θ)

ρ เป็นค่าลอกการิทึมของระยะตั้งแต่จุด Origin ถึงจุดหนึ่ง โดยมีมุมเป็น θ สำหรับลอกการิทึมสเกลในระบบพิกัดลอกโพลานั้นจะได้ดั่งสมการ 2.4

$$r = e^{\rho} \tag{2.4}$$

เราสามารถแปลงพิกัดในระบบฉากมาเป็นพิกัดลอกโพลาได้โดยสมการ 2.5

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



$$\rho = log\sqrt{x^2 + y^2} \text{ และ } \theta = arctan(\frac{y}{x}) \quad (2.5)$$

เราสามารถแปลงพิกัดลองโพลาไปเป็นพิกัดฉากได้โดยสมการ 2.6

$$x = e^{\rho}cos\theta \text{ และ } y = e^{\rho}sin\theta \quad (2.6)$$

**2.3.4 การเปลี่ยนแปลงทางเรขาคณิต (Geometric Transformation)**

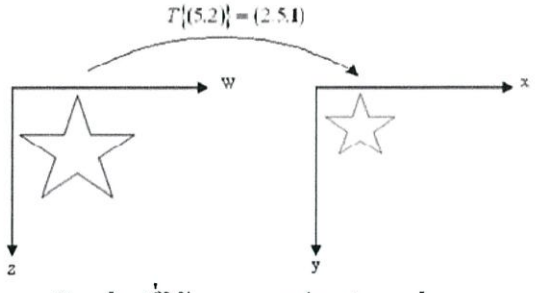
เราสามารถแบ่งการเปลี่ยนแปลงทางเรขาคณิตที่เกี่ยวข้องกับการประมวลผลภาพทางดิจิตอลนั้นได้เป็น 2 ส่วนคือ

**2.3.4.1 การแปลงเรขาคณิตเชิงตำแหน่ง (Geometric Spatial Transformation)**

ใช้เทคนิคในการเปลี่ยนแปลงตำแหน่งในส่วนของ ลักษณะ รูปร่าง หรือขนาดของวัตถุใดก็ตาม โดยจะอยู่ภายในขอบเขตที่เรากำหนด การเปลี่ยนแปลงของวัตถุนั้น สามารถเกิดได้หลายรูปแบบ เช่น การเลื่อนตำแหน่ง (Translation) การหมุน (Rotation) การย่อขนาดของภาพ (Rescale) ฯลฯ โดยให้อยู่ในรูปของสมการในการแปลงจากวัตถุจากสถานะหนึ่งไปสู่สถานะหนึ่ง โดยสมการที่ใช้ในการแปลงวัตถุจะแตกต่างกันตามรูปแบบที่เราต้องการต่างกันออกไป ตัวอย่างของการแปลงเรขาคณิตเชิงตำแหน่ง สมมติให้ภาพ f นิยามอยู่ในระบบพิกัด (w,z) ต้องการแปลงพิกัดเพื่อสร้างภาพ g ซึ่งอยู่ในพิกัด (x,y) เราสามารถแสดงได้ดังสมการ 2.7

$$(x,y) = T\{(w,z)\} \quad (2.7)$$

เมื่อ T เป็นสมการในการแปลงวัตถุโดยสามารถเขียนอยู่ในรูปเมตริกซ์ได้เช่น $(x,y) = T\{(w,z)\} = (w2,z2)$ การแปลงนี้เป็นการย่อขนาดของภาพ f โดยย่อด้านกว้าง และด้านยาวของภาพ f ลงไปอย่างละครึ่ง




**รูป 2.11 ตัวอย่างที่ได้จากการแปลงตำแหน่งภาพ**

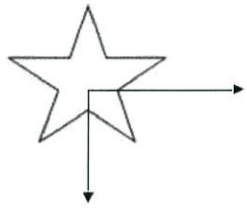
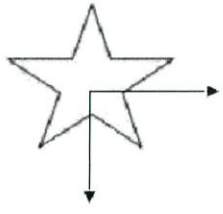
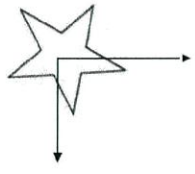
144384

รูปแบบแรกเป็นรูปแบบโดยทั่วไปของการแปลงเรขาคณิตเชิงตำแหน่ง เราสามารถเขียนให้อยู่ในรูปเมตริกซ์ ได้ดังสมการ 2.8

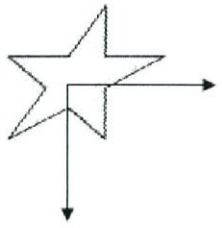
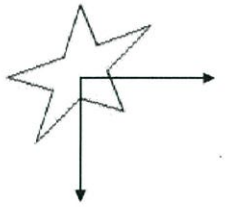
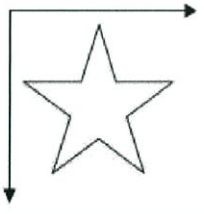
$$[x \quad y \quad z] = [w \quad z \quad 1]T = [w \quad z \quad 1]\begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix} \tag{2.8}$$

โดยที่เราสามารถทำการย่อหรือขยายขนาด หมุน เลื่อนตำแหน่ง รวมไปถึงการบิดกับเซ็ตของจุดในภาพตามที่ต้องการ โดยขึ้นอยู่กับค่าที่ถูกตั้งไว้ในเมตริกซ์ T ตาราง 2.1 จะแสดงวิธีการตั้งค่าในสมาชิกแต่ละตัวของเมตริกซ์ T เพื่อใช้ในการแปลงแต่ละประเภทดังนี้

**ตาราง** 2.1 Affine Transformation

| ชนิดการเปลี่ยนแปลง | เมตริกซ์ | สมการพิกัด | Diagram |
|---|---|---|---|
| Identity | $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | X = w<br>Y = z |  |
| Scaling | $\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = s_x w$<br>$y = s_y z$ |  |
| | $S_x$ = ค่าการขยายหรือลดขนาดในแนวแกนนอน<br>$S_y$ = ค่าการขยายหรือลดขนาดในแนวแกนตั้ง | | |
| Rotation | $\begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = wcos\theta - zsin\theta$<br>$y = wsin\theta + zcos\theta$ |  |
| | เมื่อ $\theta$ เป็นค่าของมุมการหมุน | | |

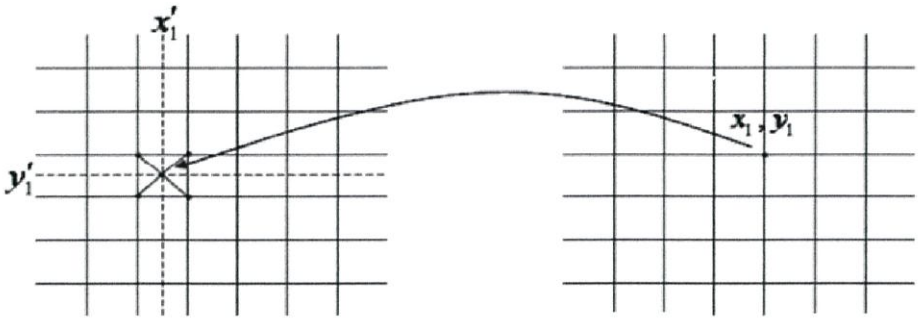
**ตาราง 2.1 Affine Transformation (ต่อ)**

| ชนิดการเปลี่ยนแปลง | เมตริกซ์ | สมการพิกัด | Diagram |
|---|---|---|---|
| Shear (horizontal) | $\begin{bmatrix} 1 & 0 & 0 \\ \alpha & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = w + \alpha z$<br>$y = z$ |  |
| | เมื่อ $\alpha$ คือค่าสัมประสิทธิ์ในแนวแกนนอน | | |
| Shear (vertical) | $\begin{bmatrix} 1 & \beta & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = w$<br>$y = \beta w + z$ |  |
| Translation | $\begin{bmatrix} 1 & 0 & \delta x \\ 0 & 1 & \delta y \\ 0 & 0 & 1 \end{bmatrix}$ | $x = w + \delta x$<br>$y = \delta y + z$ |  |

**2.3.4.2 อินเตอร์โพเลชัน (Interpolation)**

อินเตอร์โพเลชันเป็นกระบวนการที่ทำนำค่าอ้างอิงของพิกเซลในภาพเดิมมาทำการคำนวณตำแหน่งของภาพใหม่ในแต่ละพิกเซล ซึ่งต้องมีการประมาณค่าของพิกเซลในภาพ เช่น การขยายภาพ กระบวนการอินเตอร์โพเลชันจะถูกนำมาใช้เพื่อหาค่าของพิกเซลที่เพิ่มขึ้นกว่าเดิม

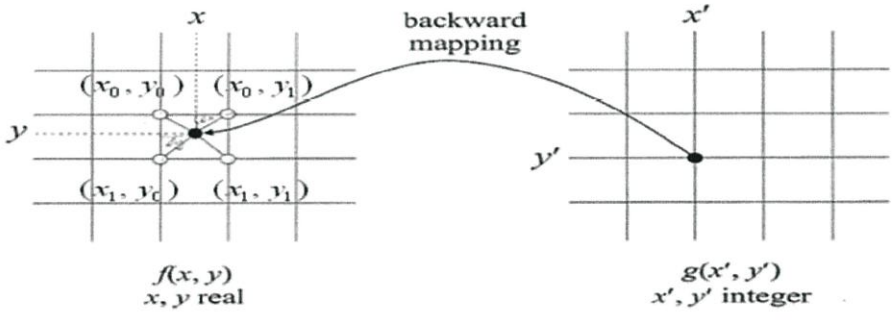
เราสามารถแบ่งวิธีการคำนวนได้เป็น 2 กลุ่มคือ วิธีการคำนวณด้วยเทคนิค Forward Mapping และ Inverse Mapping ซึ่งในการเปรียบเทียบภายถ่ายอาคาร กับภาพถ่ายในฐานข้อมูลนั้นจะใช้อินเตอร์โพเลชันโดยใช้การคำนวณรูปแบบ Inverse Mapping ซึ่งหลักการทำงานของเทคนิคนี้คือจะคำนวณในลักษณะย้อนกลับว่าแต่ละจุดในภาพใหม่ จะนำจุดจากภาพเก่าณ ตำแหน่งใดมาแทน ดังรูป 2.12




**รูป 2.12 การทำ Inverse Mapping**

ในการประมาณค่าแบบ Inverse Mapping ที่ใช้ในการประมวลผลภาพแบ่งได้เป็นหลายวิธี ซึ่งแต่ละวิธีจะมีวิธีการที่คล้ายคลึงกันโดยจะกำหนดค่าของพิกเซลจากการหาตำแหน่งพิกเซลในภาพผลลัพธ์ที่สอดคล้องกันบนภาพเริ่มต้น และมีการคำนวณค่าเฉลี่ยบนเซตของพิกเซลในบริเวณนั้น โดยค่าถ่วงน้ำหนักจะขึ้นกับระยะห่างระหว่างพิกเซลบนภาพเริ่มต้นที่นำมาคำนวณกับพิกเซลภาพผลลัพธ์ ซึ่งเราสามารถพิจารณาได้ 3 แบบ ได้แก่ Zero-order Interpolateion, First-order Interpolation และ High-order Interpolation ซึ่งในโครงงานนี้จะใช้การคำนวณแบบ First-order Interpolation

First-order Interpolation หรือ Bilinear Interpolation เป็นการประมาณค่าโดยค่าของพิกเซลในภาพผลลัพธ์จะถูกกำหนดจากการหาค่าเฉลี่ยแบบถ่วงน้ำหนัก โดยจุดภาพข้างเคียงที่ใกล้ที่สุดจำนวน 4 หรือ 2x2 พิกเซล ข้อมูลจะเรียบ และต่อเนื่องกัน




**รูป 2.13 การทำ Inverse Mapping แบบ First-order**

ซึ่ง First-order Interpolation มีรูปแบบสมการเป็นดังสมการ 2.9

$$g(x',y') = f(x_0 + y_0) + [f(x_1,y_0) - f(x_0,y_0)]\Delta x + [f(x_0,y_{1)} - f(x_0,y_0)]\Delta y$$
$$+[f(x_1,y_1) + f(x_0,y_0) - f(x_0,y_1) - f(x_1,y_0)]\Delta x\Delta y \quad (2.9)$$

โดยที่ $\Delta x = x - x_0$ และ $\Delta y = y - y_0$

$f(x_1,y_1) + f(x_0,y_0) - f(x_0,y_1) - f(x_1,y_0)$ เป็นจุดใกล้เคียงรอบจุดที่คำนวณได้ทั้งหมด 4 จุด

**2.3.5 SIFT (Scale Invariant Feature Transform)**

นำมาใช้ในการวิเคราะห์เพื่อหาจุดเด่นของรูปภาพที่ได้รับเข้ามา โดยไม่ขึ้นกับขนาดหรือทิศทางของวัตถุในภาพ ซึ่งลักษณะเด่นที่ได้จากวัตถุลักษณะเดียวกันจะมีลักษณะที่คล้ายคลึงกัน โดยวิธีการของ SIFT จะประกอบด้วย 4 ขั้นตอนที่สำคัญได้แก่

**2.3.5.1 การหาปริภูมิค่าในมิติขนาดและระยะทาง (Scale-space extrema detection)**

เราจะเริ่มด้วยการหาลักษณะเด่นของภาพที่ไม่ขึ้นกับขนาดหรือทิศทาง โดยจะใช้วิธีการเบลอ (Blur) ภาพด้วยฟังก์ชันเกาส์เซียน (Gaussian Function) ไปในแต่ละขั้น (Octave) ซึ่งในแต่ละ Octave ก็จะมีรูปหลายระดับการ Blur ซึ่งมีขนาดรูปที่เท่ากัน โดยที่แต่ละระดับการ Blur จะ Blur จากระดับปกติแล้วค่อยเพิ่ม σ (Scale parameter) ซึ่งจะมีผลทำให้ภาพ Blur มากยิ่งขึ้น แล้วทำซ้ำกับ Octave ต่อไป โดย Octave ต่อไปก็จะมีขนาดของรูปเป็นครึ่งหนึ่งของ Octave เดิม

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \quad (2.10)$$

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2.11)$$

โดยที่

L คือ ภาพที่ผ่านการ Blur

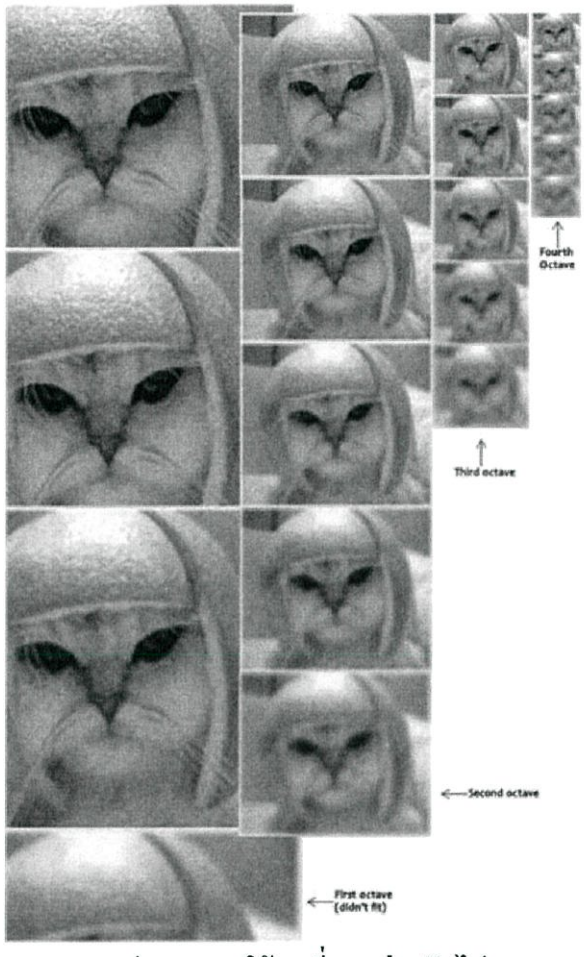
G คือ ตัวกรองแบบ Gaussian ที่มีขนาด σ

I คือ ภาพต้นฉบับ

x,y คือ พิกัดบนภาพ

σ คือ ตัวแปรขนาดของการ Blur ยิ่งมีค่ามาก ยิ่ง Blur มาก

* คือ การคอนโวลูชั่นบนภาพ I โดยใช้ Gaussian Blur G




**รูป 2.14 การใช้ σ ที่แตกต่างกันไป**

ในแนวตั้งซึ่งได้ผลลัพธ์คือภาพที่มีความ Blur มากขึ้น และการขยับ Octave ขึ้นไปทีละขั้นโดยมีขนาดรูปภาพเป็นครึ่งหนึ่งของ Octave ก่อนหน้า ซึ่งในภาพนี้ได้แสดง 4 Octave โดยแต่ละ Octave มี 5 รูปที่เหมือนกันแต่ต่างกันที่ขนาดของการ Blur

**2.3.5.2 การกำหนดตำแหน่งจุดสนใจ (Keypoint localization)**

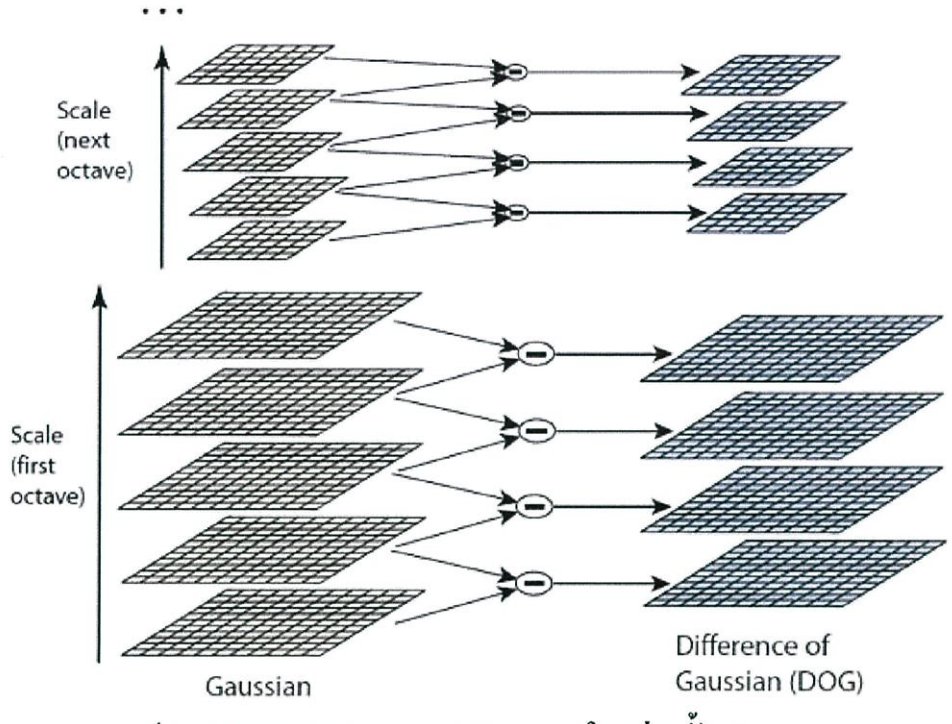
จากการหาปริภูมิค่าในมิติขนาดและระยะทางในขั้นตอนก่อนหน้า เราจะนำมาหาจุดที่น่าจะเป็นลักษณะสำคัญของภาพ โดยเราจะทำการจับคู่ภาพของแต่ละ Octave มาหาความต่างของแต่ละผลลัพธ์ที่ได้จากการ Blur (Difference of Gaussian : DoG) โดยจะทำทุกภาพในแต่ละ Octave และทำซ้ำจนครบทุก Octave

$$D(x, y, \sigma) = \big(L(x, y, k\sigma) - L(x, y, \sigma)\big) * I(x, y) \qquad (2.12)$$

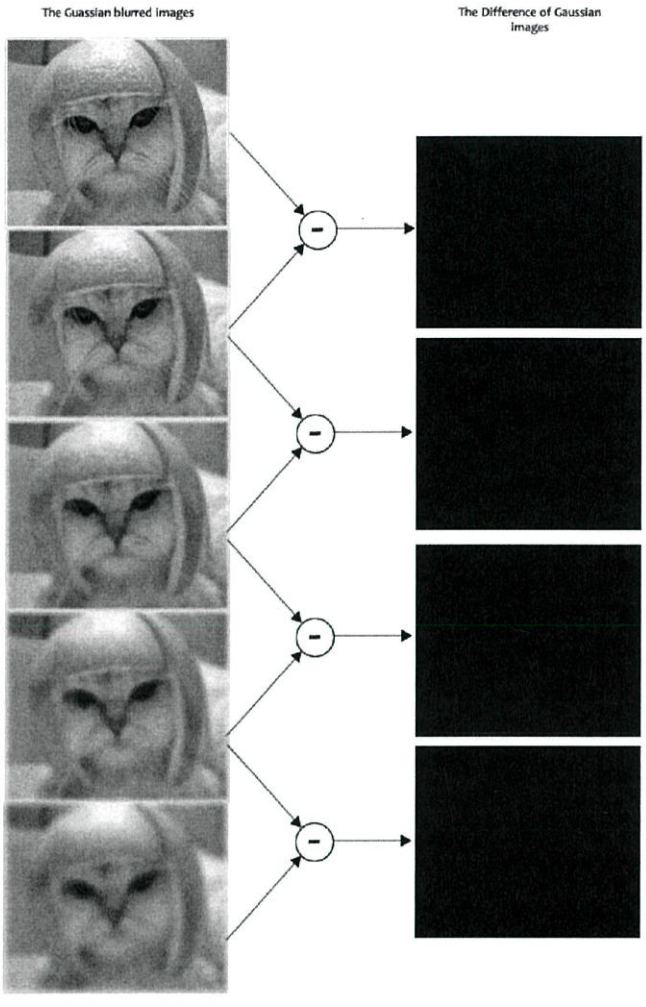
โดยที่

$L(x, y, \sigma)$ คือ คอนโวลูชั่นของภาพตั้งต้น $I(x, y)$ ที่ทำ Gaussian Blur ที่ Scale $\sigma$ ซึ่งสามารถเขียนสมการได้เป็นดังสมการ 2.13

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y) \tag{2.13}$$



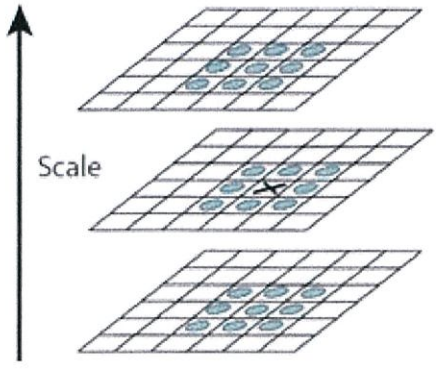

**รูป 2.15 การหา Difference of Gaussian ในแต่ละขั้นของ Octave**



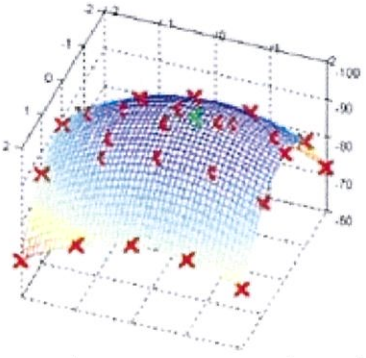

**รูป 2.16 ผลลัพธ์จากการทำ Difference of Gaussian ในแต่ละขั้นของ Octave**

**2.3.5.3 Locate maxima/minima in DoG images**

เป็นการประมาณหาค่าจุดภาพ (Pixel) ที่มีค่าสูงสุดหรือต่ำสุดแบบหยาบ เมื่อทำการเปรียบเทียบกับจุดรอบบ้านรอบข้าง 26 จุด ซึ่งแบ่งเป็น 8 จุดที่อยู่ติดกับจุดที่พิจารณา และ 9 จุดบนและล่างที่อยู่ติดกับจุดที่พิจารณา




**รูป 2.17 จุด X ที่เป็นจุดอ้างอิงเปรียบเทียบกับจุดรอบข้าง 26 จุด**



**รูป 2.18 จุดสีแดงที่เป็นจุดสูงสุดหรือต่ำสุดที่เป็น pixel**
**แต่จุดที่สูงสุด คือ จุดสีเขียว ซึ่งเป็น subpixel**

**2.3.5.4 Find subpixel maxima/minima**

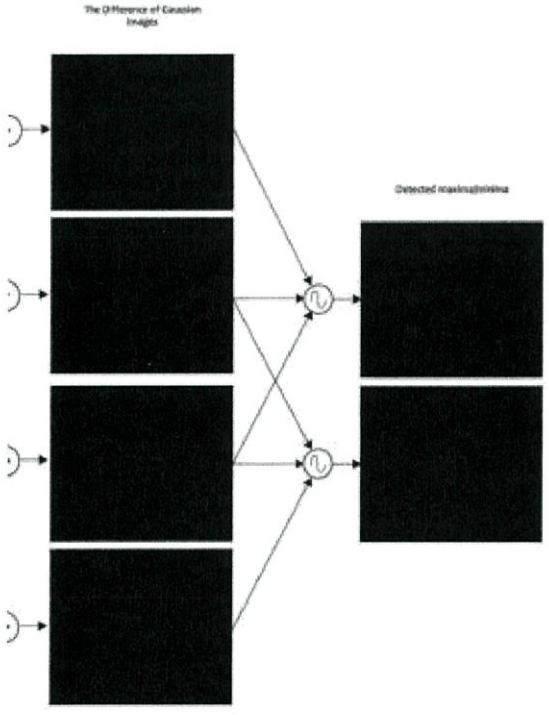
เราจะใช้สมการของ Taylor เพื่อทำการประมาณหา subpixel ที่เป็นจุดที่มีค่าสูงสุดหรือต่ำสุดอย่างแท้จริง จาก pixel ที่เราได้เลือกมาจากขั้นตอนก่อนหน้า

$$D(x) = D + \frac{\partial D^T}{\partial x}x + \frac{1}{2}x^T * \frac{\partial^2 D}{\partial x^2}x \tag{2.14}$$

โดยที่

$D(x)$ เป็น derivatives ที่ถูกคัดกรองมาจาก Candidate Keypoint

$x$ เป็น offset ของจุด (x,y,σ)



