

ระบบหาเส้นทางที่ดีที่สุดโดยใช้ชุดข้อมูลเส้นทางที่เคยผ่าน
PATH ESTIMATION SERVICE USING PATH-VISITED DATASET

ศิริตรา นามสุข

สหกิจศึกษาเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

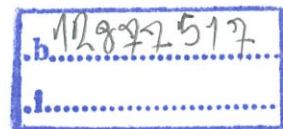
ระบบหาเส้นทางที่ดีที่สุดโดยใช้ชุดข้อมูลเส้นทางที่เคยผ่าน
PATH ESTIMATION SERVICE USING PATH-VISITED
DATASET



T148959

ศิริตรา นามสุข

เลขหมู่.....
เลขทะเบียน..... 148959
วัน,เดือน,ปี..... 1 ๘ S.ค. 2560



สหกิจศึกษาเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

PATH ESTIMATION SERVICE USING PATH-VISITED
DATASET

SIRITA NAMSOOK

COOPERATIVE EDUCATION SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF
SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2015

หัวข้อโครงการพิเศษ ระบบหาเส้นทางที่ดีที่สุดโดยใช้ชุดข้อมูลเส้นทางที่เคยผ่าน
 Path Estimation Service using Path-visited dataset

ชื่อนักศึกษา นางสาวศิริตรา นามสุข รหัสนักศึกษา 55050475

ปริญญา วิทยาศาสตรบัณฑิต


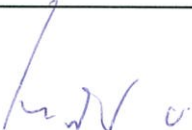
สาขาวิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2558

อาจารย์ที่ปรึกษา ผศ.กฤษฎา บุศรา

อาจารย์ที่ปรึกษา ดร.ไพรัตน์ ธรเจริญศรี

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้
 สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิชา
 วิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2558

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.กฤษฎา บุศรา ประธานกรรมการและอาจารย์ที่ปรึกษา	
ดร.ไพรัตน์ ธรเจริญศรี กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อโครงการพิเศษ	ระบบหาเส้นทางที่ดีที่สุดโดยใช้ชุดข้อมูลเส้นทางที่เคยผ่าน Path Estimation Service using Path-visited dataset
ชื่อนักศึกษา	นางสาวศิริตรา นามสุข 55050475
ปริญญา	วิทยาศาสตร์บัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา	2558
อาจารย์ที่ปรึกษา	ผศ.กฤษฎา บุศรา
อาจารย์ที่ปรึกษา	ดร.ไพรัตน์ ธรเจริญศรี

บทคัดย่อ

ในปัจจุบันเครื่องมือที่เกี่ยวข้องกับภูมิศาสตร์และการจราจรเพื่อใช้ในการคำนวณหาเส้นทางที่ดีที่สุดมีหลากหลายรูปแบบที่คอยอำนวยความสะดวกต่อผู้ที่ต้องการใช้งาน แต่การสร้างเครื่องมือที่มีประสิทธิภาพนั้นต้องคำนึงถึงปัจจัยต่างๆไม่ว่าจะเป็นเวลา น้ำมันหรือความหนาแน่นของการจราจร เป็นต้น ในงานวิจัยนี้ได้นำชุดข้อมูลความนิยมในการใช้เส้นทางมาเป็นปัจจัยในการหาเส้นทางที่ดีที่สุด ซึ่งเส้นทางใดที่มีความนิยมสูงสุดจะถูกเลือกให้เป็นเส้นทางที่ดีที่สุด โดยชุดข้อมูลความนิยมในการใช้เส้นทางจะถูกเก็บจากรถเมล์ รถตู้และรถแท็กซี่ เครื่องมือที่ได้ทำการพัฒนานั้นจะนำมาใช้งานบนเครือข่าย ซึ่งผู้ที่ต้องการใช้งานสามารถร้องขอใช้งานได้ผ่านระบบเครือข่ายคอมพิวเตอร์

คำสำคัญ: ชุดข้อมูลความนิยมในการใช้เส้นทาง, คำนวณหาเส้นทางที่ดีที่สุด

Title	Path Estimation Service using Path-visited dataset
Student	Sirita Namsook 55050475
Degree	Bachelor of Science
Major	Computer Science
Academic Year	2015
Advisor	Asst.Prof.Krudsada Budsara
Co-advisor	Dr.Pirat Thorncharoensri