**รูป 2.19 ผลลัพธ์จากการทำ Locate maxima/minima กับ 4 DoG images**

จากขั้นตอนที่ผ่านมาเราจะได้ Keypoints จำนวนมาก ซึ่งบางจุดจะวางเรียงตัวอยู่บนเส้นขอบ (Edge) หรือบางจุดมี ความคมชัด (Contrast) โดยจุดเหล่านี้จะเป็นจุดที่เราจะทำการคัดออก เพื่อจะได้ผลลัพธ์สุดท้ายที่แต่ละจุดจะมีความเหมาะสมที่จะเป็น Keypoint มากที่สุด ประกอบด้วย 2 ขั้นตอน คือ

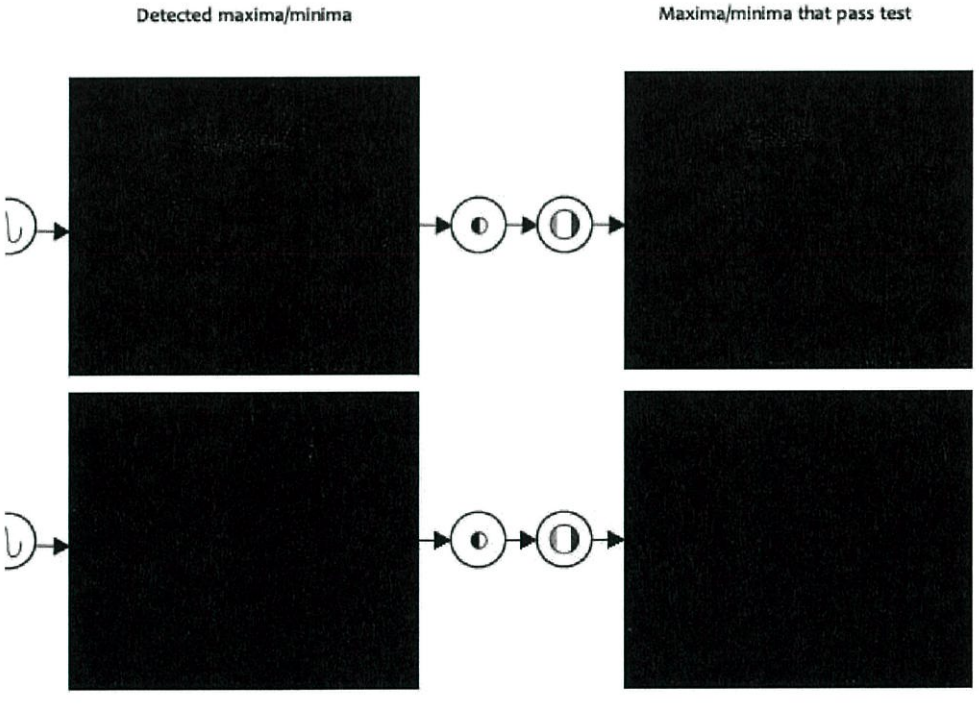
**2.3.5.4 Removing low contrast features**

เราจะทำการหาค่าความเข้ม(Intensity) ของจุดที่เป็น subpixel ซึ่งเราได้ประมาณมาจากขั้นตอนที่ผ่านมา นำมาเปรียบเทียบกับค่าที่เรากำหนดค่าหนึ่งโดยถ้ามีค่าน้อยกว่าก็จะทำการคัดจุดนั้นออกจาก Keypoints

**2.3.5.4.1 Removing edges**

เราจะทำการคำนวณ gradients 2 ทิศ ของจุดที่เป็น keypoint ซึ่งตั้งฉากกันและกัน โดยเราสามารถสรุปความสัมพันธ์บนภาพจากจุดทั้ง 2 ได้เป็น 3 แบบ คือ

1) พื้นที่ราบ (Flat region) จะมี gradient ที่มีค่าน้อยทั้งคู่
2) ขอบ (Edge) จะมี gradient ที่มีค่ามากสำหรับตัวที่ตั้งฉากกับขอบ และมี gradient ที่มีค่าน้อยสำหรับตัวที่ขนานไปกับขอบ
3) มุม (Corner) จะมี gradient ที่มีค่ามากทั้งคู่ โดยเราจะพิจารณา keypoint ที่มีลักษณะเป็นมุมเป็นหลัก ซึ่งถ้ามีค่าของ gradient ที่มากพอทั้งคู่ เราก็จะนิยามให้เป็น keypoint แต่ถ้าไม่มากพอก็จะถูกตัดทิ้ง และในขั้นตอนการพิจารณาหา keypoint ที่เป็นมุมนี้ เราสามารถใช้ Hessian Matrix เข้ามาเพื่อช่วยในการหาว่าเป็นมุมหรือไม่ ได้ดียิ่งขึ้น




**รูป 2.20 ผลลัพธ์ที่ได้หลังจากทำการหา keypoint จากการทดสอบ Edge และ Contrast ที่เหมาะสม**

**2.3.5.5 การกำหนดทิศทางของจุดสนใจ (Orientation assignment)**

ทำการเก็บรวบรวมขนาด $m(x,y)$ และ ทิศทาง $\theta(x,y)$ของ gradient ของบริเวณรอบ keypoints เพื่อที่จะกำหนดทิศทางให้ keypoint

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (2.15)$$

โดย

$m(x,y)$ เป็น Gradient Magnitude

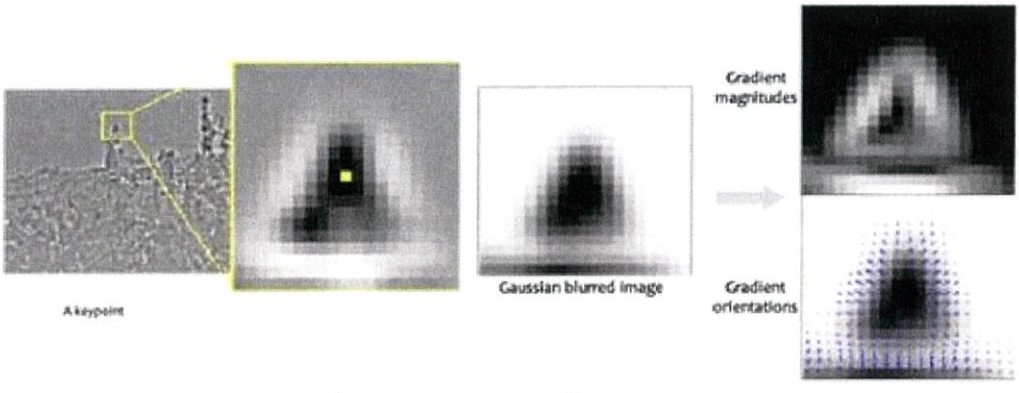
$L(x,y)$ เป็น ภาพตั้งต้นที่ผ่านการทำ Gaussian Blur ที่ Scale σ

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))) \quad (2.16)$$

โดย

$\theta(x,y)$ เป็น Orientation

$L(x,y)$ เป็น ภาพตั้งต้นที่ผ่านการทำ Gaussian Blur ที่ Scale σ




**รูป 2.21 การขนาดและทิศทางของ keypoint**

จากนั้นจะนำเอาขนาดและทิศทางของ gradient ของ pixel รอบ keypoint มาทำการสร้างภาพแท่งความถี่ (Histogram) ซึ่งจะมีแกน x เป็นช่วงขององศา และแกน y เป็นขนาดของ gradient ของ pixel นั้นรวมกับ pixel ที่แบ่งแยกตามแต่ละช่องของแกน x เดียวกัน ต่อมาจะทำการเพิ่ม keypoint โดยมีเงื่อนไขว่า ถ้า keypoint ใดมี peak ของ histogram ที่มีขนาดมากกว่า 80% ของ peakที่สูงสุด จะแบ่งออกเป็น keypoint ใหม่

**2.3.5.6 การสร้างคำอธิบายลักษณะเด่นของภาพ (Keypoint descriptor)**

เราจะทำการสร้าง 16×16 window (window คือ บริเวณที่จะทำการเก็บข้อมูลทิศทาง ซึ่งจะมีขนาดคือ 1.5*σ) รอบ keypoint และทำการแบ่งออกเป็น 4×4 windows ทั้งหมด 16 ชุด โดยในแต่ละชุดจะทำการคำนวณหาขนาดและทิศทางของ gradient แล้วนำมาสร้างเป็น histogram ที่มีขนาด 8 bin (แกน x แบ่ง 8 ช่วง ช่วงละ 45 องศา) โดยที่ขนาดของ histogram ในแต่ละส่วนจะขึ้นกับ ขนาด*ตัวถ่วงน้ำหนัก(ระยะทางที่ห่างจาก keypoint) ดังนั้นเมื่อเราทำการคำนวณเสร็จสิ้น ผลลัพธ์ที่ได้คือ 4x4x8= 128 ซึ่งจะทำการ normalize ต่อ และจะใช้เป็น feature vector ของแต่ละ keypoint นั้นเอง

**2.3.5.7 Post Estimation**

เป็นกระบวนที่นำมาใช้เพื่อทำการหาตำแหน่งและทิศทางการวางตัวของกล้อง โดยจะนำความสัมพันธ์ของจุดที่จะใช้เป็น feature point หรือ key point บนภาพถ่ายสองมิติของวัตถุชิ้นเดียวกันจากหลายมุมมองมาช่วยในการวิเคราะห์เพื่อคำนวณตำแหน่งและทิศทางได้อย่างถูกต้องและมีความใกล้เคียงมากที่สุด โดยในการคำนวณโมเดลของกล้องนี้ เราจะมีพารามิเตอร์ที่สนใจ 2 ตัวด้วยกัน ได้แก่

**2.3.5.7.1 Internal parameter**

จะแทนด้วย $K \in R^{3*3}$ โดยค่า K จะขึ้นอยู่กับคุณสมบัติของตัวกล้องโดยตรง ได้แก่ ความยาวโฟกัส คุณสมบัติของเลนส์ ซึ่งค่า K นี้ เราจะสามารถประมาณได้ด้วยการใช้เทคนิค Camera Calibration (การปรับมาตรฐานของภาพที่ได้จากกล้อง เพื่อให้ได้ภาพที่ไม่บิดเบือน) โดยจะนำมาใช้ในการเปลี่ยนจากพิกัดสามมิติของกล้องให้กลายเป็นพิกัดสองมิติบนภาพ

**2.3.5.7.2 External parameter**

จะแทนด้วย $[R|t] \in R^{3*4}$ โดยค่า R จะแทนการหมุน และค่า t จะแทนการเลื่อนขนาน โดยค่าทั้งสองจะใช้แทนการเปลี่ยนจากตำแหน่งของกล้องบนพิกัดสามมิติในโลกจริงมายังพิกัดของกล้องในระบบซึ่งเราสามารถสรุปเป็นสมการของการเปลี่ยนจากพิกัดสามมิติในโลกจริง($P \in R^4$)ให้พิกัดสองมิติบนภาพ ($P \in R^3$) คือ

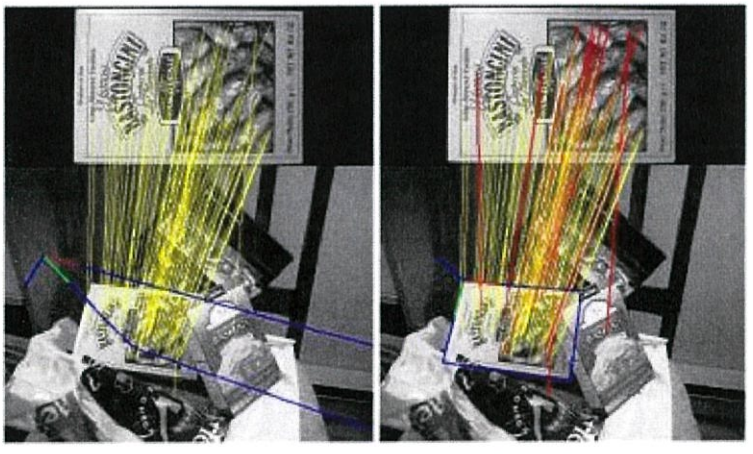
$$p = K[R|t]P \tag{2.17}$$

$$sm' = A[R|t]M' \tag{2.18}$$

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & tr_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.19}$$

ซึ่งจากความสัมพันธ์ของจุดที่จะใช้เป็น feature point หรือ key point นี่เอง เราก็จะใช้ความสอดคล้องของจุดเหล่านี้ $(C = \{P_i \leftrightarrow p_i\})$ โดยใช้สมการข้างต้น เพื่อทำการประมาณการตำแหน่งและทิศทางของกล้อง

โดยทั่วไปแล้ววิธีการหาจุดที่จะเป็น feature point หรือ key point นั้น จะมีด้วยกันหลากหลายวิธี โดยวิธีที่ค่อนข้างเป็นที่นิยม คือ SIFT และ Randomized Tree และเมื่อทำการหาความสอดคล้องได้จนครบทุกจุดในภาพแล้ว อาจพบว่ามีจุดบางจุดที่เป็นจุดที่อาจจะอยู่นอกความสนใจในการพิจารณาของเรา ซึ่งมีผลทำให้เกิดความผิดพลาดในกระบวนการประมาณของเรา เราจึงต้องทำการวิเคราะห์เพื่อคัดจุดที่อยู่นอกความสนใจของเราออก โดยใช้วิธีการ คือ RANSAC (Random Sample Consensus) ซึ่งเป็นวิธีการทำซ้ำ (Iterative Method) จนกว่าจะได้ผลลัพธ์ที่มีความเป็นไปได้สูงสุดขึ้นกับจำนวนรอบการทำซ้ำนั่นเอง



**รูป 2.22 ผลลัพธ์ของการหาตำแหน่งและทิศทาง ของการวางตัวของกล้องจากหนังสือที่ต้องการ(เส้นขอบสีน้ำเงิน)**

โดยภาพทางด้านซ้ายจะเป็นภาพที่ยังไม่ได้ใช้เทคนิค RANSAC ซึ่งจะเห็นได้ว่าจากการใช้ SIFT หา feature point ที่สอดคล้องกัน(เส้นสีเหลือง)ระหว่างภาพต้นฉบับ(ภาพด้านบน)กับภาพนำเข้า(ภาพด้านล่าง) ก่อให้เกิดความผิดพลาดในการประมาณการวางตัวของกล้องเป็นอย่างมาก ส่วนภาพทางด้านขวาเป็นการใช้เทคนิค RANSAC เข้ามาช่วย โดยจะคัด feature point ที่ไม่สอดคล้องกับภาพต้นฉบับออก(เส้นสีแดง) ทำให้ได้การประมาณการวางตัวของกล้องถูกต้องและแม่นยำมากยิ่งขึ้นและในปัจจุบัน กระบวนการประมาณตำแหน่งและทิศทางการวางตัวของกล้องนั้นมีได้หลากหลายขั้นตอนวิธี โดยจะสามารถแบ่งออกเป็น 3 แบบหลัก ได้แก่

1) Direct Linear Transformation เป็นขั้นตอนวิธีการประมาณแบบ โดยไม่สนใจเงื่อนไขข้อกำหนดที่เกี่ยวกับผลลัพธ์ที่จะเป็นไปได้ทั้งหมด (Solution Space)
2) Perspective n-Point เป็นขั้นตอนวิธีที่จะประมาณค่าตัวแปรที่เหมาะสมก่อน เพื่อที่จะจำกัดผลลัพธ์ให้ครอบคลุมเฉพาะการวางตัวของกล้องที่มีความสมเหตุสมผลออกจากผลลัพธ์ที่จะเป็นไปได้ทั้งหมด (Solution Space)
3) A priori information estimators เป็นขั้นตอนวิธีที่จะนำข้อมูลที่เกี่ยวข้องมาช่วย ทำให้สามารถหาผลลัพธ์ได้อย่างรวดเร็วและน่าเชื่อถือมากยิ่งขึ้น โดยข้อมูลที่เกี่ยวข้องที่จะนำมาพิจารณา ได้แก่
   - Inclination information จะใช้ตัวรับรู้ความเอียง (Inclination sensor) มาใช้เป็นค่าของเมทริกซ์การหมุน (Rotation Matrix)
   - Prior probability จะใช้ข้อมูลบางอย่างเพื่อที่จะจำกัดให้เหลือแต่ผลลัพธ์ที่น่าจะเป็นไปได้

**2.3.6 ASIFT Fully Affine Invariant Comparison**

ASIFT นั้นเป็นอัลกอริทึมที่ถูกพัฒนามาจาก SIFT โดย ASIFT นั้นจะทำการจำลองมุมมองภาพด้วยมุมหมุนตามแนวแกนโลกอย่างลองจิจูด ($\Psi$) และละติจูด ($\varphi$) แล้วนำภาพที่ทำการจำลองมาได้นั้นเข้าสู่กระบวนการ SIFT โดยมีขนาดของการหมุนนั้นตามพารามิเตอร์ Tilt ของภาพตั้งต้น ซึ่งหัวใจหลักของอัลกอริทึมนี้นั้นคือการเปลี่ยนแปลงภาพตั้งต้นด้วย แอฟไฟน์ โดยมีสมการเป็น

$$A = \lambda \begin{bmatrix} cos\Psi & -sin\Psi \\ sin\Psi & cos\Psi \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} cos\varphi & -sin\varphi \\ sin\varphi & cos\varphi \end{bmatrix} \tag{2.20}$$

โดย สามารถเขียนย่อได้เป็น

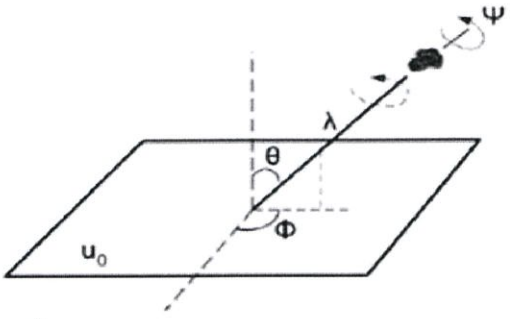
$$A = \lambda R_1(\Psi) T_t R_2(\varphi) \tag{2.21}$$

โดยที่

$\lambda$ เป็นค่าของ det(A)

$R_1$ เป็นมุมหมุน โดยมีค่า $\varphi$ ตั้งแต่ [0,180)

$T_t$ เป็นค่าของแกนเอียงโดยต้องมีค่ามากกว่า 1




**รูป 2.23 ขั้นตอนการหมุนภาพด้วย Affine Camera Model**

โดยแต่ละภาพที่รับเข้ามานั้นจะถูกเปลี่ยนแปลงด้วยการจำลองด้วย Affine camera model ที่เป็นไปได้ทั้งหมด ซึ่งการเปลี่ยนแปลงนี้จะขึ้นอยู่กับมุมสองมุมคือ ลองจิจูด และละติจูด ซึ่งละติจูดนั้นจะทำการจำลองมีโดยเพิ่มค่าไปในแต่ละ Tilt โดยมีลำดับการเพิ่มค่าเป็นอนุกรมเรขาคณิต

$$1, a, a^2, \dots, a^n \tag{2.22}$$

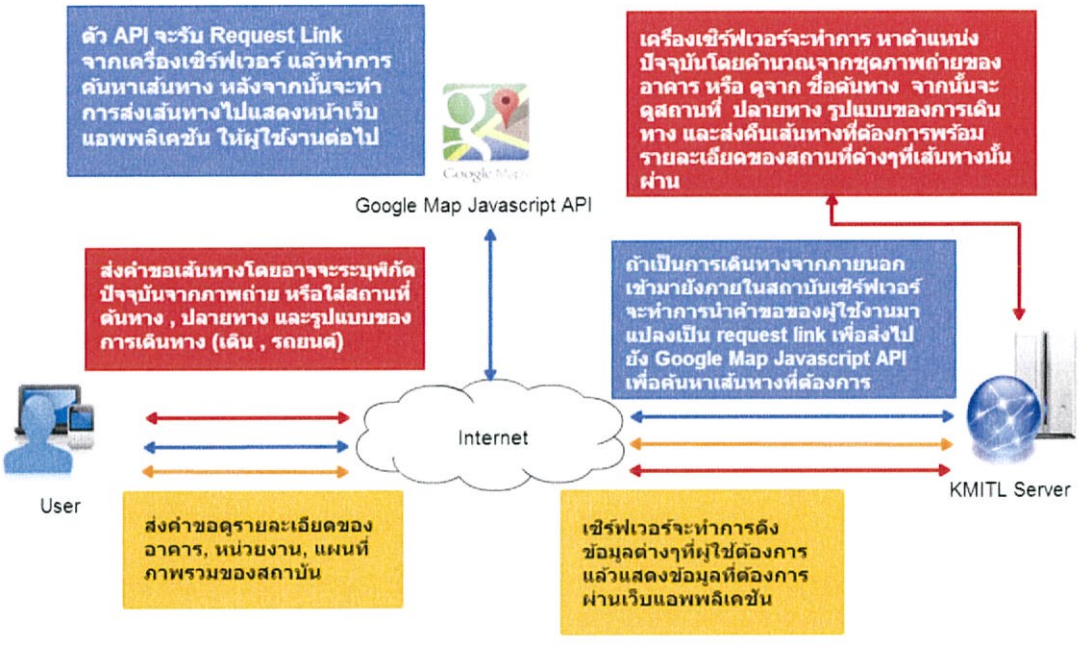
ส่วนลองจิจูดนั้นในแต่ละ Tilt จะทำการจำลองไปทีละรอบในแต่ละ Tilt โดยเพิ่มค่าเป็นลำดับอนุกรมเลขคณิต

$$0, \frac{b}{t}, \dots, \frac{kb}{t} < 180° \tag{2.23}$$

ซึ่งในโครงงานนี้จะกำหนดค่าพารามิเตอร์สองตัวนี้โดยให้ ละติจูดเพิ่มไปทีละ $\sqrt{2}$ และลองจิจูดเพิ่มไปทีละ $\frac{72°}{t}$ ตามที่ [3] กำหนดมา และหลังจากทำการจำลองในแต่ละภาพแล้ว ASIFT จะทำการนำภาพแต่ละภาพที่จำลองได้เข้าสู่กระบวนการ SIFT เพื่อประมวลผลต่อ

# บทที่ 3
# การออกแบบ และพัฒนา

## 3.1 Conceptual Design




**รูป 3.1 ภาพรวมของระบบ**

จากรูป 3.1 แสดงถึงการเชื่อมต่อกันระหว่างผู้ใช้กับเซิร์ฟเวอร์ และ Google Map Javascript API ซึ่งแสดงถึงภาพรวมของการทำงานแต่ละส่วน

ในการทำงานของเว็บแอพพลิเคชัน ผู้ใช้งานต้องเข้าผ่านเว็บเบราว์เซอร์ โดยการทำงานของระบบจะแบ่งส่วนของการทำงานเป็น 2 ส่วน (ดังรูป 3.1) ได้แก่ส่วนเว็บแอพพลิเคชัน และเซิร์ฟเวอร์

### 3.1.1 การทำงานในส่วนเว็บแอพพลิเคชัน

ในการทำงานของส่วนเว็บแอพพลิเคชัน มีหน้าที่เป็นส่วนติดต่อระหว่างผู้ใช้งานกับตัวเซิร์ฟเวอร์ ผู้ใช้งานสามารถส่งข้อมูลรูปภาพถ่ายในตำแหน่งปัจจุบัน (ในกรณีที่สถานที่ต้นทางอยู่ภายในสถาบัน) หรือระบุพิกัดสถานที่ต้นทางเข้าสู่ระบบ และทำการจำลองเส้นทางจากสถานที่ต้นทางไปยังสถานที่ปลายทางได้ด้วยการระบุชื่อของสถานที่ปลายทาง หรือระบุเป็นพิกัดของสถานที่ปลายทาง โดยผู้ใช้งานสามารถเลือกรูปแบบการจำลองเส้นทางได้สองรูปแบบด้วยกัน คือ รถยนต์ หรือ เดิน หลังจากนั้นเครื่องเซิร์ฟเวอร์จะทำการประมวลผลเพื่อหาตำแหน่งที่อยู่ ณ ปัจจุบันของ

ผู้ใช้ โดยสามารถระบุตำแหน่งพิกัดที่แท้จริงจากภาพถ่ายที่ส่งเข้ามา หรือจากพิกัดต้นทางที่ถูกระบุจากอุปกรณ์ของผู้ใช้ และนำผลลัพธ์ที่ได้จากการจำลองเส้นทาง ได้แก่ ตำแหน่งปัจจุบัน เส้นทาง วิดีโอ รูปภาพของสถานที่ที่ผ่านในเส้นทางนั้น รวมไปถึงรายละเอียดของสถานที่ เช่น ชื่ออาคาร หรือ ชื่อหน่วยงานที่สังกัดภายในอาคารนั้น เป็นต้น ตลอดเส้นทาง มาแสดงให้ผู้ใช้งานดูเป็นข้อมูลในการเดินทางในเส้นทางที่สนใจได้อย่างมีประสิทธิภาพ

นอกจากนั้นยังมีการจำลองการให้ข้อมูลรายละเอียดสถานที่ ด้วยการให้รายละเอียดข้อมูลแนะนำสถาบันให้กับผู้ใช้งานโดยทางผู้ใช้งานสามารถเข้ามาดูรายละเอียดของอาคารแต่ละอาคาร รวมไปถึงหน่วยงานแต่ละหน่วยงานที่สังกัดอยู่ภายในอาคารนั้น และผู้ใช้งานสามารถดูแผนที่ภาพรวมของสถาบัน โดยจะประกอบไปด้วยสถานที่สำคัญ ร้านอาหาร และสถานที่ให้บริการภายในสถาบัน เป็นต้น

**3.1.2 การทำงานในส่วนเซิร์ฟเวอร์**

ในการทำงานของส่วนเครื่องเซิร์ฟเวอร์มีหน้าที่ในการติดต่อกับผู้ใช้ รวมไปถึงการประมวลผล และเก็บข้อมูลรายละเอียดไม่ว่าจะเป็น เส้นทาง ภาพถ่ายของสถานที่แต่ละสถานที่ รายละเอียดสถานที่ วิดีโอ และมีหน้าที่ในการติดต่อกับกูเกิลแมพเอพีไอเพื่อนำรายละเอียดมาใช้ในการคำนวณเพื่อแสดงผลลัพธ์ให้ผู้ใช้งาน

ในการประมวลผลนั้นสำหรับกรณีที่ผู้ใช้ระบุสถานที่ต้นทางด้วยภายถ่ายของอาคาร เซิร์ฟเวอร์จะนำเอาภาพถ่ายจากผู้ใช้มา มาทำการเปรียบเทียบกับรูปภาพสถานที่ในฐานข้อมูลเพื่อนำมาหาตำแหน่งพิกัดปัจจุบันสำหรับใช้ในการยืนยันพิกัดตำแหน่งที่แท้จริง โดยในการเปรียบเทียบภาพถ่ายกับรูปภาพในฐานข้อมูลจะทำการเปรียบเทียบจากพิกัดที่ใกล้เคียงกับตำแหน่งของผู้ใช้งานปัจจุบันเพื่อลดระยะเวลาในการยืนยันตำแหน่งของผู้ใช้

นอกจากนี้เครื่องเซิร์ฟเวอร์จะเป็นตัวกลางในการเชื่อมต่อไปยัง Google Map Javascript API ถ้ามีคำขอของเส้นทางโดยรถยนต์จากสถานที่ภายในสถาบันไปยังสถานที่ภายในสถาบัน และสถานที่ภายนอกสถาบันมายังสถานที่ภายในสถาบัน โดยเครื่องเซิร์ฟเวอร์จะรับอินพุตของสถานที่ต้นทาง และสถานที่ปลายทางของผู้ใช้งานมา แล้วทำการแปลงคำขอนั้นไปเป็น Request URL เพื่อส่งไปยังเอพีไอของกูเกิลเพื่อค้นหาเส้นทางแล้วนำมาแสดงผลให้แก่ผู้ใช้ ในขณะที่ถ้ามีคำขอเส้นทางโดยการเดินเข้ามา เซิร์ฟเวอร์จะทำการค้นหาเส้นทางที่มีอยู่ในฐานข้อมูล แล้วนำมาแสดงผลให้แก่ผู้ใช้แทนการขอเส้นทางจาก Google Map Javascript API

ในส่วนของวิดีโอแนะนำเส้นทางเครื่องเซิร์ฟเวอร์จะนำภาพถ่ายจาก Google Street View API มาทำการแปลงเป็นวิดีโอ ซึ่งในส่วนของวิดีโอนั้นจะนำภาพถ่ายมาแสดงเป็นวิดีโอเฉพาะการขอเส้นทางจากภายในสถาบันไปยังภายในสถาบันเท่านั้น

## 3.2 ความต้องการของระบบ (System Requirement)

### 3.2.1 อินพุต (Input)

1) ภาพถ่ายปัจจุบัน หรือ ชื่อสถานที่ (สถานที่ต้นทาง)
2) สถานที่ปลายทาง
3) ตำแหน่งพิกัดปัจจุบัน (ละติจูด ลองจิจูด)
4) รูปแบบการเดินทางที่ต้องการ (รถยนต์ ทางเท้า)
5) สถานที่ภายในสถาบัน (สำหรับแสดงรายละเอียดสถานที่)

### 3.2.2 เอาท์พุต (Output)

1) เส้นทางระหว่างต้นทางถึงปลายทาง
2) วิดีโอจำลองเส้นทาง
3) สถานที่ที่ใกล้เคียงที่ผ่านตลอดทั้งเส้นทาง (อาคาร ร้านค้าที่ให้บริการ)
4) ระยะทางระหว่างต้นทางถึงปลายทาง
6) ข้อมูลของการเดินทางโดยใช้ขนส่งสาธารณะ

### 3.2.3 หน้าที่หลักของระบบ (Functional Requirement)

#### 3.2.3.1 ฟังก์ชันแผนที่

1) สามารถค้นหา และแสดงเส้นทางระหว่างอาคารในสถาบันไปยังอีกอาคารหนึ่งในสถาบันได้
2) สามารถค้นหา และแสดงเส้นทางจากภายนอก มายังภายในสถาบันได้
3) สามารถระบุตำแหน่งปัจจุบันจากภาพถ่ายอาคารของผู้ใช้ที่ส่งเข้ามาได้ หรือ ระบุจากชื่อสถานที่ต้นทาง หรือ พิกัดของสถานที่ต้นทางได้
4) สามารถค้นหาสถานที่ภายในสถาบันได้

2) ในแต่ละเส้นทางมีการแสดงรายละเอียด และรูปภาพของสถานที่ที่เส้นทางนั้นผ่านได้ (เฉพาะภายในสถาบัน)
3) แสดงวิดีโอเพื่อใช้เป็นตัวช่วยในการแนะนำเส้นทางได้ (เฉพาะภายในสถาบัน)
4) แสดงแผนที่ภาพรวมภายในสถาบันได้

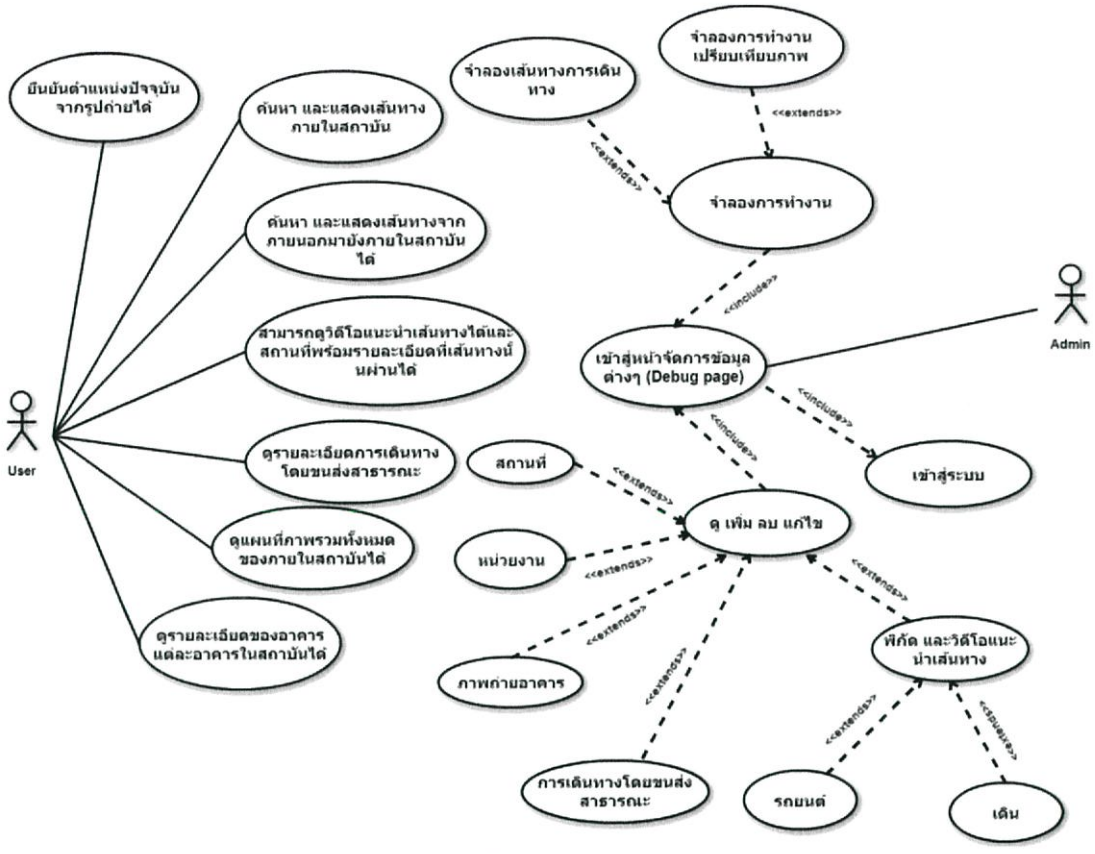
#### 3.2.3.2 ฟังก์ชันการเดินทางด้วยขนส่งสาธารณะ

แสดงวิธีการเดินทางมายังสถาบันจากขนส่งสาธารณะประเภท เช่น รถไฟ รถเมล์ รถตู้ แอร์พอร์ตลิงค์ ฯลฯ เป็นต้น

#### 3.2.3.3 ฟังก์ชันแสดงข้อมูลภายในสถาบัน

แสดงข้อมูลของอาคารภายในสถาบัน เช่น หน่วยงาน เว็บของหน่วยงาน เป็นต้น

### 3.2.4 Use Case Diagram




**รูป 3.2 แผนภาพยูสเคสไดอะแกรม (Use Case Diagram)**

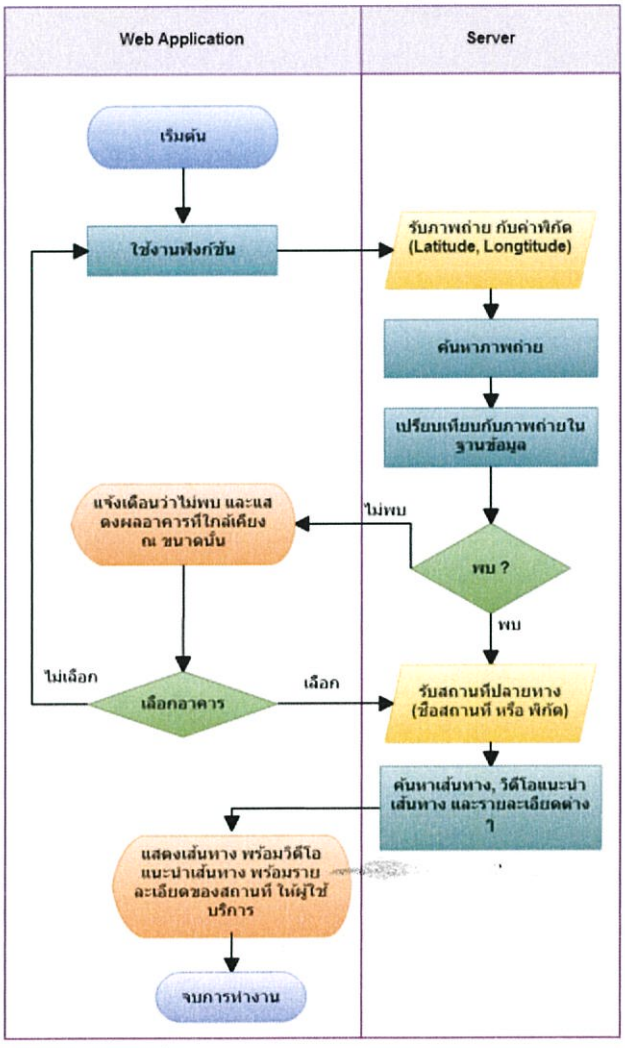
จากรูป 3.2 เป็นแผนภาพแสดงถึงการทำงานของเว็บแอพพลิเคชั่น โดยกำหนดสิทธิ์ไว้ว่าผู้ใช้งานประเภทไหน สามารถเข้าถึงการใช้งานในฟังก์ชันใดได้บ้าง

## 3.3 การทำงานในแต่ละฟังก์ชัน

### 3.3.1 การทำงานของฟังก์ชันแผนที่

ในส่วนของฟังก์ชันแผนที่ใช้สำหรับการนำทาง โดยสามารถแบ่งได้เป็น 2 ข้อหลัก

**3.3.1.1 การนำทางภายในสถาบัน (สถานที่ในสถาบัน - สถานที่ในสถาบัน) โดยรถยนต์และการเดิน**



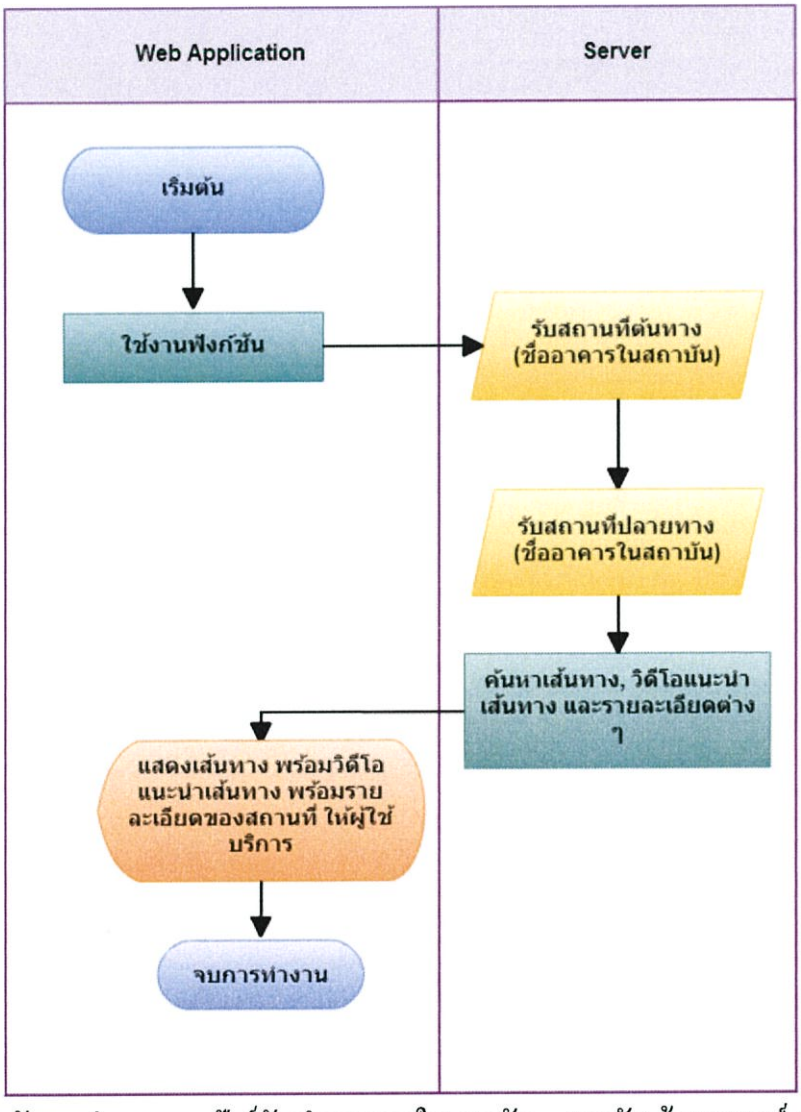

**รูป 3.3 แผนผังการทำงานของฟังก์ชันนำทางภายในสถาบัน – สถาบัน ด้วยรถยนต์ และการเดิน**

จากรูป 3.3 ในการนำทางจากสถานที่ภายในสถาบันไปสถานที่ภายในสถาบันนั้น เมื่อมีการเรียกใช้งาน ตัวเซิร์ฟเวอร์จะรับภาพถ่ายอาคารจากผู้ใช้ มาเพื่อทำการเปรียบเทียบกับภาพถ่ายอาคารในฐานข้อมูล โดยจะอ้างอิงจากค่าพิกัดของสถานที่ที่ผู้ใช้ทำการส่งภาพถ่ายอาคารมา ถ้าพบจะทำการยืนยันพิกัดของสถานที่ต้นทางที่ถูกต้องให้แก่ผู้ใช้งาน และให้ผู้ใช้งานเลือกสถานที่

ปลายทางที่ต้องการไป หลังจากนั้นตัวเซิร์ฟเวอร์จะค้นหาเส้นทาง วิดีโอ และรายละเอียดของสถานที่ใกล้เคียง แสดงผลผ่าน Web Application ให้แก่ผู้ใช้งานต่อไป

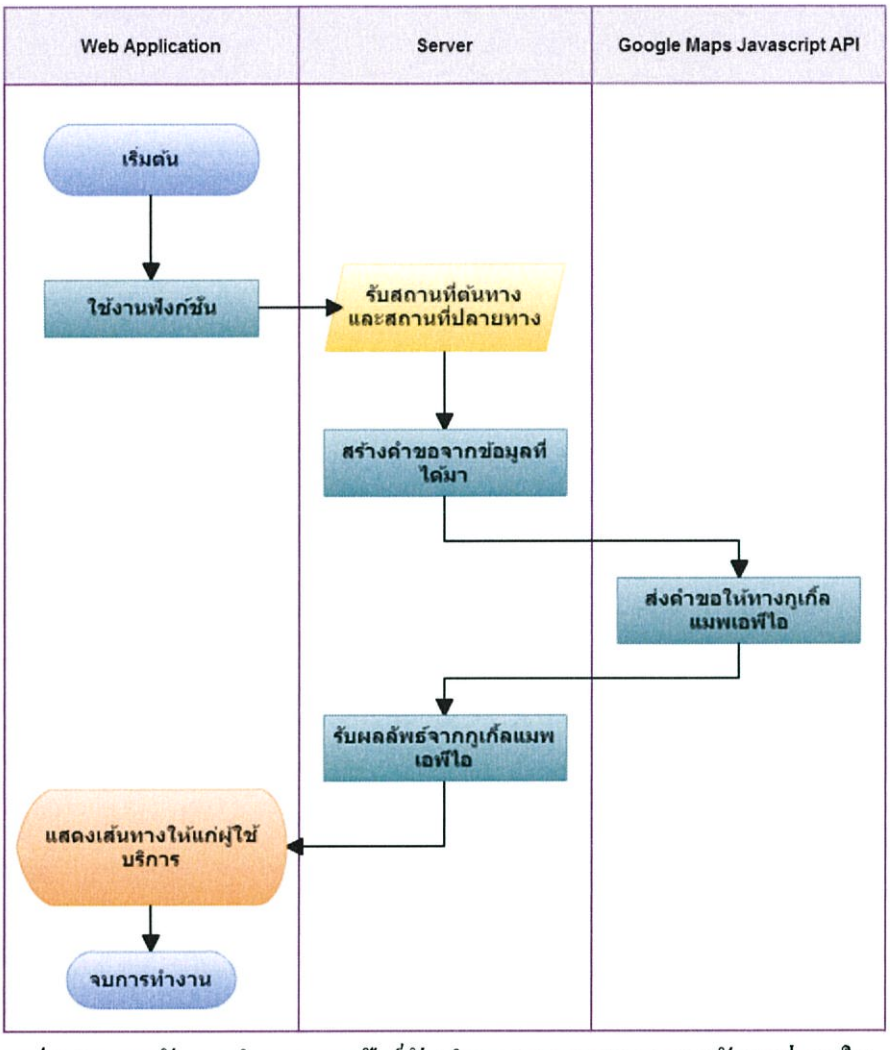
แต่ถ้าหากเปรียบเทียบแล้วไม่พบว่ามีภาพใดที่เหมือนกันเลยเซิร์ฟเวอร์จะทำการแจ้งเตือนผู้ใช้งานว่าไม่พบข้อมูล แล้วทำการคืนค่าภาพอาคารบริเวณที่ใกล้เคียงกับผู้ใช้งานเพื่อเป็นตัวเลือกว่า ณ เวลานั้นผู้ใช้งานอยู่ใกล้อาคารที่ทำการแสดงภาพให้หรือไม่ ถ้าผู้ใช้งานเลือกก็จะระบุพิกัด ให้แก่ผู้ใช้งานต่อไป แต่ถ้าหากไม่ใช่จะกลับไปสู่หน้าเริ่มต้นเพื่อกรอกข้อมูลเริ่มใหม่ต่อไป

นอกจากจะสามารถใช้ภาพยืนยันพิกัดตำแหน่งปัจจุบันได้แล้ว หากผู้ใช้งานรู้ว่าตอนนี้ตนเองอยู่ ณ ที่อาคารใด พิกัดใด ก็สามารถยืนยันพิกัดปัจจุบันได้จากชื่อสถานที่ หรือพิกัดได้เช่นกัน โดยแผนผังการทำงานจะเป็นไปตามรูป 3.4




**รูป 3.4 แผนผังการทำงานของฟังก์ชันนำทางภายในสถาบัน – สถาบัน ด้วยรถยนต์ และการเดิน โดยยืนยันตำแหน่งพิกัดปัจจุบันด้วยชื่อ**

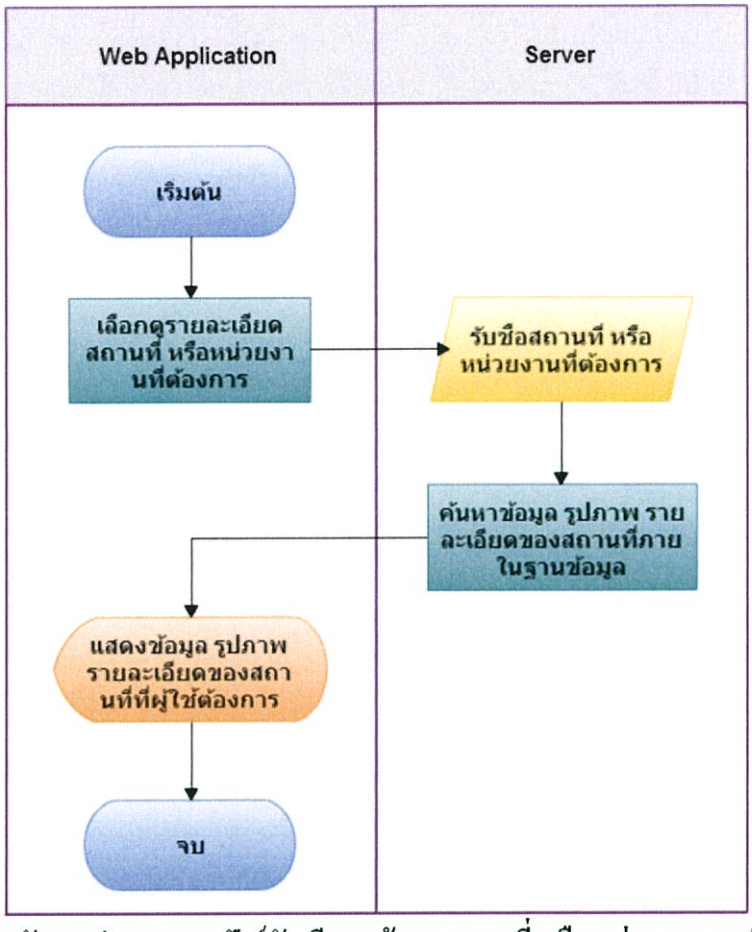
### 3.3.2 การนำทางจากภายนอกสถาบันมายังภายในสถาบัน




**รูป 3.5 แผนผังการทำงานของฟังก์ชันนำทางจากภายนอกสถาบันมาสู่ภายใน**

สำหรับรูป 3.5 เป็นแผนผังการทำงานของฟังก์ชันนำทางจากภายนอกสถาบันมาสู่ภายใน โดยเมื่อมีการเรียกใช้งานฟังก์ชัน เซิร์ฟเวอร์จะรับค่าสถานที่ต้นทาง และสถานที่ปลายทางมาสร้างเป็นลิงค์คำขอ (Request Url) เพื่อติดต่อไปยัง Google Maps API แล้วรับเส้นทางที่ได้จากเอพีไอมาแสดงผลให้แก่ผู้ใช้งานต่อไป

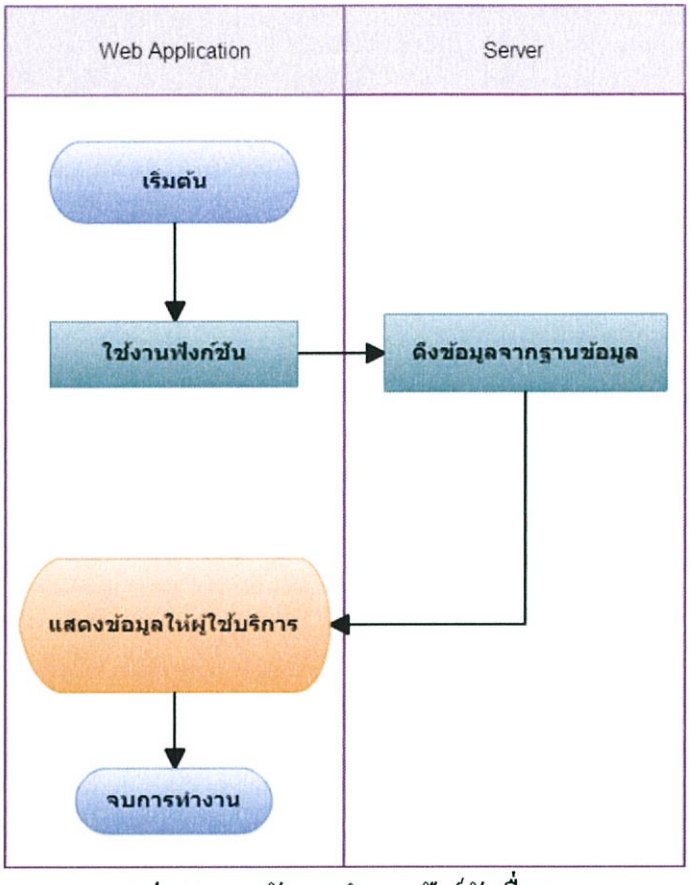
### 3.3.3 การทำงานของฟังก์ชันเรียกดูข้อมูลของสถานที่ หรือหน่วยงานภายในสถาบัน




**รูป 3.6 แผนผังการทำงานของฟังก์ชันเรียกดูข้อมูลสถานที่ หรือหน่วยงานภายในสถาบัน**

รูป 3.6 แสดงถึงขั้นตอนการทำงานของฟังก์ชันการเรียกดูข้อมูลสถานที่ หรือหน่วยงานภายในสถาบัน โดยผู้ใช้งานเลือกสถานที่ หรือหน่วยงานจากชื่อ หลังจากนั้นตัวเซิร์ฟเวอร์จะทำการค้นหาข้อมูล รูปภาพ รายละเอียดของสถานที่ หรือหน่วยงานที่ต้องการนำมาแสดงผลผ่านทาง เว็บแอพพลิเคชั่น