ABSTRACT

The present has many tool about geography and traffic for the estimation path, but implement tool to high quality consider factors the cost of oil time or traffic etc. Path-visited dataset was used for estimation path, Which the best path are the most path of popularity path. Path-visited dataset derived from bus van and taxi. Tool of implement use on application programming interface (API).

keyword: estimation path,the most path of popularity path

กิตติกรรมประกาศ

ระบบหาเส้นทางที่ดีที่สุดโดยใช้ชุดข้อมูลเส้นทางที่เคยผ่าน ผู้จัดทำขอขอบพระคุณ ดร. วสันต์ ภัทรอธิคม หัวหน้าห้องปฏิบัติการงานวิจัยระบบขนส่งและจราจรอัจฉริยะ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ ที่ได้สละเวลาในการแนะนำ ชี้แนะให้เห็นถึงปัญหาและเสนอแนวทางในการแก้ไขปัญหาที่ได้พบเจอ

นายณรงค์ อินทธีริภักซ์ ผู้คอยดูแลและช่วยเหลืองานวิจัยนี้ในทุกๆด้าน การแนะนำแนวทางชี้แจงถึงปัญหาต่างๆ การวางแผนคอยควบคุมการทำงานและให้คำปรึกษาถึงปัญหาที่พบเจอในงานวิจัย นายณพงศ์ วาณิชยพงศ์ ผู้คอยประสานงานและให้คำปรึกษาในด้านต่างๆทั้งเกี่ยวกับงานและในทุกๆด้านที่พร้อมจะให้ความช่วยเหลือเมื่อพบเจอปัญหา นายมนศักดิ์ โช้เจริญธรรม ผู้คอยชี้แจงปัญหาที่มีในงานวิจัยและให้คำแนะนำเกี่ยวกับแนวทางที่สามารถช่วยแก้ปัญหาในงานวิจัยนี้ นายวัชร พึ่งวัชรการ ผู้คอยให้คำปรึกษาและช่วยเหลือในงานวิจัย ช่วยสอนและแนะนำการปฏิบัติงานในส่วนสำคัญของงานวิจัย นางสุภรณ์ เหมือนหนู ผู้ให้ความเอ็นดูและคอยชี้แนะการทำงานและให้คำปรึกษาในด้านต่างๆทั้งงานวิจัยและเรื่องการระเบียบวินัยในการทำงานและด้านอื่นๆอีกมากมาย รวมถึงผู้ที่ให้คอยให้ความช่วยเหลือท่านอื่นๆที่ไม่ได้กล่าวไว้ ณ ที่นี้

ขอขอบคุณ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ ที่ได้ตอบรับคำขอเข้าฝึกสหกิจ อีกทั้งสาขาวิทยาการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สุดท้ายนี้ขอขอบคุณเพื่อนร่วมงานทั้งจากสถาบันเดียวกันและเพื่อนร่วมงานต่างสถาบันที่คอยเป็นกำลังใจและให้คำปรึกษาในทุกๆด้านและคุณพ่อและคุณแม่ที่เป็นกำลังใจสำคัญและสนับสนุนการศึกษา

ศิริตรา นามสุข

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	2
1.3 ขอบเขตของงานวิจัย.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 ขั้นตอนในการดำเนินงาน.....	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	3
2.1 ส่วนการแสดงผล (Display).....	3
2.2 ส่วนการประมวลผล (Processing).....	4
2.2.1 pgRouting.....	4
2.2.2 QGIS หรือ Quantum GIS.....	4
2.3 ส่วนของฐานข้อมูล (Database).....	5
2.3.1 postgresql.....	5
2.3.2 filezilla.....	6
2.3.3 putty.....	6
2.4 ฟังก์ชัน ไดจ์สตรา (Dijkstra's algorithm)	7
บทที่ 3 วิธีการดำเนินงานวิจัย.....	8
3.1 ทดสอบประสิทธิภาพฟังก์ชัน Shortest Path Dijkstra.....	8
3.1.1 การติดตั้ง pgrouting.....	8

สารบัญ(ต่อ)