### 3.3.4 การทำงานของฟังก์ชันอื่นๆ

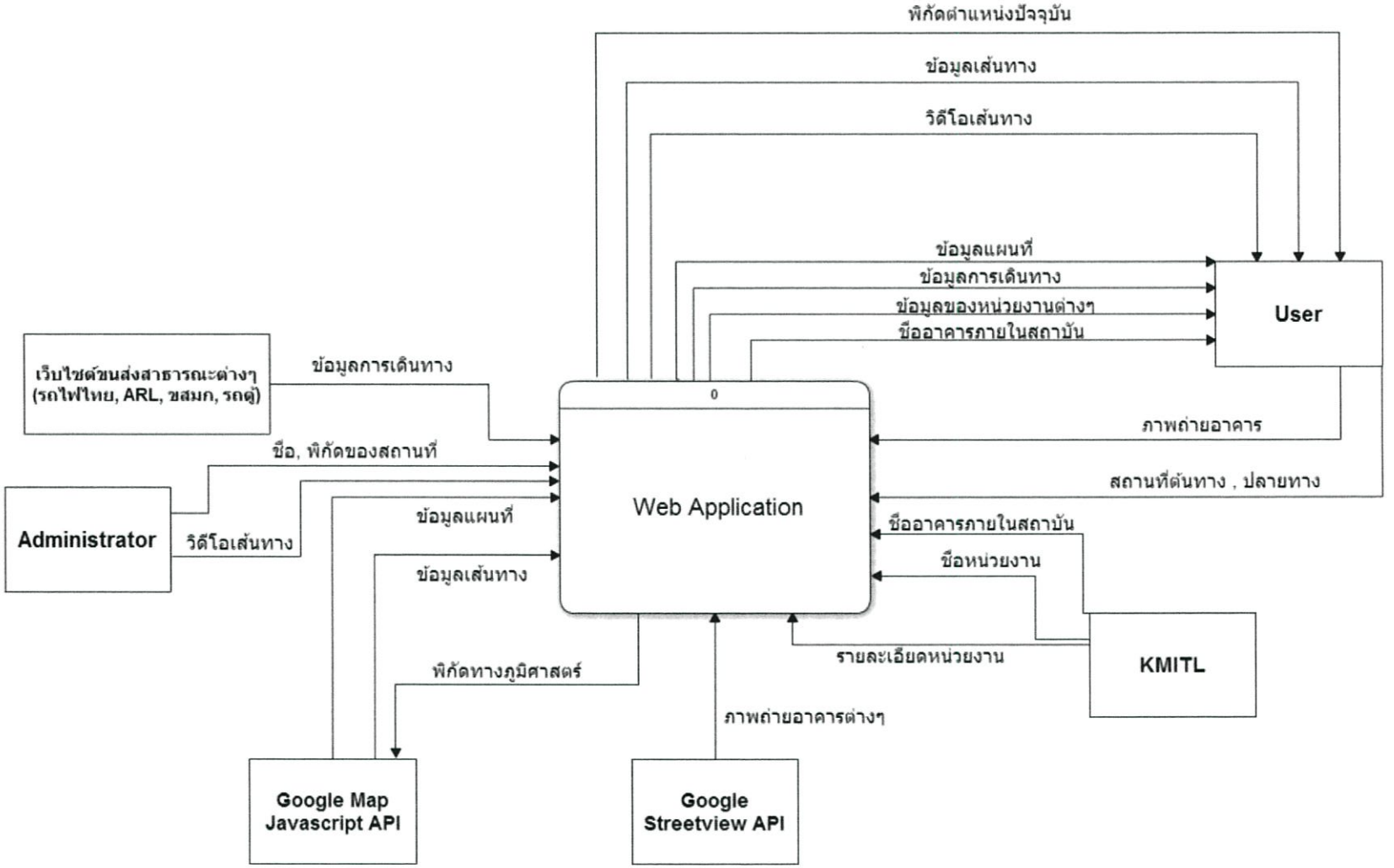



**รูป 3.7 แผนผังการทำงานฟังก์ชันอื่น ๆ**

ในส่วนการทำงานของฟังก์ชันอื่น ๆนี้ จะอธิบายถึงขั้นตอนของการทำงานในรูป 3.7 เมื่อเรียกใช้งานฟังก์ชันดูรายละเอียด อาทิ เช่น เรียกดูแผนที่ภาพรวมภายในสถาบัน เรียกดูวิธีการเดินทางด้วยขนส่งสาธารณะ เป็นต้น

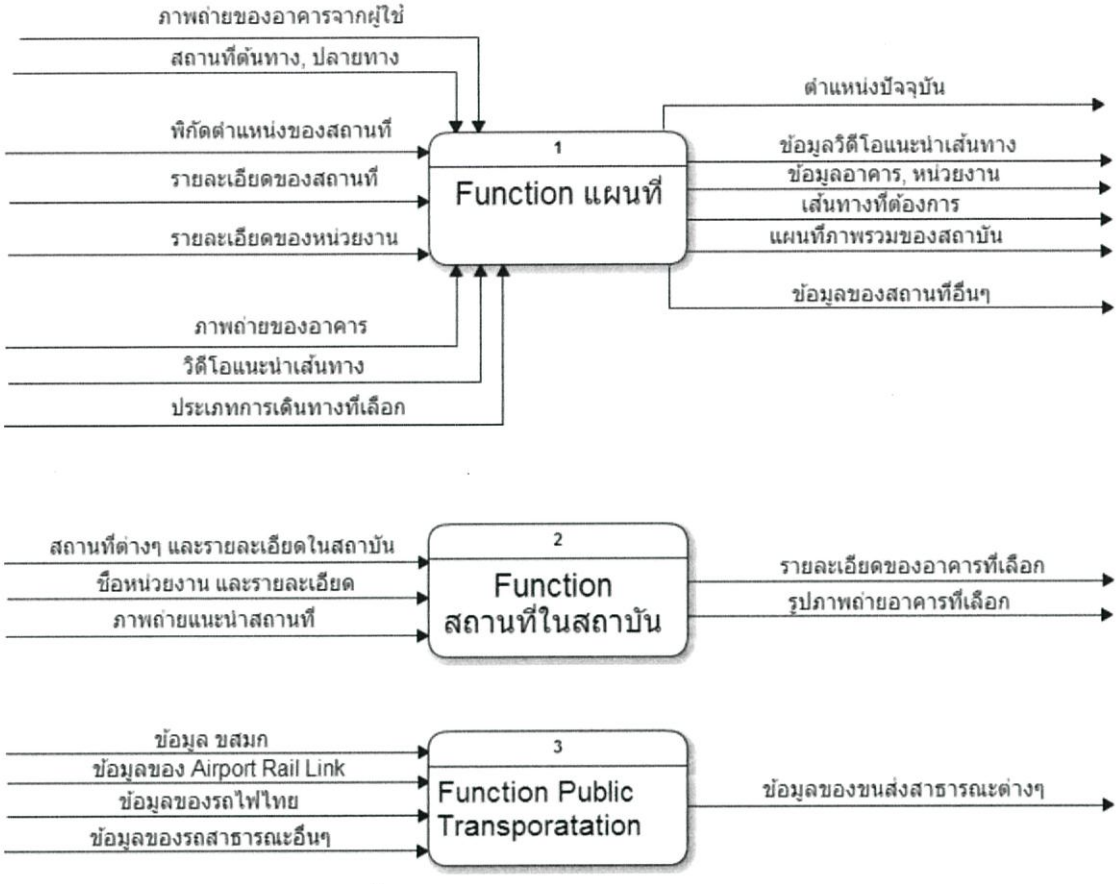
## 3.4 แผนภาพการไหลของข้อมูลของระบบ (Data Flow Diagram: DFD)

### 3.4.1 Context Diagram



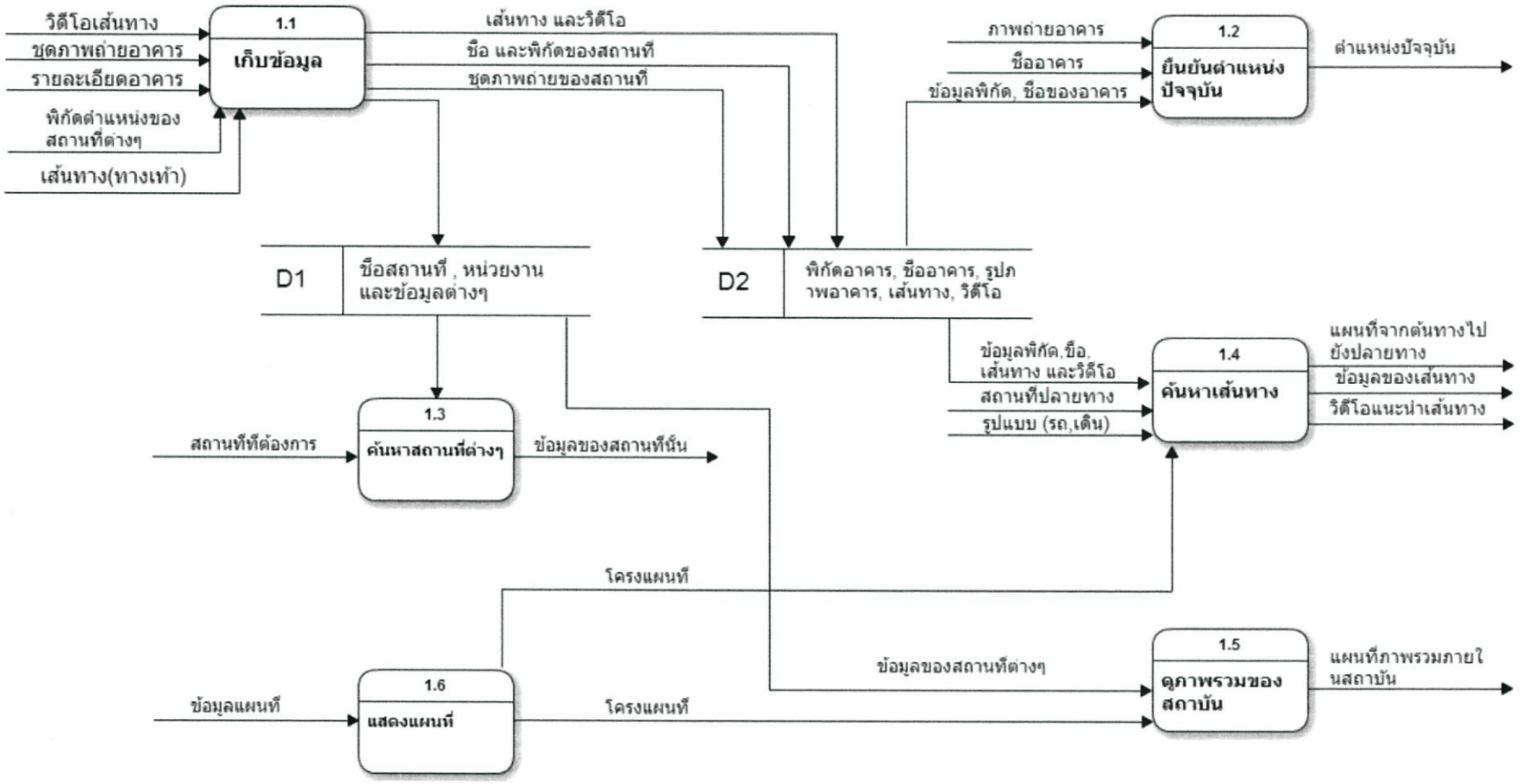

**รูป 3.8 Context Diagram**

### 3.4.2 Data Flow Diagram Level 0



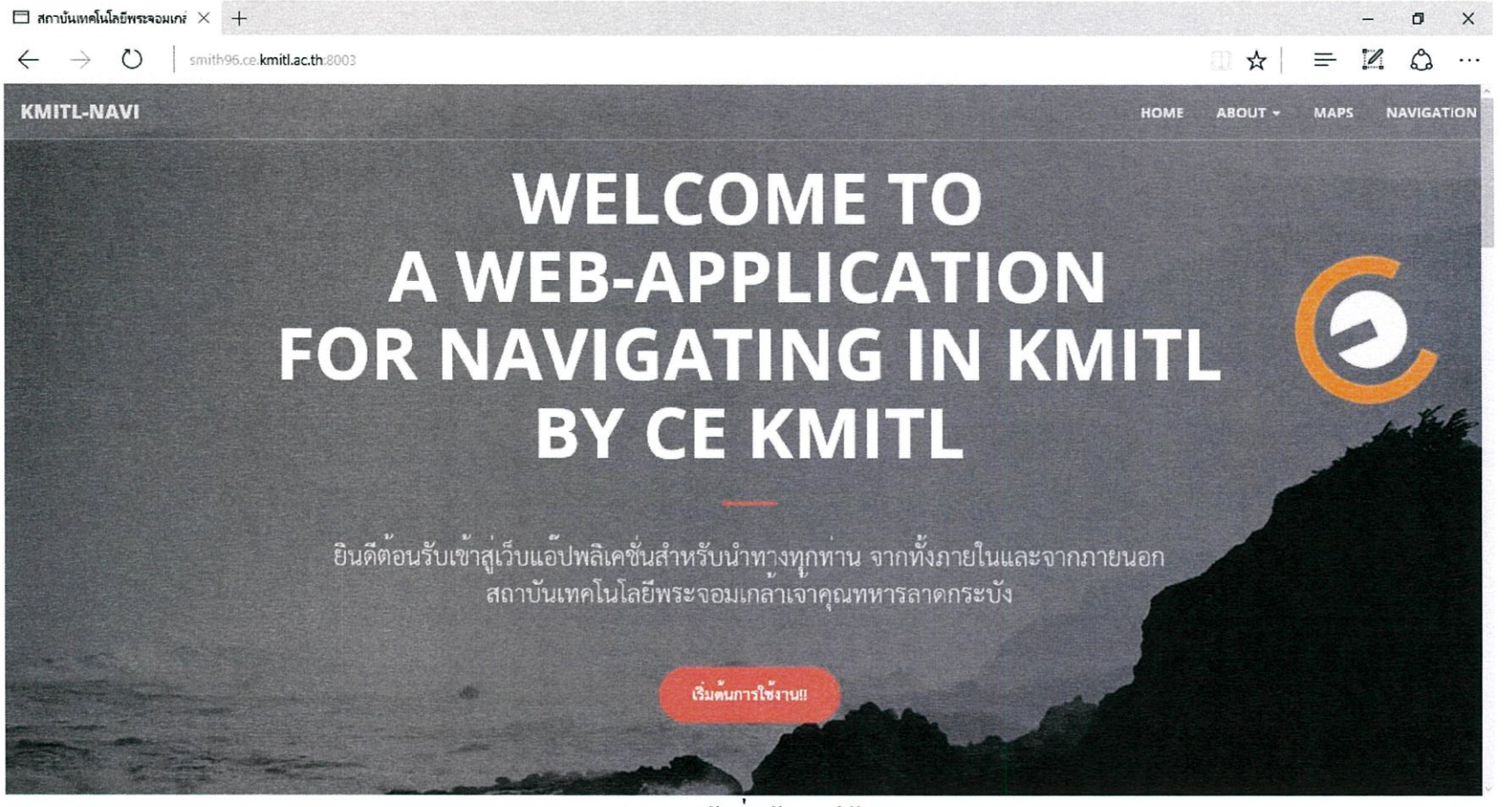

**รูป 3.9 Data Flow Diagram Level 0**

### 3.4.3 Data Flow Diagram Level 1

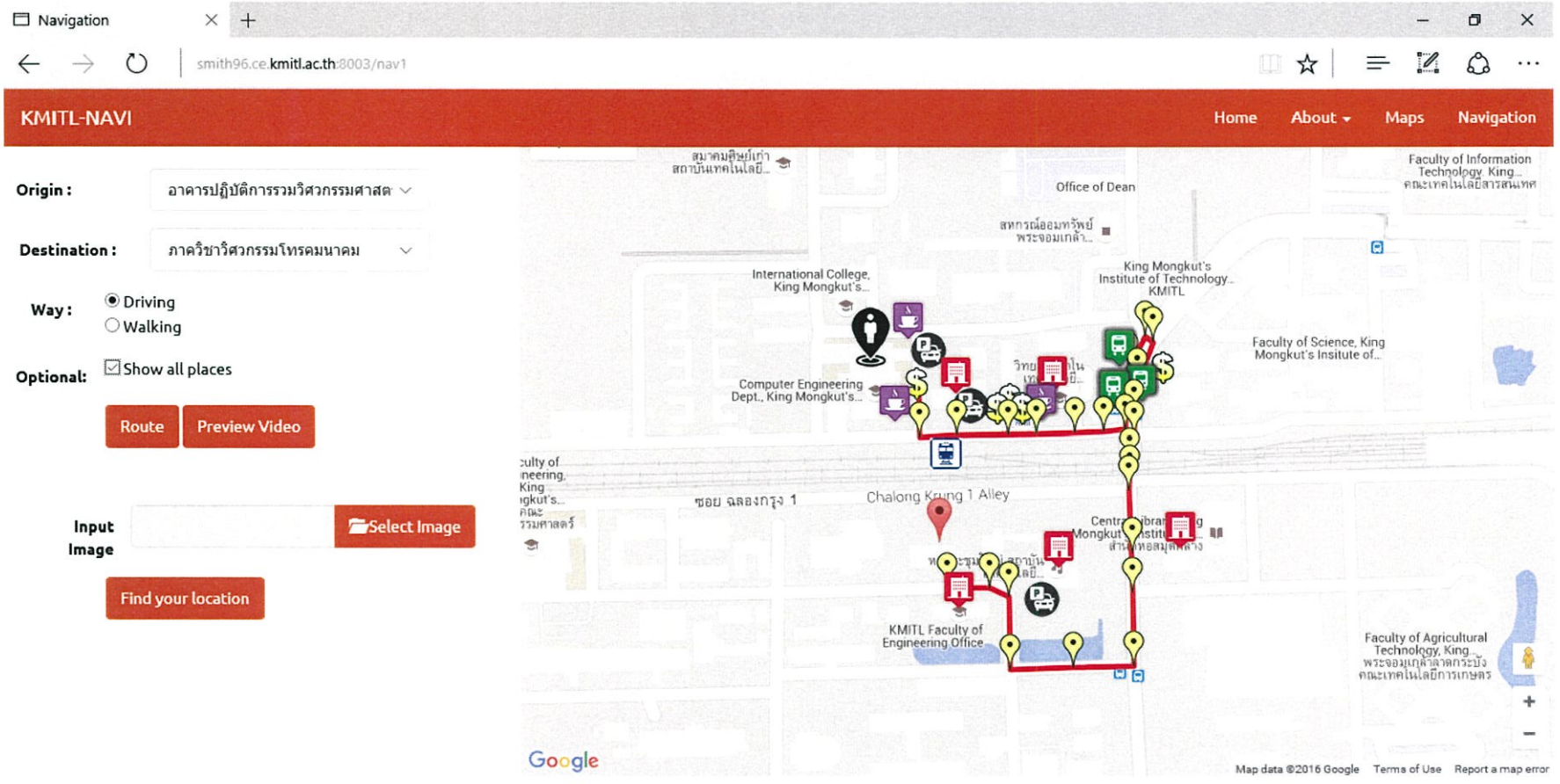



**รูป 3.10 Data Flow Diagram Level 1**

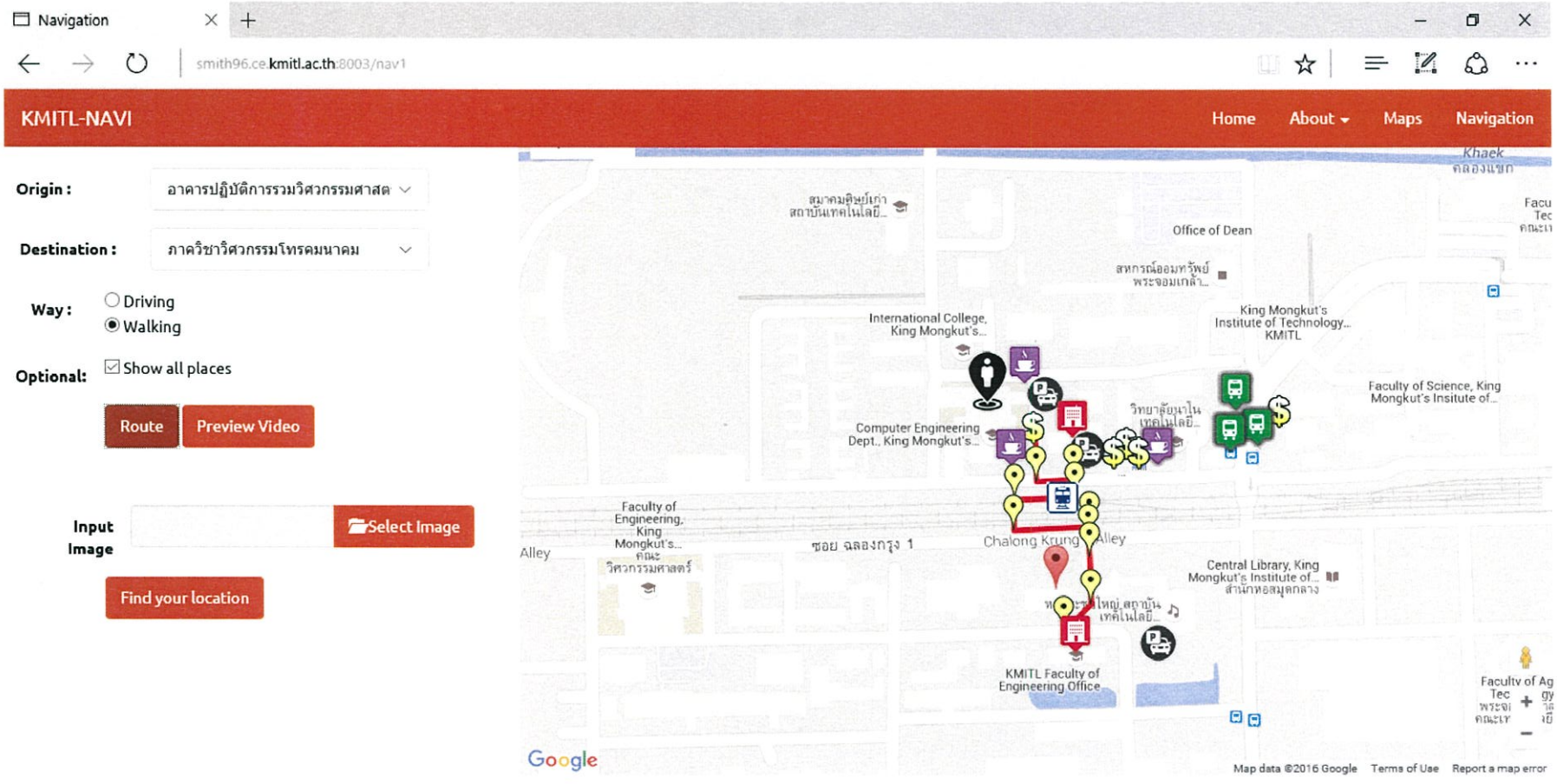
**3.5** การออกแบบส่วนติดต่อผู้ใช้งาน



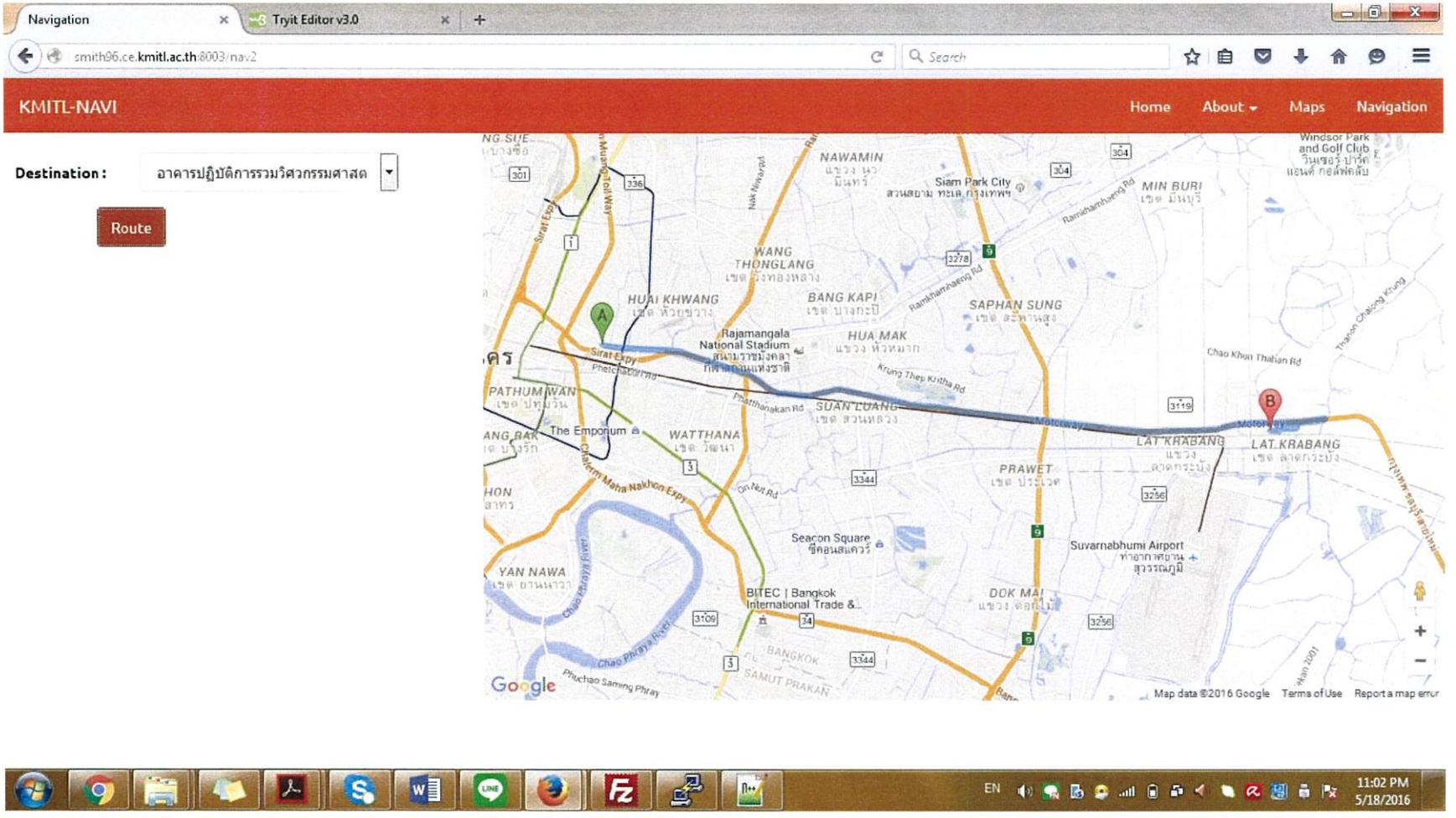

**รูป 3.11 หน้าเริ่มต้นการใช้งาน**




**รูป 3.12 การนำทางจาก ภายในสถาบัน-ภายในสถาบัน ด้วยการขับรถ และแสดงสถานที่ระหว่างทางที่ผ่าน**




รูป 3.13 การนำทางจาก ภายในสถาบัน-ภายในสถาบัน ด้วยการเดิน

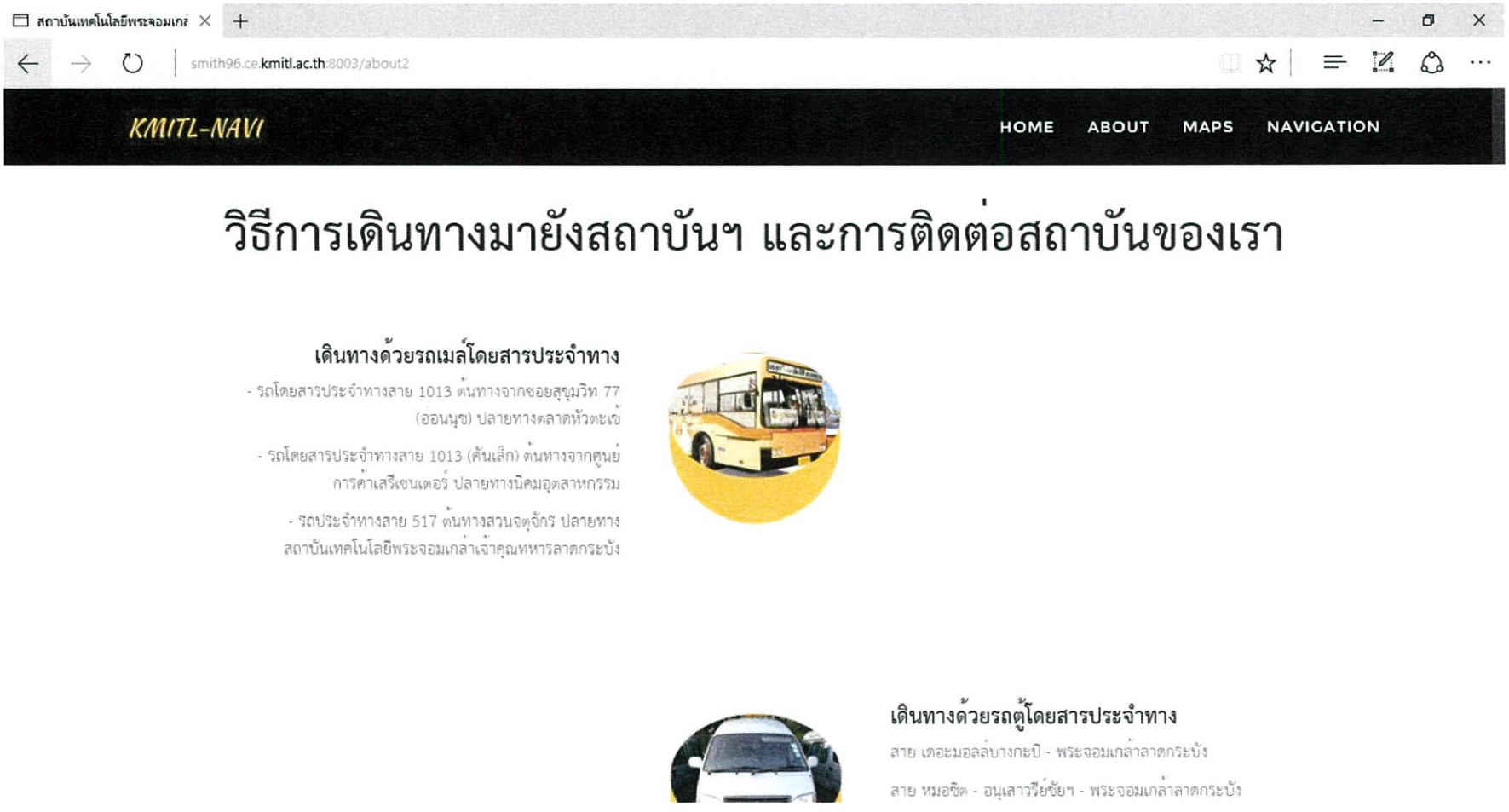



**รูป 3.14 การนำทางจาก ภายนอกสถาบัน-ภายในสถาบัน ด้วยการขับรถ**

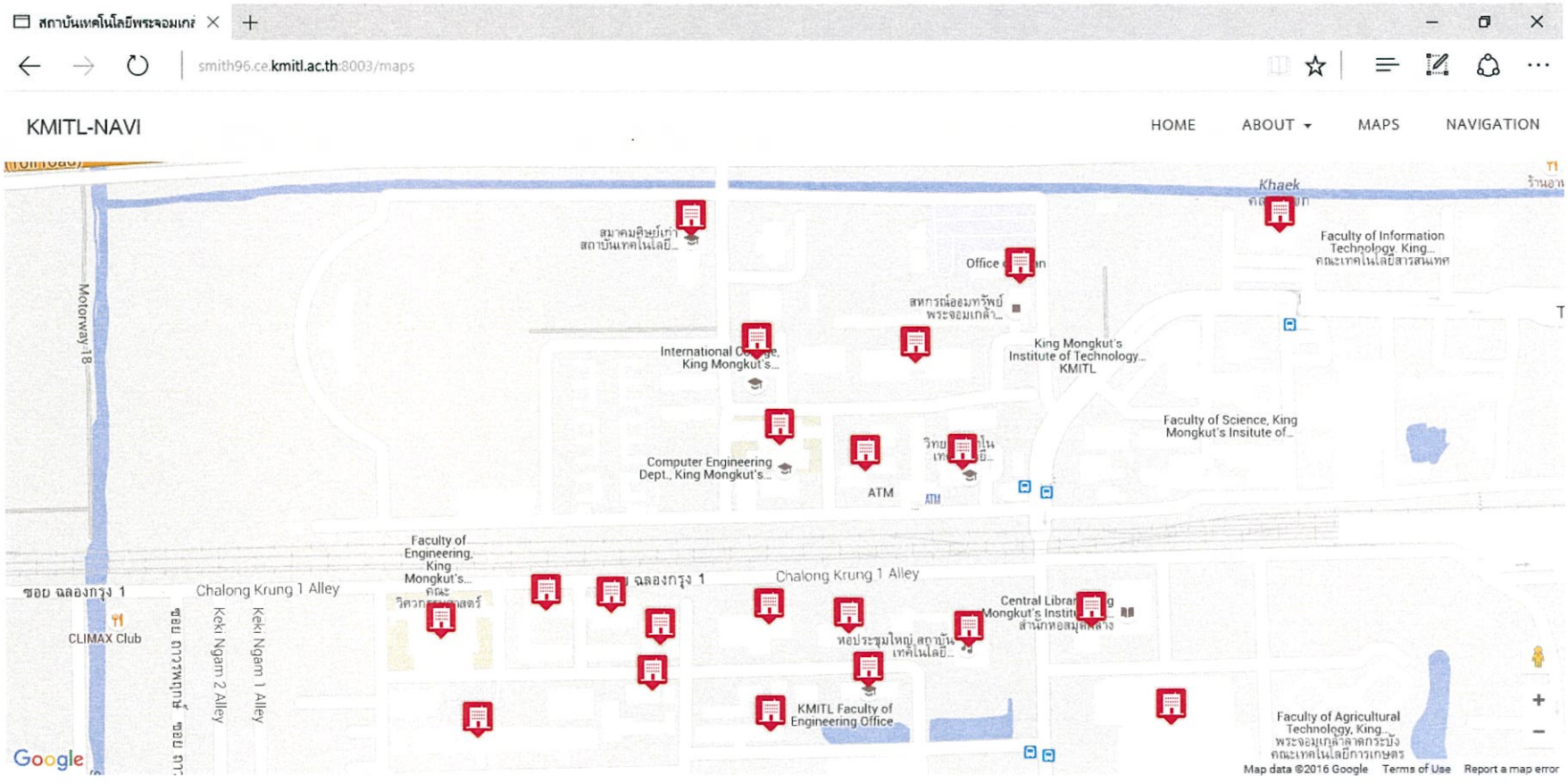
**รูป 3.15 หน้าประวัติสถาบัน**

**รูป 3.16 หน้าองค์กรและหน่วยงาน**

**รูป 3.17 หน้ารายละเอียดของตึกเมื่อกดดูในส่วนของรายละเอียด หรือ ในเมนูรายละเอียดตึก**



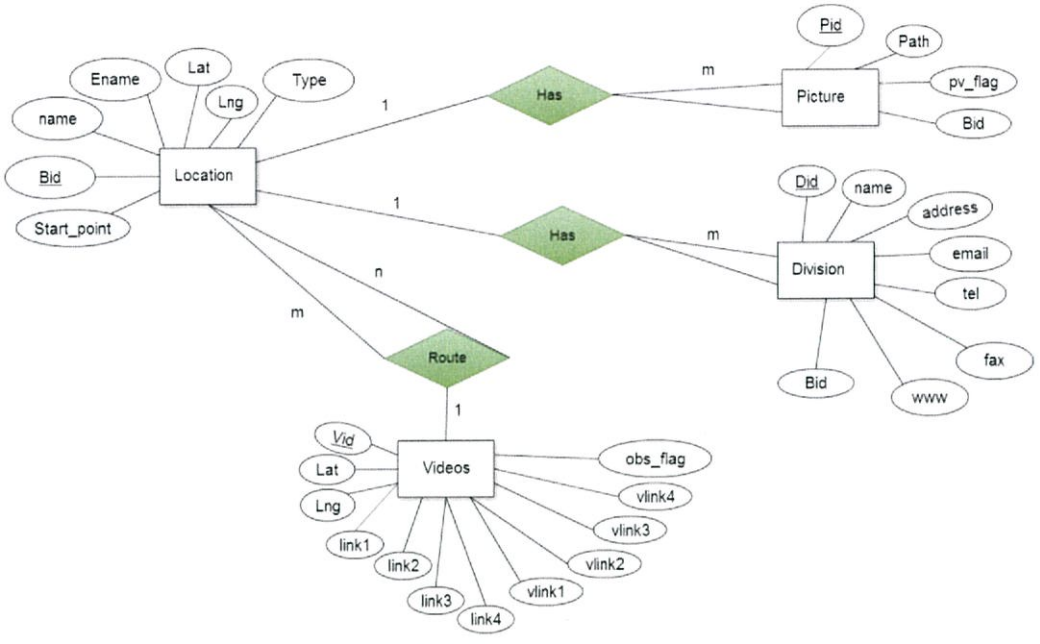

**รูป 3.18 วิธีการเดินทางมายังสถาบัน**




**รูป 3.19 แผนผังทั้งหมดของสถาบัน**

## 3.6 การออกแบบฐานข้อมูล

ในหัวข้อนี้จะกล่าวถึงการออกแบบฐานข้อมูลเพื่อเก็บชุดภาพถ่าย เส้นทางภายในสถาบัน และวิดีโอตามพิกัด ซึ่งสามารถทำการวาดโครงออกมาเป็น ER-Diagram ได้ตามรูป 3.20 ด้านล่างต่อไปนี้



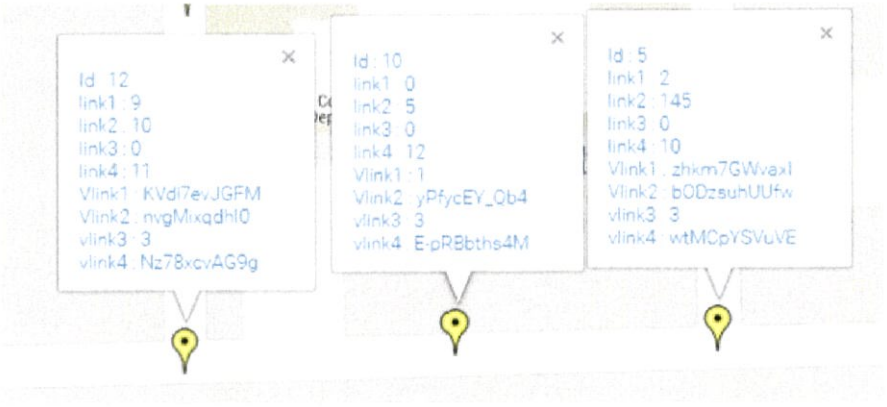

**รูป 3.20 ER-Diagram**

จากแผนภาพ Er-Diagram ดังกล่าวแสดงถึงความสัมพันธ์ของข้อมูลทั้งหมด อธิบายได้ดังต่อไปนี้ Entity Location กล่าวถึงสถานที่ที่อยู่ในสถาบันโดยสามารถเป็นได้ทั้งสถานที่ต้นทาง (Origin) หรือ สถานที่ปลายทาง (Destination) ซึ่งใช้ในการค้นหาเส้นทาง สามารถแนะนำเส้นทางได้เฉพาะสถานที่ที่เป็นอาคารเพียงเท่านั้น (Type = 1) โดยหนึ่งสถานที่ต้นทาง ต่อหนึ่งสถานที่ปลายทาง จะได้เส้นทางออกมาหนึ่งเส้นทาง และประกอบไปด้วยวิดีโอแนะนำเส้นทางจาก Entity Videos และในสถานที่แต่ละสถานที่จะประกอบไปด้วยภาพ (Picture) และข้อมูลของหน่วยงาน (Division) ที่อยู่ของหน่วยงานภายในแต่ละสถานที่นั้น รวมไปถึงรายละเอียดต่างของหน่วยงานนั้น ซึ่งเราสามารถสร้างโครงตารางจาก ER-Diagram ข้างต้นได้ทั้งหมด 4 ตารางด้วยกัน

1) Location Table เป็นโครงตารางที่ไว้ใช้เก็บข้อมูลสำหรับอาคาร หรือสถานที่ภายในสถาบันโดยประกอบไปด้วยข้อมูลดังนี้
   - Bid คือ รหัสของสถานที่ภายในสถาบัน (Primary Key)
   - Lat คือ ค่าละจิจูดของสถานที่นั้น
   - Long คือ ค่าลองจิจูดของสถานที่นั้น

- Name คือ ชื่อของสถานที่ภายในสถาบัน
- Ename คือ ชื่อภาษาอังกฤษของสถานที่นั้น
- Type คือ ประเภทของสถานที่นั้นโดยประกอบได้ด้วย
  - อาคารหรือตึก (1)
  - โรงอาหาร (2)
  - สนามกีฬา (3)
  - ร้านกาแฟ (4)
  - ธนาคาร และตู้เอทีเอ็ม (5)
  - ร้านค้า (6)
  - ที่จอดรถ (7)
- Start_point คือ จุดที่ใช้เชื่อมต่อสำหรับการนำทาง

2) Videos Table เป็นโครงตารางที่ใช้เก็บจุดในการแนะนำเส้นทางให้แก่ผู้ใช้งาน รวมไปถึงวิดีโอแนะนำเส้นทางเช่นกัน ประกอบไปด้วย
   - Vid คือ รหัสของพิกัดแนะนำเส้นทาง ณ จุดนั้น (Primary Key)
   - Lat คือ ค่าละจิจูดของพิกัดนั้น
   - Long คือ ค่าลองจิจูดของพิกัดนั้น
   - Link1 คือ จุดที่เชื่อมต่อจากจุดปัจจุบันในทิศเหนือ
   - Link2 คือ จุดที่เชื่อมต่อจากจุดปัจจุบันในทิศตะวันออก
   - Link3 คือ จุดที่เชื่อมต่อจากจุดปัจจุบันในทิศใต้
   - Link4 คือ จุดที่เชื่อมต่อจากจุดปัจจุบันในทิศตะวันตก
   - Vlink1 คือ วิดีโอแนะนำเส้นทางจากจุดปัจจุบันไปยังจุดต่อไปในทิศเหนือ
   - Vlink2 คือ วิดีโอแนะนำเส้นทางจากจุดปัจจุบันไปยังจุดต่อไปในทิศตะวันออก
   - Vlink3 คือ วิดีโอแนะนำเส้นทางจากจุดปัจจุบันไปยังจุดต่อไปในทิศใต้
   - Vlink4 คือ วิดีโอแนะนำเส้นทางจากจุดปัจจุบันไปยังจุดต่อไปในทิศตะวันตก
   - Obs_flag คือ เป็น flag ที่ใช้บ่งบอกว่าจุดนั้นมีการซ่อมแซม หรือปรับปรุงถนน




**รูป 3.21 ตัวอย่างการเก็บข้อมูลใน Videos table**

จากรูป 3.21 จะเห็นว่าในโหนดที่ 10 (ตรงกลาง) มีการเก็บข้อมูลเป็น link1 เท่ากับ 0 หมายความว่าไม่มีโหนดใดที่เชื่อมต่อกับโหนดที่ 10 ในทิศเหนือ link2 เท่ากับ 5 แสดงถึงมีโหนด 5 อยู่ในทางทิศตะวันออก และเช่นเดียวกันกับ link3 (ทิศใต้) และ link4 (ทิศตะวันตก) ตามลำดับ ส่วน vlink นั้นจะทำการเก็บชื่อวิดีโอที่เดินทางไปยังจุดต่อไปในทิศนั้นที่อัพโหลดขึ้นยูทูป ตัวอย่างเช่น ที่โหนด 10 มี Vlink2 คือ yPfycEY_Ob4 หมายถึงมีตัวอย่างวิดีโอที่จะเดินไปยังโหนด 5 (ทิศตะวันออก) ซึ่งสามารถเข้าถึงวิดีโอด้วยลิ้งค์บนยูทูปคือ https://www.youtube.com/watch?v=yPfycEY_Qb4

**รูป 3.22 ตัวอย่างวิดีโอ https://www.youtube.com/watch?v=yPfycEY_Qb4**

3) Picture table เป็นโครงตารางที่ไว้เก็บลิงค์ของชุดภาพอาคารและบริเวณในสถาบัน มี Pid เป็น Primary key ประกอบกับ id ของตาราง place เป็น foreign key ประกอบไปด้วย

- Pid คือ รหัสของรูปภาพ (Primary Key)
- Path คือ เส้นทางที่เข้าถึงรูปภาพ

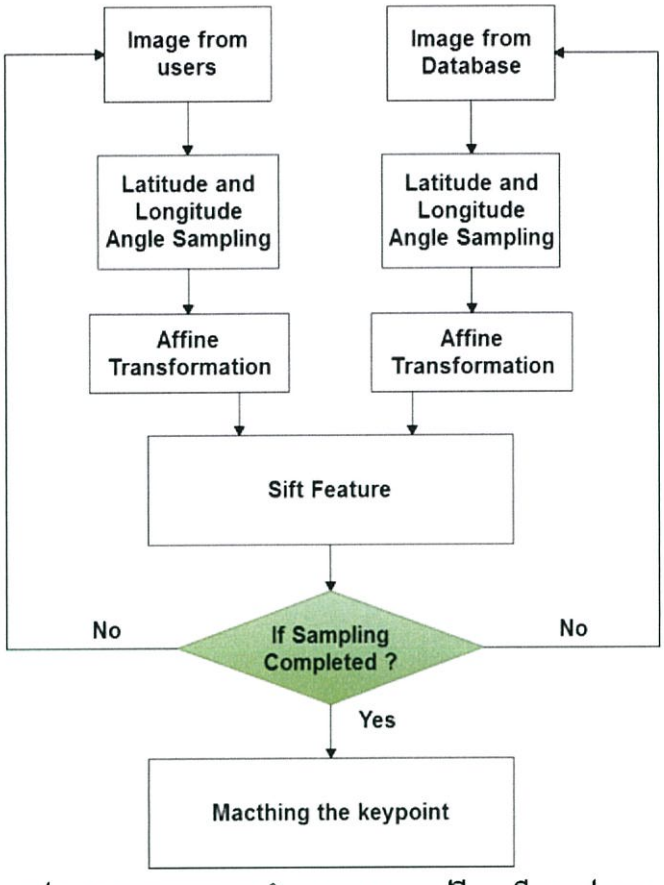
- Pv_flag คือ เป็น flag ใช้ในการเลือกเป็นภาพแนะนำอาคาร
- Bid คือ รหัสของสถานที่ (Foreign Key)

4) Division Table เป็นโครงตารางที่แสดงถึงหน่วยงานของสถานที่ในสถาบัน สถาบัน มี Did เป็น Primary Key ประกอบกับ id ของตาราง Location เป็น foreign key ประกอบไปด้วย

- Did คือ รหัสของข้อมูลของหน่วยงาน (Primary Key)
- Name คือ ชื่อของหน่วยงานนั้น
- Address คือ ที่อยู่ของหน่วยงานนั้น
- Email คือ อีเมล์ที่ติดต่อได้ของหน่วยงานนั้น
- Tel คือ เบอร์โทรศัพท์ของหน่วยงานนั้น
- Fax คือ เบอร์แฟกซ์ของหน่วยงานนั้น
- www คือ เว็บไซต์ของหน่วยงานนั้น
- Bid คือ รหัสของสถานที่ (Foreign Key)

## 3.7 การทำงานในส่วนเปรียบเทียบรูปภาพ

สำหรับแนวคิดในการเปรียบเทียบรูปภาพที่อยู่ในฐานข้อมูล กับ รูปภาพที่ถ่ายจากผู้ใช้งานจะอ้างอิงจากวิทยานิพนธ์เรื่อง "ASIFT: An Algorithm for Fully Affine Invariant Comparison" [3] ซึ่งโครงงานหัวข้อนี้ได้ทำการนำเทคนิคจากวิทยานิพนธ์ใน [3] มาประยุกต์ใช้ แสดงเป็นภาพรวมการทำงานดังนี้




**รูป 3.23 ภาพรวมการทำงานของการเปรียบเทียบรูปภาพ**

ในส่วนของการเปรียบเทียบภาพนั้น ทางระบบจะทำการรับค่าพิกัดปัจจุบันจากผู้ใช้มาก่อนเพื่อนำมาเป็นค่าอ้างอิงในการเลือกข้อมูลชุดภาพของอาคารแต่ละอาคารภายในสถาบัน พอได้ภาพที่ใกล้เคียงกับพิกัดปัจจุบันที่เป็นค่าอ้างอิงแล้ว ระบบจะทำการเปรียบเทียบภาพแต่ละภาพในฐานข้อมูลที่ได้มา กับ ภาพถ่ายตัวอาคารจากผู้ใช้งานทีละภาพ โดยระบบจะทำการเลือกคู่ภาพที่มีจำนวนคุณลักษณะที่เหมือนกันมากที่สุดเพื่อยืนยันตำแหน่งปัจจุบันให้แก่ผู้ใช้งาน โดยใช้พิกัดจากสถานที่นั้น แต่ถ้าหากว่าในการเปรียบเทียบภาพครั้งนั้นไม่มีคู่ใดที่ใกล้เคียงกันเลยระบบจะทำการคืนค่าภาพอาคารใกล้เคียงให้แก่ผู้ใช้งาน และแจ้งเตือนว่าไม่พบการจับคู่กันอย่างใด ถ้าผู้ใช้งานเลือกภาพอาคารบริเวณใกล้เคียงที่ระบบทำการส่งกลับมาให้ก็จะใช้พิกัดของอาคารนั้นเป็นพิกัดปัจจุบันแทน แต่ถ้าหากไม่เลือกก็จะกลับสู่กระบวนการเริ่มต้นต่อไป

### 3.8 การทำงานในส่วนของการแนะนำเส้นทาง

ในส่วนของแนวคิดในการค้นหาเส้นทางที่ดีที่สุดนั้นในโครงการนี้เราเลือกใช้การค้นหาแบบเอ-สตาร์ (A* Algorithm) ซึ่งอัลกอริทึมนี้จะค้นหาแบบแนวกว้างเพื่อค้นหาเส้นทางที่เป็นไปได้ซึ่งผลลัพธ์ที่ได้จากอัลกอริทึมนี้ถือว่าเป็นเส้นทางที่ดีที่สุด โดยใช้ฟังก์ชันฮิวริสติกมาทำการทำนายค่าระยะทางที่จะต้องใช้ แล้วเลือกเส้นทางที่เสียค่าใช้จ่ายน้อยที่สุดมีสมการเป็น

$$F(n) = g(n) + h(n) \tag{3.1}$$

โดย

F(n) คือค่าระยะทางทั้งหมดที่ต้องใช้

g(n) คือค่าระยะทางจากโหนดเริ่มต้นมายังจุดปัจจุบัน

h(n) คือ ค่าประมาณฮิวริสติกที่ยอมรับได้ ที่จะไปยังจุดต่อไป

ซึ่งค่าประมาณฮิวริสติกที่ใช้ในโครงงานนี้จะใช้ระยะทางระหว่างจุดทั้งสองจุดซึ่งคำนวณจาก

$$h(n) = 2R\arcsin\sqrt{\left(\frac{1}{2} - \frac{\cos((lat2-lat1)*p)}{2} + \frac{\cos(lat1*p)*\cos(lat2*p)*(1-\cos((lng2-lng1)*p))}{2}\right)} \tag{3.2}$$

โดย

R คือ รัศมีของโลกมีค่าประมาณ 6,371 กิโลเมตร

lat1 คือ ละติจูดของจุดที่หนึ่ง

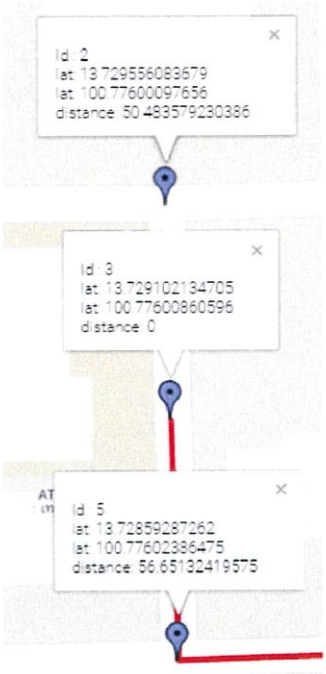
lat2 คือ ละติจูดของจุดที่สอง

lng1 คือ ลองจิจูดของจุดที่หนึ่ง

lng2 คือ ลองจิจูดของจุดที่สอง

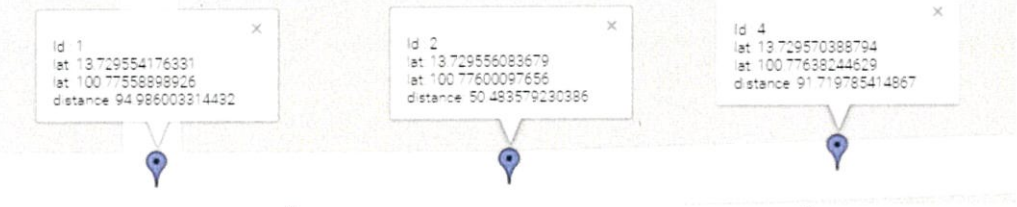
p คือ ค่าพายในหน่วยเรเดียน (Radians)

ซึ่งสมการ 3.2 นี้เป็นสมการหาระยะทางบนพื้นผิวโลกระหว่างพิกัดสองพิกัดด้วยกันโดยไม่คำนึงถึงสภาพแวดล้อมทั้งสิ้น