เรื่อง	หน้า
3.1.2 การจัดเตรียมข้อมูลบนฐานข้อมูล.....	9
3.1.3 การคำนวณฟังก์ชัน Shortest Path Dijkstra.....	9
1) เพิ่มคอลัมน์ข้อมูล	9
2) ใช้คำสั่งสร้าง Topology.....	9
3) CREATE INDEX.....	9
4) เพิ่มตัวแปร cost และ reverse_cost.....	10
3.1.4 ทดสอบประสิทธิภาพฟังก์ชัน Shortest Path Dijkstra.....	10
1) Google Map.....	10
2) การเปรียบเทียบเส้นทางบนรถที่ติดตั้ง GPS.....	14
3.2 การพัฒนาปรับปรุงฟังก์ชัน Shortest Path Dijkstra.....	17
3.2.1 เขียนโปรแกรมหาค่า count.....	18
3.2.2 ปรับปรุงฐานข้อมูล.....	18
3.2.3 การทำ Multiple Path.....	20
3.2.4 การเลือก Path ที่ได้มาจากการทำ Multiple Path.....	21
บทที่ 4 ผลการวิจัยและการอภิปรายผล.....	22
4.1 ประสิทธิภาพที่วัดได้ของเส้นทางแต่ละประเภทและตัวอย่างเส้นทาง.....	22
4.2 สรุปสาเหตุที่ประเภทของเส้นทางมีเปอร์เซ็นต์ของ Routing ไม่ตรงกัน.....	33
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	34
5.1 สรุปผลการวิจัย.....	34
5.2 ข้อเสนอแนะ.....	34
รายการอ้างอิง.....	35
ภาคผนวก.....	36

สารบัญตาราง

ตารางที่	หน้า
3.1 คำสั่ง SQL ที่ใช้ในการติดตั้งฟังก์ชันในการใช้งาน.....	8
3.2 คำสั่ง SQL เพิ่มคอลัมน์ของ source และ target.....	9
3.3 คำสั่ง SQL ที่ใช้คำสั่งเพื่อคำนวณฟังก์ชันสร้าง Topology.....	9
3.4 คำสั่ง SQL ทำการ CREATE INDEX.....	9
3.5 คำสั่ง SQL เพื่อเพิ่มคอลัมน์และกำหนดค่าของ cost และ reverse_cost.....	10
3.6 คำสั่ง SQL ที่ใช้ในการ Query ข้อมูลของเส้นทาง.....	11
3.7 ตัวอย่างคำสั่งที่ใช้ Query ข้อมูลของรถที่ติดตั้ง GPS.....	14
3.8 คำสั่ง SQL ที่ใช้ในการ Query ข้อมูลของเส้นทาง.....	15
3.9 ส่วนประกอบของฐานข้อมูล frequently.....	18
3.10 คำสั่ง SQL ที่ใช้คำสั่งเพื่อคำนวณฟังก์ชันสร้าง Topology.....	20
4.1 Accuracy ที่ทำการ Test Routing ระหว่าง Google Maps กับข้อมูลแผนที่บน Databas.....	22
4.2 Accuracy ที่ทำการ Test Routing ระหว่าง เส้นทางเดินรถกับข้อมูลแผนที่บน Database.....	35

สารบัญรูปภาพ

รูปที่	หน้า
2.1 กูเกิล โครม.....	3
2.2 หน้าต่างโปรแกรม QGIS หรือ Quantum GIS.....	4
2.3 หน้าต่างโปรแกรม postgresSql.....	5
2.4 หน้าต่างโปรแกรม filezilla.....	6
2.5 หน้าต่างโปรแกรม putty.....	6
3.1 ฟังก์ชันที่มีเมื่อทำการติดตั้งแล้ว.....	8
3.2 ข้อมูลที่ได้จากการ Query ของจุดเริ่มและจุดสิ้นสุดของเส้นทาง.....	11
3.3 การนำข้อมูลเส้นทางมาเปิดดูบนโปรแกรม QGIS.....	12
3.4 เส้นทางที่ได้จากการ Query แสดงบนโปรแกรม QGIS.....	12
3.5 การนำค่า edge ไปหาค่า lat_start lng_start lat_end และ lng_end.....	13
3.6 การนำค่า lat_start lng_start lat_end และ lng_end ไปค้นหาเส้นทางบน google map...	13
3.7 เส้นทางที่ได้จากการ Query ช่วงเวลาของรถที่ติดตั้ง GPS.....	14
3.8 ข้อมูลที่ได้จากการ Query ของจุดเริ่มและจุดสิ้นสุดของเส้นทาง.....	15
3.9 การนำข้อมูลเส้นทางมาเปิดดูบนโปรแกรม QGIS.....	16
3.10 เส้นทางที่ได้จากการ Query แสดงบนโปรแกรม QGIS.....	16
3.11 GPS ที่ยังไม่ผ่านการ mapmatch.....	17
3.12 GPS ที่ยังผ่านการ mapmatch แล้ว.....	17
3.13 flow chart แสดงการดึงข้อมูลของค่า count.....	18
3.14 ฐานข้อมูลของ frequently.....	19
3.15 ขนาดความถี่ของจุด GPS ตามเส้นทางที่มีการเดินรถ.....	19
3.16 flow chart การทำ multiple path.....	20
3.17 ตัวอย่าง Multiple path.....	21
3.18 ตัวอย่างเส้นทางที่ดีที่สุดจาก Multiple path.....	21
4.1 เส้นทางบน Google Maps ของประเภททางโค้ง.....	23
4.2 เส้นทางบน Database ของประเภททางโค้ง.....	23
4.3 เส้นทางบน Google Maps ของประเภททางตรง.....	24
4.4 เส้นทางบน Database ของประเภททางตรง.....	24
4.5 เส้นทางบน Google Maps ของประเภททางเลียว.....	25