**รูป 3.24 ขั้นตอนการทำงานของอัลกอริทึมเอสตาร์รอบที่ 1**

จากรูป 3.24 สมมติว่าเราเริ่มที่จุดที่มีไอดีเท่ากับ 3 พออัลกอริทึมเอสตาร์ทำงาน เอสตาร์จะทำงานโหนดใกล้เคียงที่สามารถเดินทางไปได้จากรูปจะเห็นว่าที่โหนด 3 สามารถเดินไปได้สองทางคือโหนด 2 และโหนด 5 โดยที่โหนด 2 มีค่าฟังก์ชันฮิวริสติกเป็น 50.48 เมตร และโหนด 5 มีค่าฮิวริสติกเป็น 56 เมตรสำหรับรอบนี้เอสตาร์จะเลือกไปข้างบนคือโหนด 2 เพราะว่ามีค่าฮิวริสติกรวมกับระยะทางทั้งหมดที่น้อยกว่าโหนด 5 และทำการเก็บข้อมูลของโหนด 5 ไว้เพื่อพิจารณาต่อไป โดยในรอบแรกนี้ระยะทางทั้งหมดมีค่าเป็น 0




**รูป 3.25 ขั้นตอนการทำงานของอัลกอริทึมเอสตาร์รอบที่ 2**

จากรูป 3.25 เมื่อเดินมาที่โหนด 2 แล้วตอนนี้จะมีค่า F(n) เท่ากับ 50.48 เมตร แล้วเมื่อพิจารณาจากอัลกอริทึมนี้ต่อที่โหนด 2 สามารถเดินไปยังโหนดที่ 1 โดยมีค่า F(n) เท่ากับ ระยะทางทั้งหมด + ฟังก์ชันฮิวริสติกซึ่งมีค่าเป็น 50.48 + 94.98 = 141.46 เมตร และเช่นเดียวกันก็สามารถเดินไปยังโหนดที่ 4 ด้วยค่า F(n) เท่ากับ 141.71 เมตร แต่ว่าด้วยอัลกอริทึมเอสตาร์นั้นจะมีการเก็บโหนดที่ไม่ได้เลือกไว้ในแต่ละรอบมาพิจารณาด้วยซึ่งจากการทำงานรอบที่ 1 นั้นมีการเก็บโหนดที่ 5 ไว้อยู่

ซึ่งก็ต้องนำมาพิจารณาโดยโหนดที่ 5 นั้นจะมีค่า F(n) เท่ากับ 56.65 เมตร ซึ่งผลการทำงานรอบที่สองนี้จะทำการเลือกโหนดที่ 5 เพราะว่ามีค่า F(n) น้อยที่สุด แล้วจะทำงานเช่นนี้ไปเรื่อยจนกว่าจะเจอจุดที่เป็นสถานที่ปลายทาง

จุดเด่นของอัลกอริทึมนี้คือเวลาเลือกเส้นทางเอ-สตาร์นั้นจะเลือกเส้นทางที่มีค่าน้อยที่สุดก่อนเสมอ โดยยังคงเก็บเส้นทางสำรองที่มันสามารถไปไว้ได้อยู่ด้วย และถ้าหากเส้นทางที่ไปตอนแรกนั้นกลับมันสามารถย้อนกลับไปยังจุดที่ผ่านมาแล้วเลือกไปยังเส้นทางที่มีค่าใช้จ่ายน้อยกว่าปัจจุบันได้

# บทที่ 4

# การทดลองและผลการทดลอง

## 4.1 ผลการทดลองของโครงงาน

ในบทนี้จะกล่าวถึงการทดลอง และผลการทดลองของระบบค้นหาเส้นทาง และจำลองวิดีโอเส้นทาง ซึ่งการทำงานของระบบนั้นประกอบไปด้วย ส่วนของเว็บแอพพลิเคชั่นที่เป็นส่วนติดต่อกับผู้ใช้งาน ส่วนของการนำทาง ส่วนของการแสดงผลวิดีโอแนะนำเส้นทาง และส่วนของการเปรียบเทียบภาพถ่ายซึ่งผลของการทดลองได้ผลสรุปดังต่อไปนี้

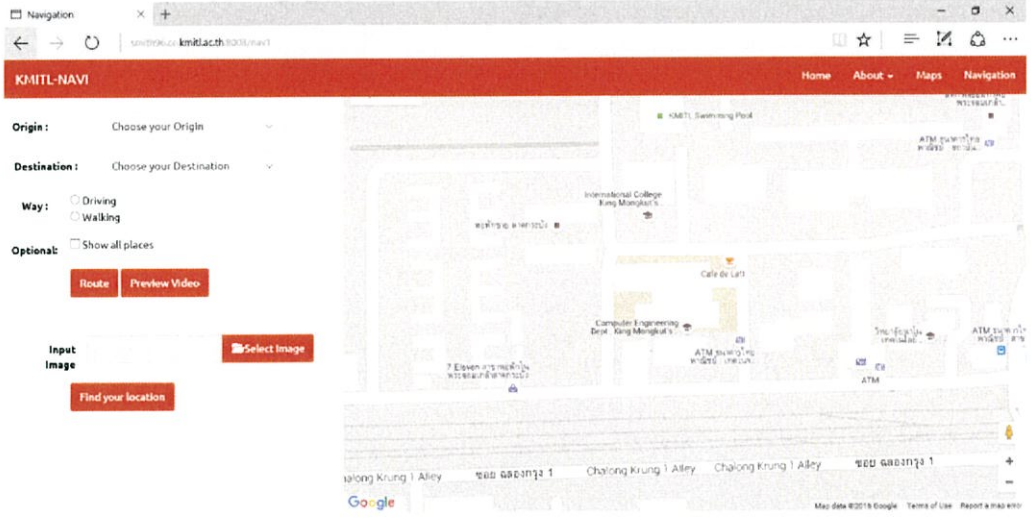
### 4.1.1 ส่วนของแอพพลิเคชั่น

การใช้งานในส่วนของแอพพลิเคชั่นประกอบด้วยการใช้งาน 3 ส่วน อธิบายเพิ่ม ว่ามีส่วนไหนบ้าง ผลการทดสอบการใช้งานในแต่ละส่วนเป็นดังนี้

#### 4.1.1.1 ส่วนการค้นหาเส้นทาง

ในส่วนการทำงานของการค้นหาเส้นทางนั้นมีไว้เพื่อให้ผู้ใช้ทำการค้นหาเส้นทางที่ผู้ใช้ต้องการ แบ่งออกเป็น 2 ประเภท

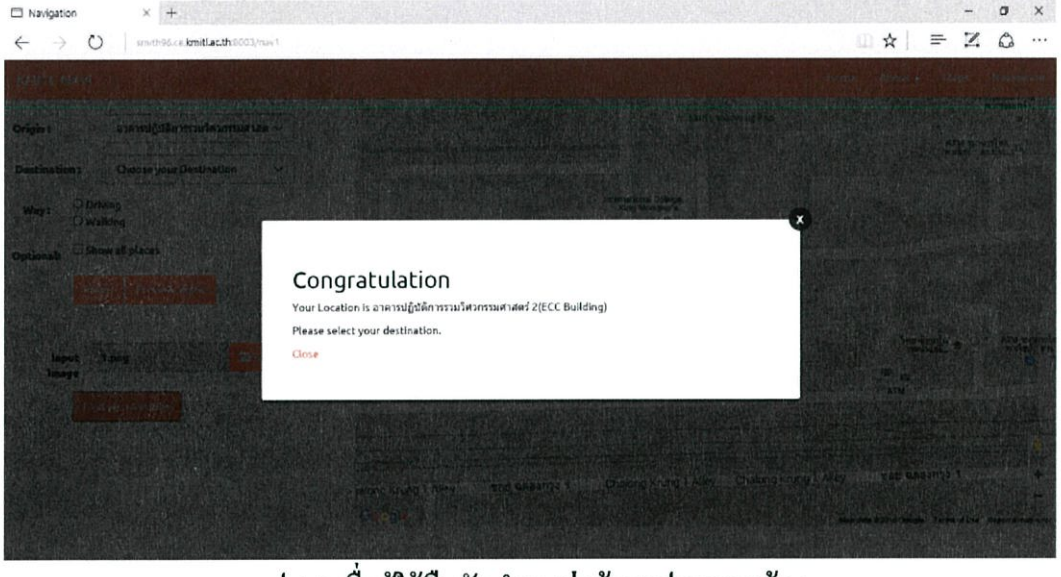
1) การค้นหาเส้นทางจาก ภายในสถาบัน-ภายในสถาบัน โดยผู้ใช้สามารถกำหนดจุดเริ่มต้นได้ด้วยการเลือกสถานที่ และการอัพโหลดภาพถ่าย และจุดปลายทางด้วยการเลือกสถานที่ พร้อมทั้งแสดงสถานที่ใกล้เคียงและวีดีโอนำทาง




รูป 4.1 หน้าหลักของส่วนการค้นหาเส้นทาง

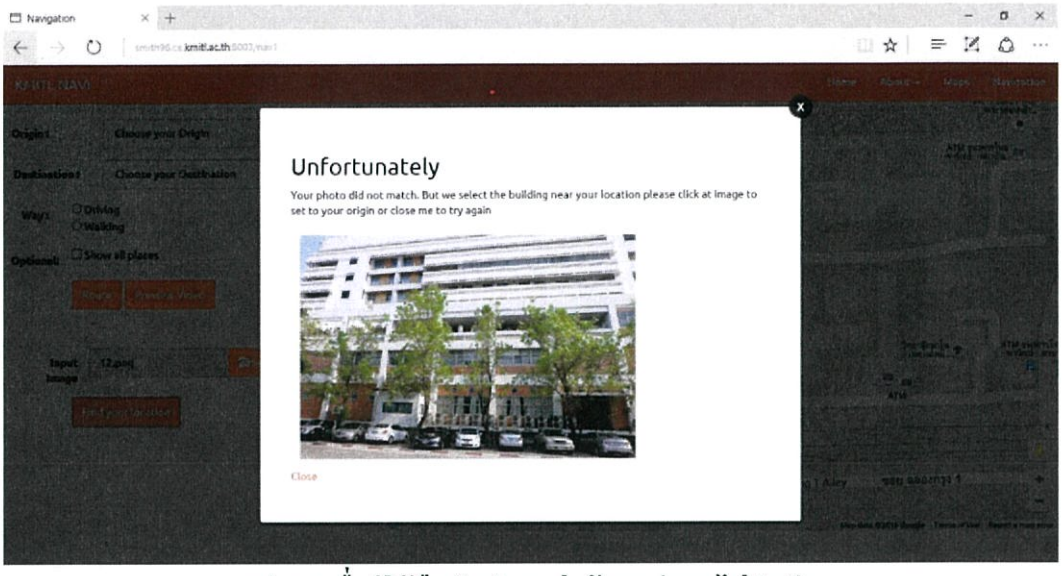
การทำงานของแอพพลิเคชั่นนั้นสมมติว่า ณ เวลานั้น ผู้ใช้งานยืนอยู่ที่ตึกอาคารเรียนรวมวิศวกรรมศาสตร์ 2 (ตึก ECC) แล้วจะเดินทางไปยังอาคารวิศวกรรมโทรคมนาคม

ในการใช้งานผู้ใช้งานนั้นหากผู้ใช้งานรู้ตำแหน่งปัจจุบันหรือชื่ออาคารที่ตนเองอยู่ใกล้ที่สุด ผู้ใช้งานสามารถระบุตำแหน่งต้นทางจากชื่ออาคาร หรือหน่วยงานนั้นได้เลย แต่ถ้าหากผู้ใช้งานไม่รู้ว่าอยู่ในที่ส่วนใดของสถาบัน ผู้ใช้งานสามารถทำการถ่ายภาพอาคารบริเวณใกล้เคียงเพื่อยืนยันตำแหน่งปัจจุบันได้เช่นกัน



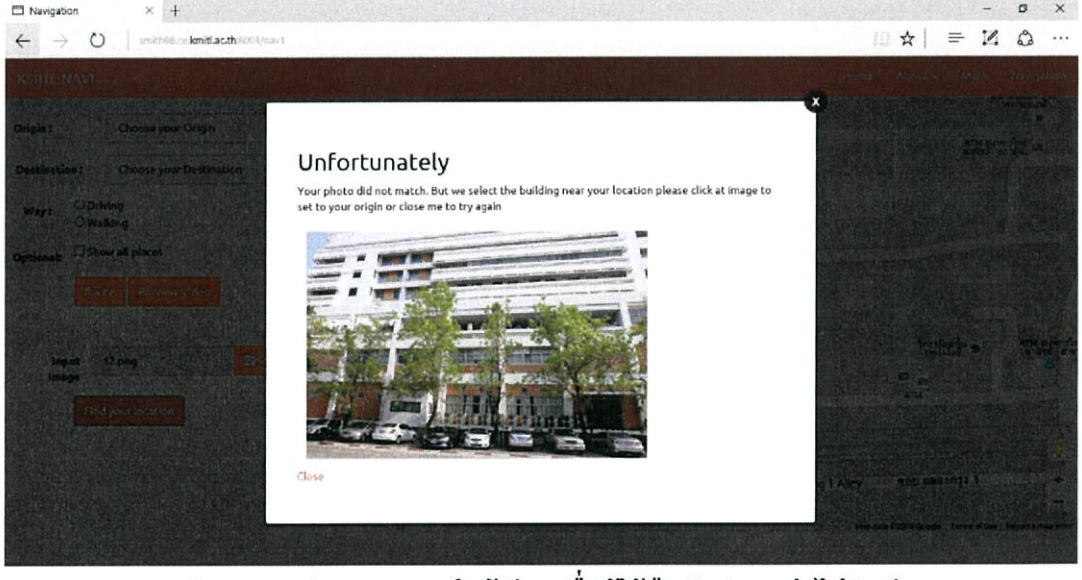

**รูป 4.2 เมื่อผู้ใช้ยืนยันตำแหน่งด้วยรูปภาพถูกต้อง**

แต่ถ้าหากการยืนยันตำแหน่งปัจจุบันด้วยภาพถ่ายนั้นพบว่าไม่ตรงกับอาคารใดเลย




**รูป 4.3 เมื่อผู้ใช้ยืนยันตำแหน่งด้วยรูปภาพไม่สำเร็จ**

ระบบจะทำการแนะนำอาคารใกล้เคียงให้แก่ผู้ใช้งานว่าใช่อาคารนั้นหรือไม่ ถ้าหากผู้ใช้งานเลือกอาคารที่ระบบแนะนำ ระบบจะทำการระบุอาคารนั้นเป็นสถานที่ปัจจุบัน แต่ถ้าหากไม่ใช่ จะเข้าสู่กระบวนการในข้อ 1 ใหม่อีกครั้งนึง



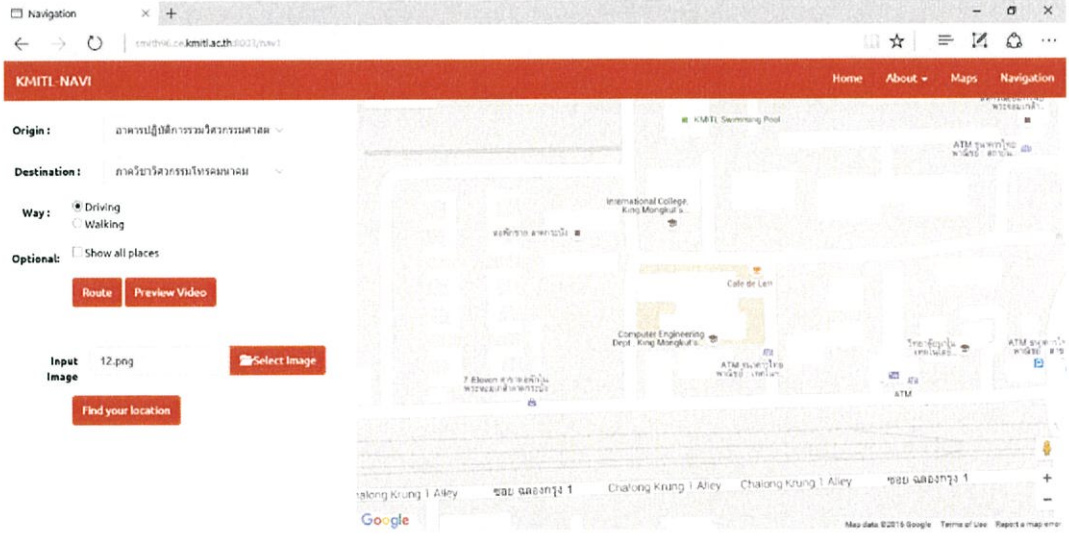

**รูป 4.4 แนะนำภาพอาคารใกล้เคียง เมื่อผู้ใช้ยืนยันตำแหน่งไม่สำเร็จ**

หลังจากระบบทำการประมวลผล แล้วระบุตำแหน่งปัจจุบันแล้วให้แก่ผู้ใช้งานแล้ว ให้ผู้ใช้เลือกสถานที่ปลายทางจากลิสต์ชื่ออาคารทั้งหมดในช่อง ซึ่งใน

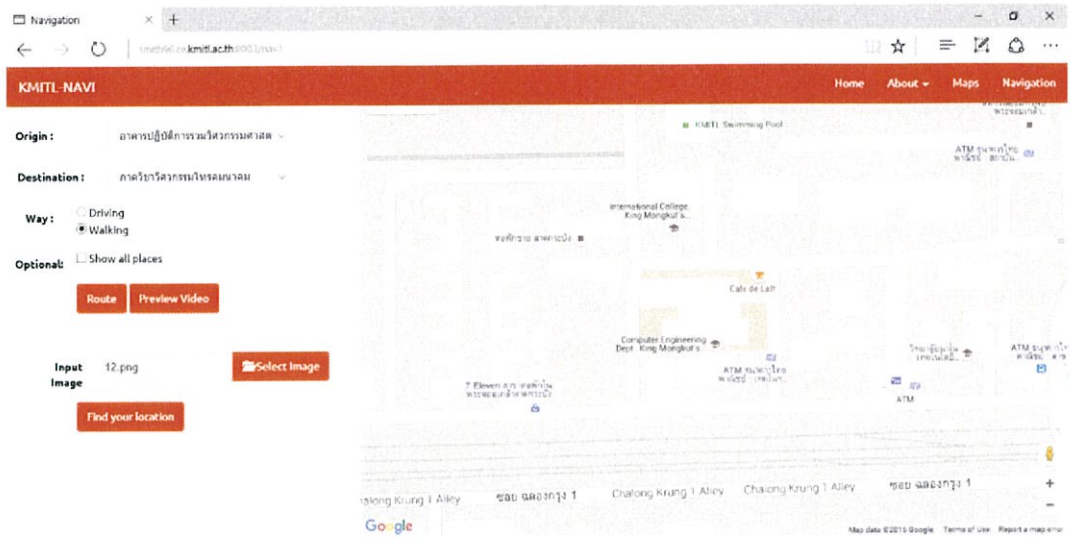
เหตุการณ์นี้คือผู้ใช้งานเลือกสถานที่ปลายทางเป็นอาคารวิศวกรรมโทรคมนาคม

**รูป 4.5 การเลือกสถานที่ปลายทาง**

หลังจากเลือกอาคารปลายทางเสร็จแล้ว ให้ผู้ใช้งานเลือกประเภทของการเดินทางโดยมีด้วยกันทั้งสองแบบคือ รถยนต์ หรือ เดิน




**รูป 4.6 เลือกประเภทการเดินทางด้วยรถยนต์**

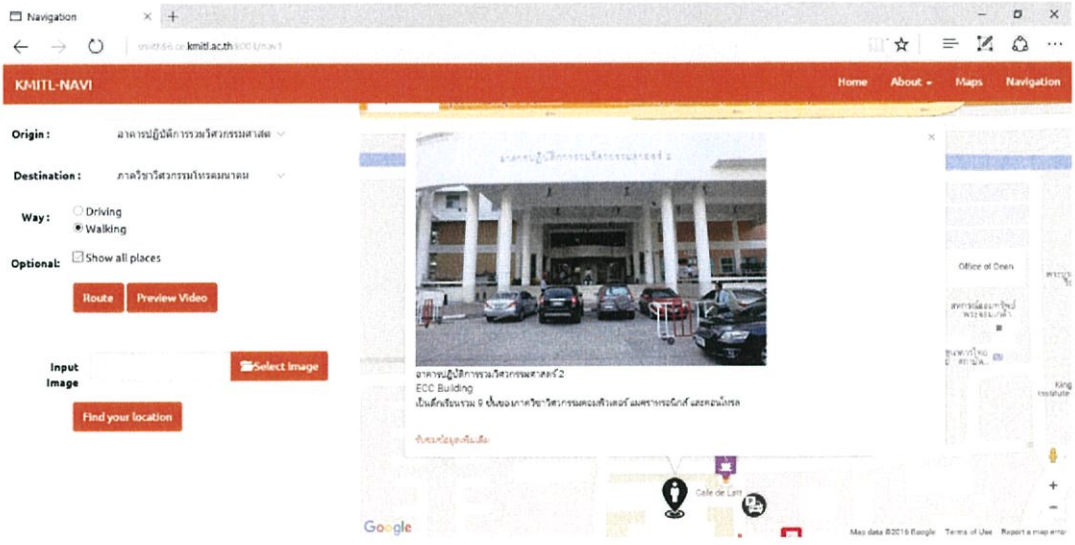



**รูป 4.7 เลือกประเภทการเดินทางด้วยการเดิน**

เมื่อเลือกประเภทแล้วให้ผู้ใช้งานกดปุ่ม ROUTE หลังจากนั้นระบบจะทำการประมวลผล แล้วจะทำการลากเส้นทางจากสถานที่ต้นทาง ไปยังสถานที่ปลายทาง พร้อมทั้งวิดีโอแนะนำเส้นทาง รวมไปถึงสถานที่ ร้านค้า สิ่งอำนวยความสะดวกรอบตัวผู้ใช้งานในระยะ 50 เมตร

**รูป 4.8 การนำทางจาก ภายในสถาบัน-ภายในสถาบัน ด้วยการขับรถ หมุดรูปสถานที่แสดงถึงสถานที่สำคัญระหว่างทางที่ผ่าน**

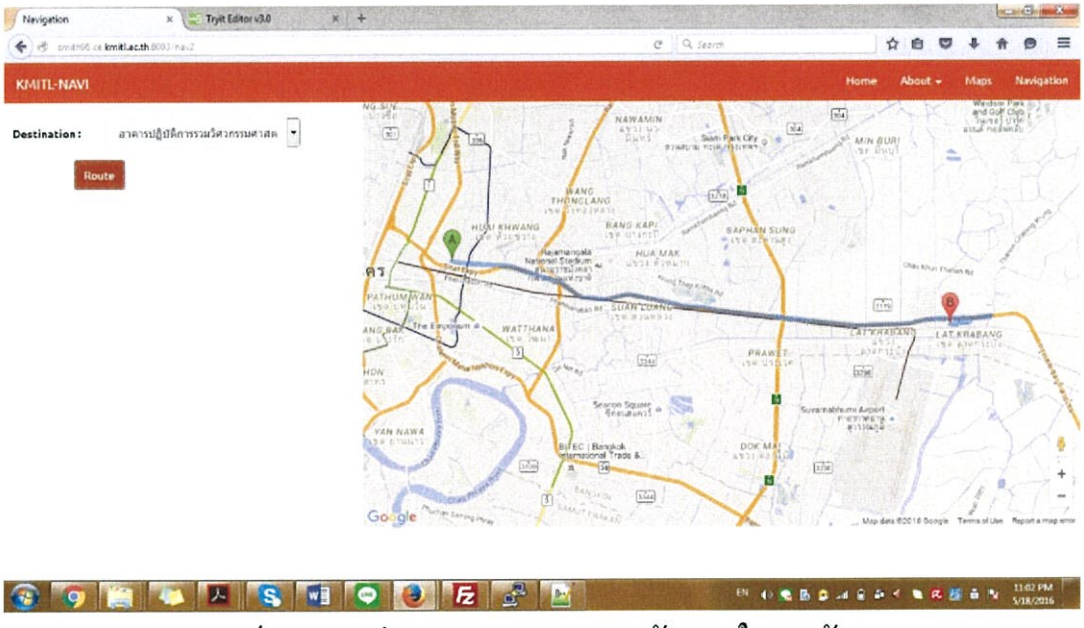
**รูป 4.9 การนำทางจาก ภายในสถาบัน-ภายในสถาบัน ด้วยการเดิน หมุดรูปสถานที่แสดงถึงสถานที่สำคัญระหว่างทางที่ผ่าน**




**รูป 4.10 รายละเอียดเมื่อเลือกที่หมุดรูปสถานที่**

**รูป 4.11 วีดีโอนำทางเมื่อเลือกที่หมุดสีเหลือง**

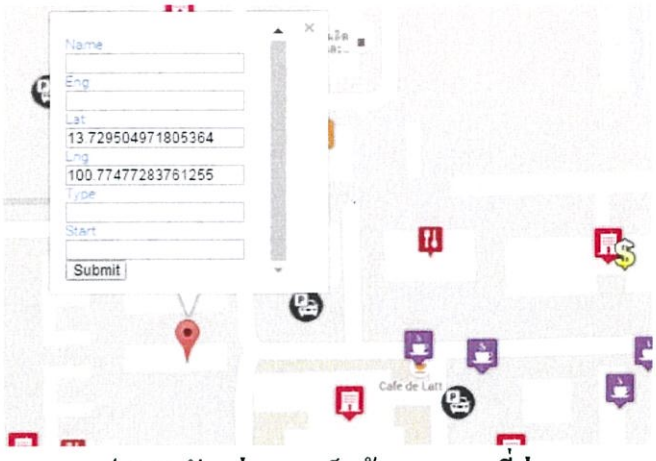
2) การค้นหาเส้นทางจาก ภายนอกสถาบัน-ภายในสถาบัน โดยระบบจะทำการกำหนดจุดเริ่มต้นให้จากตำแหน่งพิกัดปัจจุบันของผู้ใช้ และให้ผู้ใช้ทำหารเลือกจุดปลายทางด้วยการเลือกสถานที่จากลิสต์ชื่ออาคารทั้งหมดในช่อง




**รูป 4.12 การนำทางจาก ภายนอกสถาบัน-ภายในสถาบัน**

### 4.1.2 ส่วนของเซิร์ฟเวอร์

สำหรับส่วนของเครื่องเซิร์ฟเวอร์นั้นจะมีการสร้างฐานข้อมูลไว้เพื่อเก็บสถานที่ภายในสถาบัน ภาพถ่ายอาคาร รายละเอียดของหน่วยงานภายในอาคารนั้น ตำแหน่งพิกัดในสถาบันเอาไว้ใช้นำทาง และลิ้งค์ของวิดีโอแนะนำเส้นทาง




**รูป 4.13 ตัวอย่างการเก็บข้อมูลสถานที่ต่างๆ**

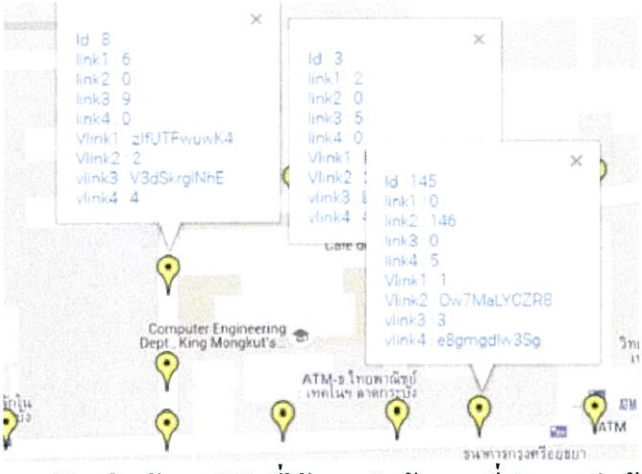
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Edit Copy Delete | 15 | ภาควิชาวิศวกรรมเครื่องกล | Department of Mechanical Engineering (ME) | 13.727683671068958 | 100.77397558830853 | 1 | 255 | 2016-04-02 23:45:42 |
| Edit Copy Delete | 16 | ภาควิชาวิศวกรรมอุตสาหการ | Department of Industrial Engineering (IE) | 13.727707121303395 | 100.77335663125268 | 1 | 255 | 2016-04-03 00:16:04 |
| Edit Copy Delete | 17 | อาคารเรียนรวมวิศวกรรมศาสตร์ | E12 | 13.7274465621743 | 100.77236894510881 | 1 | 278 | 2016-04-03 00:18:54 |
| Edit Copy Delete | 18 | ภาควิชาอิเล็กทรอนิกส์ | Department of Electronics Engineering | 13.727579447321 37 | 100.77544812109863 | 1 | 218 | 2016-04-03 00:33:08 |
| Edit Copy Delete | 19 | ภาควิชาวิศวกรรมการวัดคุม | Department of Instrumention Engineering (I) | 13.72739184474362 | 100.77443692825909 | 1 | 241 | 2016-04-03 09:07:07 |
| Edit Copy Delete | 20 | ภาควิชาวิศวกรรมโยธา | Department of Civil Engineering | 13.726959315692556 | 100.77435914427952 | 1 | 241 | 2016-04-03 09:08:22 |

**รูป 4.14 ตัวอย่างข้อมูลในตารางฐานข้อมูล**

**รูป 4.15 ตัวอย่างการเก็บข้อมูลภาพถ่ายอาคารต่างๆ**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | img_build/1/1_WTdS7.png | 0 | 1 | 2016-04-08 16:52:37 |
| Edit Copy Delete | 2 | img_build/1/1_Nr6Wg.png | 0 | 1 | 2016-04-08 16:52:37 |
| Edit Copy Delete | 3 | img_build/1/1_mWKe1.png | 0 | 1 | 2016-04-08 16:52:46 |
| Edit Copy Delete | 4 | img_build/1/1_dGmOa.png | 0 | 1 | 2016-04-08 16:52:47 |
| Edit Copy Delete | 5 | img_build/17/17_GajUk.png | 0 | 17 | 2016-04-08 16:54:11 |
| Edit Copy Delete | 6 | img_build/17/17_qCy6e.png | 0 | 17 | 2016-04-08 16:54:11 |
| Edit Copy Delete | 7 | img_build/17/17_AThua.png | 0 | 17 | 2016-04-08 16:54:51 |

**รูป 4.16 ตัวอย่างข้อมูลของภาพถ่ายที่เก็บอยู่ในฐานข้อมูล**




**รูป 4.17 ตัวอย่างข้อมูลพิกัดที่ใช้แนะนำเส้นทางที่ทำการเก็บข้อมูล**

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | 13.729554176331 | 100.77558898926 | 0 | 2 | 0 | 6 | 0 | 13v9Ei-0kOk | 0 | HkoCkgaCNpA |
| Edit Copy Delete | 2 | 13.729556083679 | 100.77600097656 | 0 | 1 | 3 | 4 | 1 | cDiErcnfMzc | FeGA4xUrKZA | _3VVGf4XRJU |
| Edit Copy Delete | 3 | 13.729102134705 | 100.77600860596 | 2 | 0 | 5 | 0 | Elr6RZhAiRg | 2 | Lv7rztPDGdk | 4 |
| Edit Copy Delete | 4 | 13.729570388794 | 100.77638244629 | 0 | 124 | 0 | 2 | 1 | dggW7onI6Uo | 3 | gIV7IfY1q94 |
| Edit Copy Delete | 5 | 13.72859287262 | 100.77602386475 | 2 | 145 | 0 | 10 | zhkm7GWvaxI | bODzsuhUUfw | 3 | wtMCpYSVuVE |
| Edit Copy Delete | 6 | 13.729568481445 | 100.77510070801 | 7 | 1 | 8 | 0 | XGL_x_1YYMc | Bb3ismKMEI | b0SJ8HbP6B0 | 4 |
| Edit Copy Delete | 7 | 13.729907989502 | 100.77504730225 | 32 | 0 | 6 | 0 | sLGZ8hFW7qo | 2 | 4QeKEkDKk78 | 4 |
| Edit Copy Delete | 8 | 13.729187965393 | 100.77505493164 | 6 | 0 | 9 | 0 | zlfUTFwuwK4 | 2 | V3dSkrgINhE | 4 |
| Edit Copy Delete | 9 | 13.728791236877 | 100.77504730225 | 8 | 0 | 12 | 0 | OBsCgthGbSE | 2 | q16aBmRZKhk | 4 |

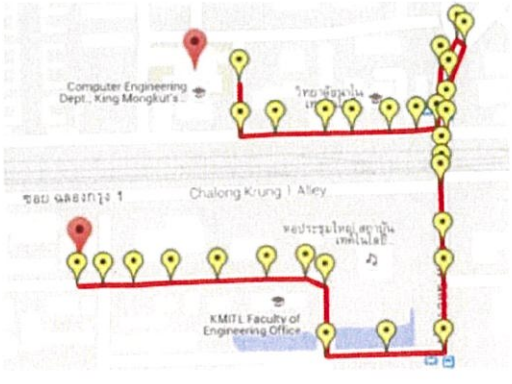
**รูป 4.18 ตัวอย่างข้อมูลพิกัดที่ใช้แนะนำเส้นทางในฐานข้อมูล**

### 4.1.3 ส่วนของการนำทาง

ในส่วนของการนำทางนั้น เส้นทางที่ได้จะเป็นเส้นทางที่สั้นที่สุดจากจุดใดจุดหนึ่งภายในสถาบัน โดยตัวอัลกอริทึมคือเอสตาร์ (A* Algorithm) [4] ที่เลือกมาใช้งานนั้นมีทั้งผลลัพธ์บางจุดดีกว่าที่ทางกูเกิลให้ข้อมูลมา และมีบางจุดที่ได้ผลรวมของระยะทางเท่ากัน ในทั้งรูปแบบการเดินทางแบบทางรถยนต์ และการเดิน

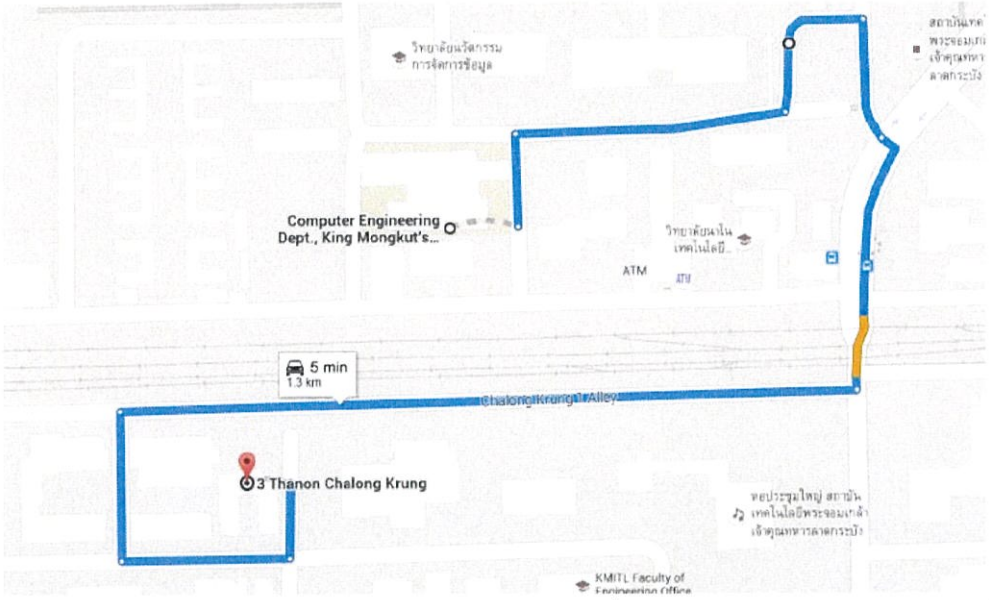
**ตาราง 4.1 ตัวอย่างการเปรียบเทียบระยะทางที่ใช้จากสถานที่ต้นทางไปยังสถานที่ปลายทางโดยรถยนต์ที่ได้จากระบบแนะนำและระยะทางที่กูเกิลแนะนำ**

| สถานที่ต้นทาง | สถานที่ปลายทาง | ระยะทางที่ระบบแนะนำ (กิโลเมตร) | ระยะทางที่กูเกิลแนะนำ (กิโลเมตร) |
|---|---|---|---|
| อาคารปฏิบัติการรวมวิศวกรรมศาสตร์ 2 (ECC) | ภาควิชาวิศวกรรมวัดคุม | 1.178 | 1.3 |
| อาคารเรียนรวมวิศวกรรมศาสตร์ (E12) | หอประชุมเจ้าพระยาสุรวงษ์ไวยวัฒน์ (วร บุนนาค) | 0.864 | 1.3 |
| สำนักหอสมุดกลาง | ภาควิชาวิศวกรรมโยธา | 0.581 | 0.593 |
| ภาควิชาวิศวกรรมวัดคุม | สมาคมศิษย์เก่า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง | 1.383 | 1.5 |




Go to: 213 Total Distance: 978.33606528881
Go to: 218 Total Distance: 1030.2945521
Go to: 219 Total Distance: 1080.5846477577
Go to: 221 Total Distance: 1120.1935834493
Go to: 224 Total Distance: 1152.3384971313
Go to: 241 Total Distance: 1178.7104374689

**รูป 4.19 เส้นทางจากอาคารปฏิบัติการรวมวิศวกรรมศาสตร์ 2 ไปยัง ภาควิชาวิศวกรรมวัดคุมโดยรถยนต์พร้อมแสดงระยะทางที่ใช้จากระบบ**




**รูป 4.20 เส้นทางจากอาคารปฏิบัติการรวมวิศวกรรมศาสตร์ 2 ไปยัง ภาควิชาวิศวกรรมวัดคุมโดยรถยนต์พร้อมแสดงระยะทางที่ใช้จากกูเกิล**

**ตาราง 4.2 ตัวอย่างการเปรียบเทียบระยะทางที่ใช้จากสถานที่ต้นทางไปยังสถานที่ปลายทางโดยการเดินที่ได้จากระบบแนะนำและระยะทางที่กูเกิลแนะนำ**

| สถานที่ต้นทาง | สถานที่ปลายทาง | ระยะทางที่ระบบแนะนำ (กิโลเมตร) | ระยะทางที่กูเกิลแนะนำ (กิโลเมตร) |
|---|---|---|---|
| อาคารปฏิบัติการรวมวิศวกรรมศาสตร์ 2 (ECC) | ภาควิชาวิศวกรรมวัดคุม | 0.564 | 1.0 |
| อาคารปฏิบัติการรวมวิศวกรรมศาสตร์ 2 (ECC) | ภาควิชาวิศวกรรมเครื่องกล (ME) | 0.445 | 0.750 |
| อาคารเรียนรวมวิศวกรรมศาสตร์ (E12) | วิทยาลัยนาโนเทคโนโลยี | 0.764 | 0.850 |
| อาคารวิศวกรรมศาสตร์ (ตึก A) | อาคารเรียนรวมสมเด็จพระเทพฯ | 0.581 | 0.750 |

ก) ข)

**รูป 4.21 เส้นทางจากอาคารเรียนรวมสมเด็จพระเทพไปยังตึกวิศวกรรมศาสตร์ (ตึก A) โดยการเดิน**

ก) เส้นทางจากกูเกิล

ข) เส้นทางจากระบบ

### 4.1.4 ส่วนของวิดีโอแนะนำเส้นทาง

ในส่วนของวิดีโอนั้น เนื่องจากภาพที่ทำการร้องขอจาก Google Maps Streetview API ในบางพิกัดที่ทำการร้องขอไปอาจจะได้ภาพที่มีความคาดเคลื่อนมาบ้าง สาเหตุมาจากการเก็บข้อมูลของทางกูเกิลที่ทำการเก็บข้อมูลด้วยกล้อง 360 องศา จึงทำให้ต้องแก้ปัญหาความคลาดเคลื่อนของภาพนี้โดยการลงไปถ่ายภาพจากสถานที่จริงได้แค่บางจุด แต่บางจุดไม่สามารถลงไปเก็บข้อมูลได้เนื่องจากมีสภาพการจราจรหนาแน่นทั้งวัน

ก) ข) ค)

**รูป 4.22 วิดีโอตัวอย่างที่ทำการร้องขอมาจาก Google Streetview API**

ก) ภาพตัวอย่างวิดีโอที่มีความผิดพลาด (กระโดดเลน)

ข) ภาพตัวอย่างวิดีโอที่ถูกต้อง

ค) ภาพตัวอย่างวิดีโอที่ถูกต้อง

จากรูป ข) และ ค) จะเห็นว่าเส้นทางที่ถูกนั้นต้องอยู่ที่เลนซ้าย แต่ว่ารูป ก) นั้นเกิดการผิดพลาดจากการเก็บข้อมูลของทางเอพีไอเองซึ่งได้รูปที่กระโดดเลนจากเลนซ้ายกลายเป็นเลนขวาแทน

**4.1.5 ส่วนของการเลือกชุดภาพเพื่อเปรียบเทียบ**

ในส่วนของการเลือกชุดภาพของสถานที่เพื่อเปรียบเทียบนั้น ขั้นตอนในการเลือกจะคัดเลือกจากจุดที่ผู้ใช้งานอยู่โดยรับค่าพิกัดของผู้ใช้ปัจจุบัน แล้วทำการค้นหาสถานที่ที่ใกล้เคียงที่เป็นอาคารจากผู้ใช้งานเป็นรัศมีรอบตัวโดยวัดเป็นเมตร ซึ่งหาระยะทางได้จากสมการ 4.1

$$N = Racos\sqrt{\cos(lat1) * \cos(lat2) * \cos(lng2 - lng1) + \sin(lat1) * \sin(lat2)} \qquad (4.1)$$

R คือ รัศมีของโลกมีค่าประมาณ 6,371,000 เมตร

lat1 คือ ละติจูดของจุดที่หนึ่งในหน่วยเรเดียน

lat2 คือ ละติจูดของจุดที่สองในหน่วยเรเดียน

lng1 คือ ลองจิจูดของจุดที่หนึ่งในหน่วยเรเดียน

lng2 คือ ลองจิจูดของจุดที่สองในหน่วยเรเดียน

โดยพิกัดที่มี N น้อยกว่ารัศมีที่กำหนดไว้จะถือว่าอยู่ในบริเวณรอบพิกัดนั้นเป็น จากการทดลองพบว่าค่ารัศมีของพิกัดปัจจุบันมีผลต่อการคัดเลือกสถานที่ที่ใกล้เคียง โดยค่าที่เหมาะสมคือ 50 เมตร ซึ่งถ้าหากกำหนดให้มีค่าต่ำกว่า 50 เมตรจะได้อาคารที่อยู่ใกล้ตัวของผู้ใช้งานจริงเท่านั้น ไม่ใช่อาคารที่ใกล้เคียง และถ้าหากใส่มากกว่า 50 เมตร อาคารที่ได้จะมีได้อาคารที่ไม่ได้อยู่ใกล้ตัวผู้ใช้งานติดมาด้วย ซึ่งให้ผลการทดลองดังต่อไปนี้

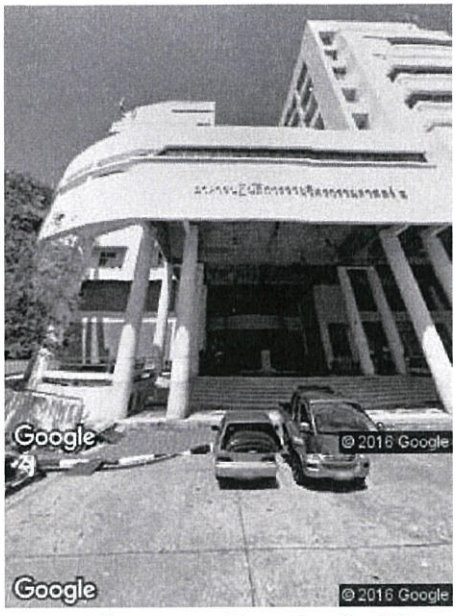
**รูป 4.23 จำลองการค้นหาอาคารใกล้เคียงในระยะ 50 เมตรที่พิกัด 13.7282473,100.7759738**

สมมติว่าตอนนี้ผู้ใช้งานยืนอยู่ที่ตำแหน่งพิกัดละติจูด ลองจิจูดตามรูป 4.23 จะได้รูปภาพใกล้เคียงรอบตัวผู้ใช้งานดังต่อไปนี้

**รูป 4.24 ชุดภาพในระยะ 50 เมตร จากพิกัด 13.7282473,100.7759738**

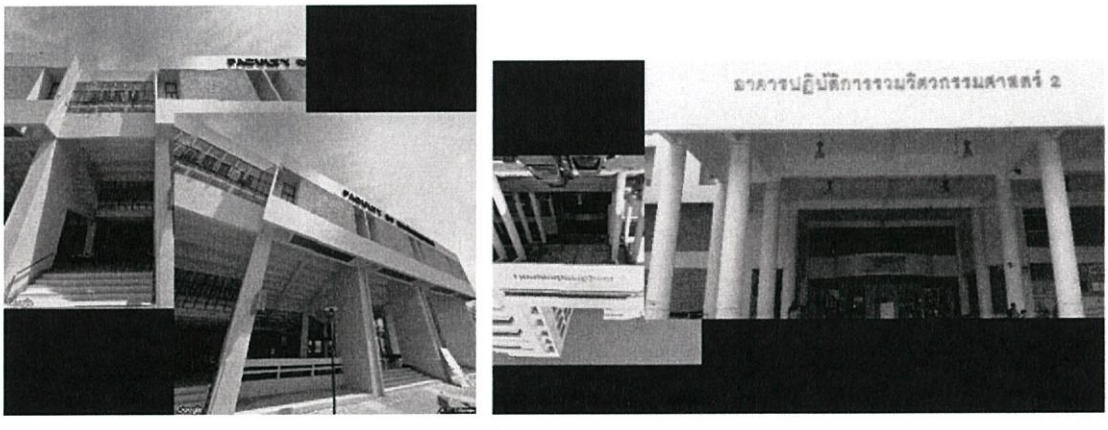
**4.1.6 ส่วนของการเปรียบเทียบภาพถ่าย**

ในขั้นตอนการเปรียบเทียบภาพถ่ายนั้นได้มีการนำเทคนิคมาใช้ด้วยกันทั้งสองเทคนิคเพื่อเปรียบเทียบว่าเทคนิคไหนสามารถใช้งานได้ดีกว่า และมีประสิทธิภาพในการยืนยันว่าภาพที่ผู้ใช้งานส่งเข้ามาเหมือนกับภาพในฐานข้อมูลมากกว่ากัน ซึ่งเทคนิคแรกที่ใช้นั้นคือ “การจัดเรียงภาพย่อยเพื่อประกอบเป็นภาพใหญ่โดยใช้เทคนิคการแปลงข้อมูลแบบฟาสฟูเรียร์ และวิธีการค้นหาแบบเลือกค่าที่ดีที่สุดก่อน” [2] ซึ่งจากการทดลองนั้นพบว่าเทคนิคนี้ไม่เหมาะกับงานชนิดนี้ เนื่องจากภาพอินพุตที่เข้ามานั้นไม่สามารถคาดเดาล่วงหน้าได้เลยว่าผู้ใช้งานจะทำการส่งเข้ามาในมุมมองไหนซึ่งเทคนิคนี้นั้นรองรับแค่ในมุมหมุนธรรมดาเท่านั้น




**รูป 4.25 ผลลัพธ์ของการเปรียบเทียบโดยใช้เทคนิค [2]**

จากรูป 4.25 จะเห็นว่าเทคนิค [2] นี้สามารถให้คำตอบได้ดีเมื่อภาพที่ทำการเปรียบเทียบทั้งสองภาพถูกถ่ายมาจากมุมมองเดียวกัน แต่ถ้าหากเราทำการถ่ายจากคนละมุมมองแล้วจะพบว่าภาพทั้งสองภาพนั้นแทบจะมีความผิดพลาดสูงมาก หรือแม้แต่ถ้ามีการถ่ายภาพอาคารที่มีการซูมเข้าไปในระยะใกล้ ก็จะมีผลที่ผิดพลาดสูงเช่นกัน



ก) ข)

**รูป 4.26 การเปรียบเทียบภาพโดยใช้เทคนิค [4] แล้วค่ามีความผิดพลาด**

ก) การเปรียบเทียบภาพที่มีมุมมองต่างกัน

ข) การเปรียบเทียบภาพที่มีลักษณะการซูม

จากรูป 4.26 ในข้อ ก) จะสังเกตเห็นได้ว่าถ้าภาพทั้งสองภาพนั้นถ้าภาพใดภาพหนึ่งมาลักษณะการถ่ายภาพในคนละมุมมองกันผลลัพธ์ที่ได้จะมีความคลาดเคลื่อนสูงมากเนื่องจากเทคนิค [2] นี้รองรับแค่มุมมองการหมุนแค่ในแกนปกติเท่านั้น และเช่นกันในรูป ข) ถ้าหากว่ามีการซูมเข้าหรือซูมออกจะทำให้รายละเอียด และมุมมองของภาพนั้นแตกต่างออกไป ซึ่งทำให้เทคนิค [2] นั้นทำงานได้ไม่ดีเท่าที่ควร

และอีกเทคนิคที่ใช้ในการเปรียบเทียบภาพถ่ายนั้นได้เลือกใช้อัลกอริทึม (ASIFT Algorithm) [3] โดยค่าของการจับคู่คุณลักษณะของภาพแต่ละคู่นั้นขึ้นอยู่กับพารามิเตอร์แกนหมุนในแนวพิกัดโลก (Tilt) ซึ่งยิ่งเราตั้งค่าให้ทำมีแกนหมุนมากขึ้นเท่าไร จำนวนภาพที่ทำการจำลองเพื่อใช้ในการเปรียบเทียบภาพแต่ละภาพก็จะมากขึ้น และผลของการเปรียบเทียบก็จะมีค่าความถูกต้องมากขึ้น

**ตาราง 4.3 การเปรียบเทียบเมื่อเปลี่ยนพารามิเตอร์ Tilt**

| ค่า Tilt (แกนเอียง) | จำนวนแกนหมุนที่ทำการจำลองเพื่อเปรียบเทียบ (Affine Simulations) |
|---|---|
| 1 | 1 |
| 2 | 5 |
| 3 | 10 |
| 4 | 17 |
| 5 | 27 |
| 6 | 41 |
| 7 | 61 |

จากตาราง 4.3 สามารถสรุปได้ว่า เมื่อเราทำการปรับพารามิเตอร์ค่านี้เท่ากับ 1 การเปรียบเทียบภาพถ่ายนั้นจะใช้ได้ก็ต่อเมื่อ ภาพถ่ายอินพุตจากผู้ใช้งาน และภาพถ่ายจากฐานข้อมูลต้องเป็นรูปเดียวกันเท่านั้น และยิ่งปรับค่ามากเท่าใดค่าความถูกต้องจะยิ่งสูงขึ้นเนื่องจากมีการจำลองภาพด้วยแกนหมุนตามแนวโลกมากขึ้น

จากผลการทดลองเมื่อใช้ภาพอินพุตเป็นภาพอาคารเรียนปฏิบัติการวิศวกรรม 2 ขนาดของภาพอินพุตมีขนาดเป็น 1280x720 เทียบกับชุดภาพที่ ณ พิกัดที่ละจิจูด 13.729323 ลองติจูด 100.7759

**รูป 4.27 อินพุตจากผู้ใช้งาน**

**รูป 4.28 ตัวอย่างที่ใช้ในการเปรียบเทียบจากฐานข้อมูล**

**ตาราง 4.4 จำนวนคุณลักษณะเด่นที่จับคู่กันในการตั้งค่าพารามิเตอร์ Tilt**

| ค่า Tilt | จำนวนคุณลักษณะเด่นที่จับคู่กัน (Match Keypoints) | จำนวนคุณลักษณะเด่นที่ผิดพลาด | ค่าความถูกต้องเฉลี่ย(%) |
|---|---|---|---|
| 1 | 0 | 0 | 0 % |
| 2 | 16 | 3 | 81.25 % |
| 3 | 21 | 4 | 80.95 % |
| 4 | 23 | 4 | 82.60 % |
| 5 | 25 | 4 | 84 % |
| 6 | 26 | 4 | 84.6 % |
| 7 | 27 | 6 | 77.77 % |

จากตาราง 4.4 สามารถสรุปผลการทดลองได้ว่าถ้าหากว่ามีการเปรียบเทียบระหว่างภาพอินพุตจากผู้ใช้งาน และภาพตัวอย่างอาคารจากฐานข้อมูล ที่มีมุมมองในของภาพถ่ายไม่ต่างกันมากนักไม่ว่าจะปรับค่าพารามิเตอร์ Tilt เท่าไรผลลัพธ์ที่ได้มีโอกาสในการจับคู่ที่มีความถูกต้องกันค่อนข้างสูง และมีจำนวนความผิดพลาดที่จับคู่ผิดใกล้เคียงกัน แต่ยังยอมรับได้ในระดับหนึ่ง โดยค่าความถูกต้องเฉลี่ยทั้งหมดนั้นคิดจากจำนวนคุณลักษณะที่ถูกต้องทั้งหมดที่พบในการเปรียบเทียบโดยหักจำนวนที่ผิดพลาด ซึ่งจะเห็นว่าพารามิเตอร์นี้จะไม่ค่อยมีผลเท่าไรนักกับการเปรียบเทียบภาพที่ไม่มีการหมุนในแนวแกนโลก

**รูป 4.29 ผลการเปรียบเทียบระหว่างภาพอินพุตกับตัวอย่างภาพจากอาคารโดย Tilt เท่ากับ 6**

จากรูป 4.29 จะสังเกตเห็นได้ว่าจำนวนคุณลักษณะที่จับคู่กันได้ถูกต้องจะมีน้ำหนักอยู่ตรงที่ตัวอักษรที่ติดอยู่ในอาคาร และส่วนที่ผิดพลาดจะเกิดจากตัวเสาอาคารเนื่องจากว่าเสาอาคารนั้นมีลักษณะใกล้เคียงกัน

**รูป 4.30 ผลการเปรียบเทียบระหว่างภาพอินพุตกับตัวอย่างภาพจากอาคารโดย Tilt เท่ากับ 7**

แต่เนื่องจากเราไม่สามารถคาดการณ์ได้เลยว่าผู้ใช้งานนั้นจะทำการถ่ายภาพตัวอาคารมาจากในมุมมองใด ระบบจึงต้องทำการทดลองพารามิเตอร์ Tilt ที่ให้ผลความถูกต้องในการเปรียบเทียบภาพจากหลายมุม ดังนั้นเราจึงต้องทำการเก็บภาพอาคารในฐานข้อมูลหลายมุมด้วยกัน โดยกำหนดภาพอินพุตเป็นภาพเดิม และภาพตัวอย่างอาคารจากฐานข้อมูลเป็นภาพต่อไปนี้

**รูป 4.31 ตัวอย่างจากฐานข้อมูลในมุมอื่นๆ**

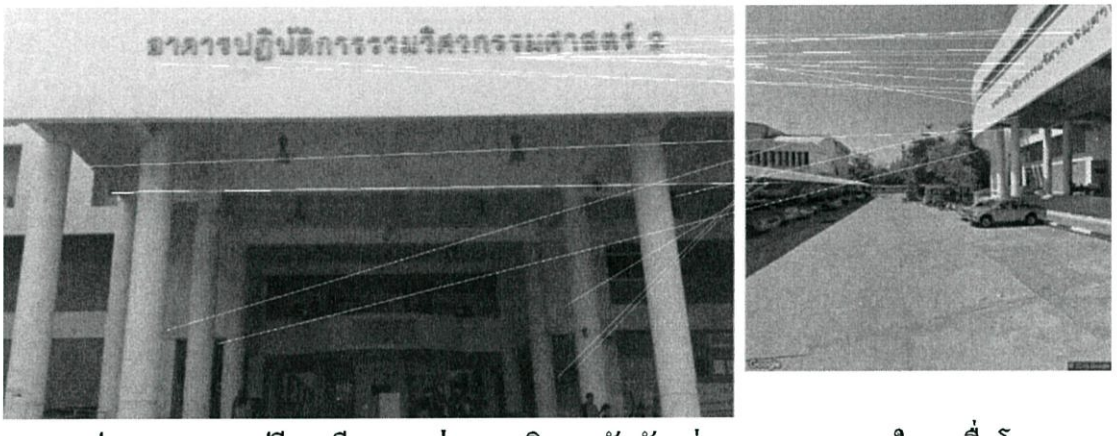
**ตาราง 4.5 จำนวนคุณลักษณะเด่นที่จับคู่กันในการตั้งค่าพารามิเตอร์ Tilt ในจับคู่ภาพมุมมองอื่นๆ**

| ค่า Tilt | จำนวนคุณลักษณะเด่นที่จับคู่กัน (Match Keypoints) | จำนวนคุณลักษณะเด่นที่ผิดพลาด | ค่าความถูกต้องเฉลี่ย (%) |
|---|---|---|---|
| 1 | 0 | 0 | 0 % |
| 2 | 0 | 0 | 0 % |
| 3 | 0 | 0 | 0 % |
| 4 | 16 | 5 | 68.75 % |
| 5 | 19 | 5 | 73.68 % |
| 6 | 21 | 8 | 61.9 % |
| 7 | 22 | 8 | 72.72 % |

จากตาราง 4.5 สามารถสรุปได้ว่าพารามิเตอร์ Tilt นั้นเริ่มมีผลกับการเปรียบเทียบภาพถ่ายอย่างมีนัยยะสำคัญ ซึ่งถ้าหากเราทำการตั้งค่าพารามิเตอร์เท่ากับ 3 หรือน้อยกว่านั้น ผลของการ

เปรียบเทียบจะไม่สามารถจับคู่คุณลักษณะเด่นได้เลย ซึ่งต้องตั้งค่าพารามิเตอร์มากกว่า 3 ขึ้นไปถึงจะเริ่มเห็นถึงการจับคู่คุณลักษณะเด่นของภาพทั้งสองภาพ

**รูป 4.32 ผลการเปรียบเทียบระหว่างภาพอินพุตกับตัวอย่างภาพจากอาคารในมุมอื่นโดย Tilt เท่ากับ 5**



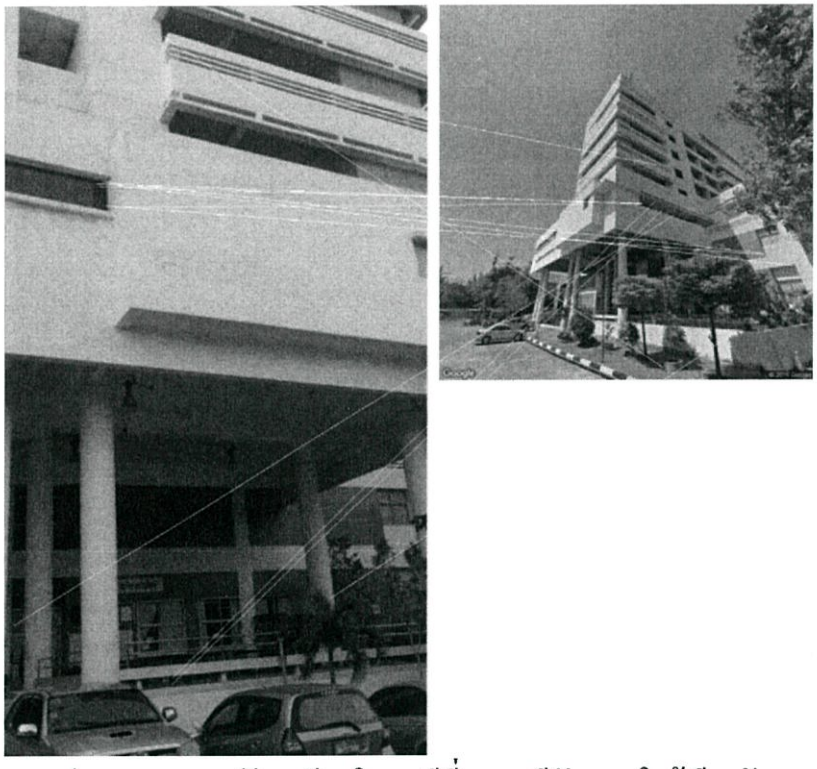
**รูป 4.33 ผลการเปรียบเทียบระหว่างภาพอินพุตกับตัวอย่างภาพจากอาคารในมุมอื่นโดย Tilt เท่ากับ 7**

จากผลการทดลองข้างตั้งทั้งสองการทดลองสรุปได้ว่าในการเปรียบเทียบภาพถ่ายนั้น พารามิเตอร์ Tilt ที่เหมาะสมกับโครงงานนี้คือมากกว่า 6 ซึ่งยังให้ผลความถูกต้องในระดับที่สามารถยอมรับได้ แต่ในโครงงานนี้จะเลือกกำหนดพารามิเตอร์ตัวนี้ให้มีค่าเท่ากับ 6 เนื่องจากสามารถให้ผลของการจับคู่ได้ดีในระดับหนึ่ง และรวมไปถึงเวลาในการตอบสนองที่ค่อนข้างเร็วพอสมควรด้วยเช่นกันซึ่งแสดงในตาราง 4.6 ซึ่งเวลาเฉลี่ยที่ได้มานั้นทดสอบบนเครื่องเซิร์ฟเวอร์ที่ใช้รันเว็บแอพพลิเคชันโดยมีระบบปฏิบัติการเป็น Ubuntu 14.04 มี CPU Cores ทั้งหมด 4 Cores โดยมีการประมวลผลอยู่ที่ 1.9 Ghz/Core Ram 2 Gb และมีหน่วยความจำทั้งหมด 20 GB

**ตาราง 4.6 เวลาเฉลี่ยต่อการเปรียบเทียบภาพถ่าย 1 ครั้ง**

| ค่า Tilt | เวลาที่ใช้เฉลี่ยต่อการเปรียบเทียบหนึ่งครั้ง (วินาที) |
|---|---|
| 1 | 3 |
| 2 | 4 |
| 3 | 8 |
| 4 | 10 |
| 5 | 16 |
| 6 | 24 |
| 7 | 30 |

สำหรับข้อจำกัดของการเปรียบเทียบโดยใช้อัลกอริทึม ASIFT Algorithm[3] นี้คือถ้าหากว่าตัวอาคารต่างกันแต่มีลักษณะเด่นบางส่วนที่ซ้ำซ้อนกัน หรือมีลักษณะรูปแบบที่ใกล้เคียงกัน ผลลัพธ์ในการเปรียบเทียบที่ได้มานั้นจะไม่มีประสิทธิภาพเท่าที่ควร

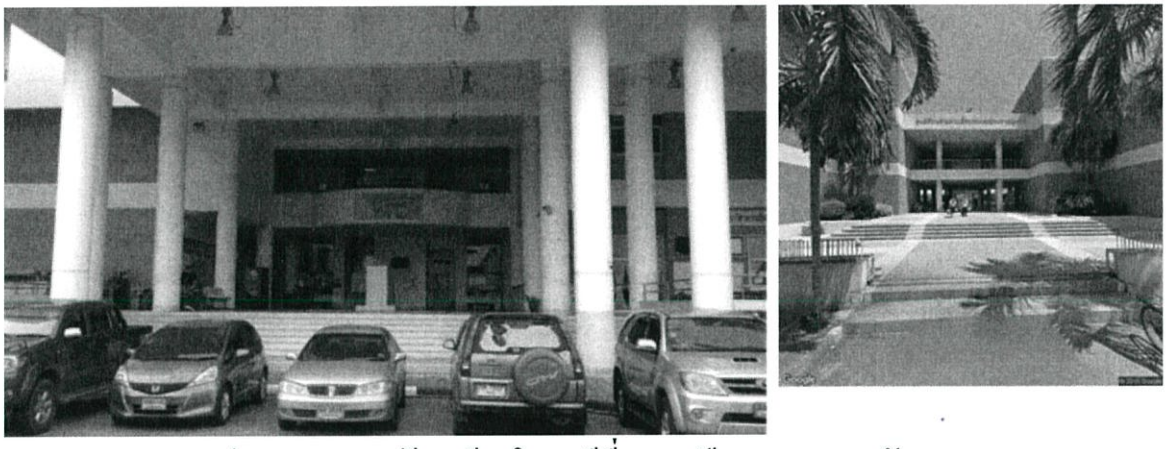


**รูป 4.34 ผลการเปรียบเทียบในกรณีที่อาคารมีลักษณะใกล้เคียงกัน**

จากรูป 4.34 ข้างต้นจะเห็นข้อผิดพลาดได้ว่าด้านซ้ายเป็นภาพอินพุตจากผู้ใช้งานซึ่งถ่ายภาพจากด้านหน้าของอาคารปฏิบัติการรวมวิศวกรรมศาสตร์ 2 แล้วเมื่อเปรียบเทียบกับตัวอย่าง

ภาพจากฐานข้อมูลในมุมมองด้านข้างของตึก น้ำหนักของการจับคู่จะอยู่ตรงที่ขอบกำแพง ซึ่งขอบกำแพงในสองภาพนี้อยู่กันคนละด้านของอาคาร ซึ่งภาพอินพุตอยู่ทางด้านหน้าของอาคาร และภาพจากฐานข้อมูลตัวขอบกำแพงอยู่อีกทางฝั่งหนึ่งทางด้านข้างของตัวอาคาร

แต่ถ้าหากว่าตัวอาคารทั้งสองอาคารนั้นไม่ได้มีความเกี่ยวข้องกันเลยหรือเป็นคนละอาคาร ผลลัพธ์ของการจับคู่ได้ประสิทธิภาพค่อนข้างดี



**รูป 4.35 ผลการเปรียบเทียบในกรณีที่อาคารเป็นคนละอาคารกัน**

จากรูป 4.35 เป็นตัวอย่างการเปรียบเทียบด้านซ้ายเป็นภาพอาคารปฏิบัติการรวมวิศวกรรม 2 ตั้งอยู่ที่พิกัด (13.72920,100.77554) และด้านขวาเป็นภาพอาคารเรียนรวมสมเด็จพระเทพตั้งอยู่ที่พิกัด (13.72999,100.76858) ซึ่งมีระยะห่างกันถึง 180 เมตร เมื่อคำนวณหาระยะห่างทั้งสองอาคารจากสมการ 3.2 โดยผลการเปรียบเทียบจะเห็นว่าทั้งสองอาคารนี้ไม่ได้มีส่วนเกี่ยวข้องกัน

# บทที่ 5
# บทสรุป

## 5.1 ผลที่ได้รับจากโครงงาน

### 5.1.1 ผลของส่วนแอพพลิเคชัน

ในการค้นหาเส้นทางภายในสถาบันนั้น พิกัดปัจจุบันที่ได้รับจากเว็บแอพพลิเคชันนั้นจะมีผลคลาดเคลื่อนน้อยกว่าพิกัดจีพีเอสเพียงอย่างเดียวเพื่อบอกว่า ขณะนี้ผู้ใช้งานอยู่ในบริเวณไหนของสถาบัน และเมื่อทำการร้องขอเส้นทาง ผู้ใช้งานจะได้รับเส้นทางที่สั้นที่สุดจากสถานที่ต้นทางไปยังสถานที่ปลายทาง ไม่ว่าจะเป็นการขับรถยนต์ หรือ การเดิน พร้อมทั้งได้รับสถานที่ รอบตัวผู้ใช้งานเป็นระยะ 50 เมตร และยังสามารถดูวิดีโอแนะนำเส้นทางจากสถานที่ต้นทางไปยังสถานที่ปลายทางได้ ทั้งนี้วิดีโอแนะนำเส้นทางสามารถแสดงผลได้เพียงการนำทางภายในสถาบันไปยังสถานที่ภายในสถาบันเท่านั้น นอกจากนี้ผู้ใช้งานยังเข้ามาดูรายละเอียดต่างของหน่วยงานที่สังกัดอยู่ภายในอาคาร วิธีการเดินทางโดยรถสาธารณะ ได้เช่นกัน

### 5.1.2 ผลของการเก็บข้อมูลด้วยกูเกิลเอพีไอ

ในการเก็บข้อมูลนั้นส่วนใหญ่ข้อมูลที่ทำการเก็บรายละเอียด ไม่ว่าจะเป็น พิกัดของอาคารสถานที่ วิดีโอแนะนำเส้นทาง พบว่าข้อมูลที่ได้มีความแม่นยำพอสมควร แต่ว่าจะมีค่าผิดพลาดอยู่บ้าง หรือ มีข้อมูลในบางจุดมีความละเอียดไม่พอ ซึ่งทำให้คณะผู้จัดทำต้องทำการเก็บข้อมูล ด้วยตนเองเพื่อแก้ข้อผิดพลาดในจุดนั้น

### 5.1.3 ผลของการแนะนำเส้นทาง

ในการแนะนำเส้นทางนั้น เส้นทางที่ได้พบว่ามีความใกล้เคียงกับเส้นทางที่ทำการร้องขอจากกูเกิลเอพีไอ แต่จะได้เส้นทางที่ดีกว่าในบางเส้นทาง เกิดจากการที่ในเว็บแอพพลิเคชันนั้นมีการเก็บข้อมูลพิกัดบางพิกัดที่ทางกูเกิลเอพีไอไม่ได้ทำการเก็บข้อมูลไว้ซึ่งทำให้เส้นทางที่ได้จากเว็บแอพพลิเคชั่นมีระยะทางที่ดีกว่า สั้นกว่าตัวกูเกิลเอพีไอ

### 5.1.4 ผลของการเปรียบเทียบภาพถ่าย

ในการเปรียบเทียบภาพถ่ายนั้นจะทำการเปรียบเทียบภาพอินพุตจากผู้ใช้งานกับภาพอาคารที่เซิร์ฟเวอร์ทำการดึงมาจากฐานข้อมูลรอบ ตัวผู้ใช้งานเป็นระยะ 50 เมตร ซึ่งสามารถสรุปได้ว่าผลของการเปรียบเทียบนั้นจะมีประสิทธิภาพมากที่สุดเมื่อผู้ใช้งานทำการถ่ายภาพที่มีคุณภาพดีต่อการเปรียบเทียบ หมายความว่าตัวภาพต้องประกอบด้วยตัวอาคาร มีแสงที่พอดี ไม่มืดจนมองไม่เห็นตัวอาคาร

## 5.2 ปัญหาและอุปสรรค

เนื่องจากโครงงานมีส่วนเสริมขึ้นมาคือฟังก์ชันการประมวลผลภาพซึ่งในตัวอัลกอริทึมนี้เองมีความซับซ้อนมาก และเวลาที่ใช้ในการเปรียบเทียบนั้นค่อนข้างใช้เวลาพอสมควรกว่าจะได้ผลลัพธ์ ซึ่งปัญหาและอุปสรรคที่ได้พบจะถูกแบ่งเป็นส่วนดังนี้

### 5.2.1 ปัญหาและอุปสรรคในส่วนแอพพลิเคชัน

ปัญหาและอุปสรรคส่วนใหญ่เกิดจากการย้ายระบบจากเครื่องที่ใช้พัฒนาทั้งหมดไปยังเครื่องเซิร์ฟเวอร์ซึ่งต้องทำการปรับปรุงในหลายส่วนโดยส่วนมากจะเป็นส่วนที่ติดต่อกับเอพีไอทั้งหมด ทั้งเอพีไอที่ติดต่อกับกูเกิล ยูทูป ซึ่งต้องทำการแก้ไขเพื่อให้ระบบใช้ได้ รวมไปถึงการประมวลผลที่ค่อนข้างนาน เนื่องจากทรัพยากรเครื่องเซิร์ฟเวอร์มีข้อจำกัด ในการยืนยันผลลัพธ์พิกัดที่ถูกต้องเพื่อยืนยันตำแหน่งให้แก่ผู้ใช้งานโดยการเปรียบเทียบภาพถ่าย

### 5.2.2 ปัญหาและอุปสรรคของการเก็บข้อมูล

ปัญหาและอุปสรรคในการเก็บข้อมูลนั้นไม่ว่าจะเป็นพิกัดที่ใช้นำทาง ภาพถ่ายวิดีโอที่นำมาใช้ทำเส้นทาง ในขั้นตอนการทำนั้นทางคณะผู้จัดทำใช้อัลกอริทึมในการเก็บข้อมูล ซึ่งผลลัพธ์ของข้อมูลที่ได้จะมีบางจุดที่เกิดความผิดพลาด อย่างเช่นผลลัพธ์ของภาพที่นำมาใช้ทำวิดีโอบางจุดจะได้ภาพที่กระโดดเลนมาบ้าง หรือบางจุดที่ทำการร้องขอไปไม่ได้มีการเก็บข้อมูลโดยเอพีไว้ จึงต้องทำการลงไปเก็บข้อมูลเอง

### 5.2.3 ปัญหาและอุปสรรคในการแนะนำเส้นทาง

เส้นทางที่ได้จากระบบนั้นเป็นเส้นทางที่สั้นที่สุดก็จริง แต่ถ้าหากว่าในเวลาใดเวลาหนึ่ง ณ ขณะนั้นมีบางเส้นทางที่ทำการปรับปรุง หรือ ปิดให้บริการ จะทำให้เส้นทางที่ทางผู้ใช้งานทำการร้องขอมานั้นไม่มีประสิทธิภาพพอที่จะนำผู้ใช้งานไปยังสถานที่ปลายทาง เนื่องจากตัวอัลกอริทึมที่ใช้ในการแนะนำเส้นทางนั้น จะได้เส้นทางที่สั้นที่สุดจากสถานที่ต้นทาง ไปยังสถานที่ปลายทางเสมอ

### 5.2.4 ปัญหาและอุปสรรคในการเปรียบเทียบภาพถ่าย

1) ถ้าในฐานข้อมูลไม่มีภาพถ่ายที่มีลักษณะ หรือ มีมุมในการถ่ายภาพคล้ายคลึงกับภาพที่ผู้ใช้งาน ผู้ใช้งานจะไม่สามารถรับพิกัดที่ถูกต้องในขณะนั้นได้
2) ถ้าอาคารในภาพถ่ายนั้นมีลักษณะที่มีความซ้ำซ้อน และเกิดไปตรงกับภาพถ่ายที่อยู่ในฐานข้อมูลในลักษณะคล้ายคลึงกัน หรือผู้ใช้งานนำเข้าภาพถ่ายที่มีคุณลักษณะโดดเด่นเป็นประติมากรรม รูปปั้น ที่อยู่บริเวณอาคารนั้น แทนที่จะเป็นภาพถ่ายของอาคาร อาจจะทำให้ผลลัพธ์ในการยืนยันตำแหน่งมีความคลาดเคลื่อนได้
3) ถ้าหากบริเวณที่ผู้ใช้งานทำการร้องขอยืนยันตำแหน่งพิกัดปัจจุบันนั้น มีอาคารอยู่หลายอาคารด้วยกัน จะทำให้เวลาในการประมวลผลนาน เนื่องจากมีชุดภาพถ่ายของอาคารแต่ละอาคารนั้นอยู่เยอะพอสมควร ในการเปรียบเทียบ

4) ถ้าหากภาพถ่ายที่ผู้ใช้งานนำเข้ามามีคุณภาพไม่ดีพอต่อการเปรียบเทียบ กล่าวคือ ภาพถ่ายย้อนแสง ถ่ายภาพอาคารตอนกลางคืน อาจจะทำให้ไม่ได้ผลลัพธ์ตามที่ต้องการ
5) เนื่องจากตัวอัลกอริทึมที่นำเข้ามาใช้เป็น Open Source จึงต้องทำให้ใช้เวลาในการศึกษา ติดตั้ง และแก้ไขค่อนข้างนานกว่าที่ควร
6) เนื่องจากตัวอุปกรณ์ของผู้ใช้งานในการถ่ายภาพนั้นมีความหลากลายซึ่งเวลารับภาพมาต้องทำการปรับลดขนาดของภาพให้มีความเหมาะสมกับอัลกอริทึมจึงอาจจะทำให้รายละเอียดของภาพที่รับเข้ามานั้นลดลงไปบ้าง

## 5.3 แนวทางในการแก้ไข

### 5.3.1 แนวทางในการแก้ไขปัญหาในส่วนแอพพลิเคชัน

สำหรับการแก้ไขในส่วนนี้ทำการแก้ไขโดยทำการแมพลิ้งค์ของเซิร์ฟเวอร์เป็นโดเมนเนมของทางภาควิชา เพื่อให้รองรับต่อการใช้เอพีไอต่างได้อย่างถูกต้อง

### 5.3.2 แนวทางแก้ไขในการเก็บข้อมูล

สำหรับข้อมูลบางพิกัด หรือภาพถ่ายบางภาพถ่าย ที่มีความผิดพลาดทางเทคนิค ทางคณะผู้จัดทำได้ทำการแก้ไขปัญหานี้โดยลงพื้นที่ไปเก็บข้อมูลจริง เพื่อนำไปใช้ในแอพพลิเคชันต่อไป

### 5.3.3 แนวทางแก้ไขในการแนะนำเส้นทาง

สำหรับแนวทางการแก้ไขพิกัดที่ใช้แนะนำเส้นทางนั้น ทางคณะผู้จัดทำได้เปิดให้สามารถทำการแก้ไขพิกัดที่มีการปิดให้บริการในช่วงเวลาใดเวลาหนึ่งในขณะนั้นได้ เพื่อให้อัลกอริทึมที่จัดทำขึ้นมาได้ทำการค้นหาเส้นทาง และนำทางไปยังสถานที่ปลายทางได้อย่างมีประสิทธิภาพมากขึ้น

### 5.3.4 แนวทางในการแก้ไขในการเปรียบเทียบภาพถ่าย

ควรจะมีการจัดลำดับขั้นตอนการทำงานให้ดีขึ้น เพิ่มพิกัดลงในภาพถ่ายอาคารในแต่ละภาพ เพื่อลดเวลาในการเปรียบเทียบภาพถ่ายที่อยู่คนละมุมจากอาคารนั้นให้มีระยะเวลาในการประมวลผลน้อยลงยิ่งขึ้น และทำการศึกษาทฤษฎีของอัลกอริทึม (ASIFT Algorithm [3]) ที่ใช้ให้ละเอียดมากขึ้น เพื่อให้ประสิทธิภาพในการเปรียบเทียบแต่ละครั้งมีความถูกต้องมากที่สุด

# เอกสารอ้างอิง


[1] ผศ.ดร.อรฉัตร จิตต์โสภักตร์. **ทฤษฎีการประมวลผลภาพดิจิตอล.** กรุงเทพมหานคร: บริษัท สงวนกิจพริ้นท์ แอนด์ มีเดีย. 2552.

[2] โอพาร สัมฤทธิ์เจียรผล.**"การจัดเรียงภาพย่อยเพื่อประกอบเป็นภาพใหญ่โดยใช้เทคนิคการแปลงข้อมูลแบบฟาสฟูเรียร์ และวิธีการค้นหาแบบเลือกค่าที่ดีที่สุดก่อน"**วิทยานิพนธ์วิศวกรรมศาสตร์มหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2552.

[3] Guoshen Yu, Jean-Michel Morel. **"ASIFT:An algorithm for Fully Affine Invariant Comparison"** [Online]. Available: http://www.ipol.im/pub/art/2011/my-asift/

[4] Wikipedia. **"A* Search Algorithm"** [Online]. Available: https://en.wikipedia.org/wiki/A*_search/algorithm

[5] Google.**"Google Maps Javascript API V3"** [Online]. Available: https://developers.google.com/maps/documentation/javascript/reference

[6] Google, **"Youtube API"** [Online]. Available: https://developers.google.com/youtube/

[7] Thaitracking. "**GPS คืออะไร" [Online].** Available: https://www.thaitracking.net/articles.php?article_id=10000

[8] Wikipedia. **"Scale-Invariant feature transform" [**Online]. Available: https://en.wikipedia.org/wiki/Scale- invariant_feature_transform