สารบัญรูปภาพ (ต่อ)

4.6	เส้นทางบน Database ของประเภททางเลี้ยว.....	25
4.7	เส้นทางบน Google Maps ของประเภททางระยะยาว.....	26
4.8	เส้นทางบน Database ของประเภททางระยะยาว.....	26
4.9	เส้นทางบน Google Maps ของประเภททางยูเทิร์น.....	27
4.10	เส้นทางบน Database ของประเภททางยูเทิร์น.....	27
4.11	ตัวอย่างเส้นทางจากรถที่ติดตั้ง GPS ของประเภททางวงเวียน.....	29
4.12	ตัวอย่างเส้นทางบน Database ของประเภททางวงเวียน.....	29
4.13	ตัวอย่างเส้นทางจากรถที่ติดตั้ง GPS ของประเภททางเลี้ยว.....	30
4.14	ตัวอย่างเส้นทางบน Database ของประเภททางเลี้ยว.....	30
4.15	ตัวอย่างเส้นทางจากรถที่ติดตั้ง GPS ของประเภททางแยก.....	31
4.16	ตัวอย่างเส้นทางบน Database ของประเภททางแยก.....	31
4.17	ตัวอย่างเส้นทางจากรถที่ติดตั้ง GPS ของประเภททางตรง.....	32
4.18	ตัวอย่างเส้นทางบน Database ของประเภททางตรง.....	32
4.19	ตัวอย่างเส้นทางจากรถที่ติดตั้ง GPS ของประเภททางโค้ง.....	33
4.20	ตัวอย่างเส้นทางบน Database ของประเภททางโค้ง.....	33

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

Traffy @ITS Lab เป็นหน่วยวิจัยที่ใช้เทคโนโลยีสารสนเทศด้านการเดินทาง ขนส่ง และจราจร เพื่อผลิตผลงานที่มีประโยชน์ต่อประเทศและมีคุณค่าทางวิชาการ ซึ่งอยู่ภายใต้สังกัดของศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (nectec) เป็นหน่วยงานภายใต้สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ กระทรวงวิทยาศาสตร์และเทคโนโลยี ปัจจุบันมีฟังก์ชันมากมายหลากหลายรูปแบบที่ใช้ในการหาเส้นทางที่ดีที่สุดเพื่ออำนวยความสะดวกในด้านการจราจร เช่น Dijkstra's algorithm, Bellman-Ford Algorithm, A* algorithm, Floyd-Warshall algorithm และ Donald B. Johnson algorithm เป็นต้น ซึ่งแต่ละฟังก์ชันมีขั้นตอนในการทำงานที่แตกต่างกัน

ฟังก์ชัน Dijkstra's algorithm เป็นอีกฟังก์ชันที่สามารถนำมาประยุกต์ใช้ในการหาเส้นทางที่สั้นที่สุดได้ซึ่ง pgrouting เป็นเครื่องมือที่จัดหาฟังก์ชันต่างๆสำหรับสร้างเครื่องมือในการคำนวณหาระยะทาง Dijkstra's algorithm เป็นหนึ่งในฟังก์ชันที่ pgrouting จัดหามาให้ใช้งานซึ่ง pgrouting จะนำฟังก์ชัน Dijkstra's algorithm มาปรับให้ตรงตามความต้องการของผู้ใช้ในปัจจุบัน แต่ด้วยประสิทธิภาพของเส้นทางที่ฟังก์ชันคำนวณที่ได้มานั้นยังไม่ให้ผลลัพธ์ที่น่าพอใจ เมื่อนำเส้นทางที่คำนวณด้วยฟังก์ชัน Dijkstra's algorithm ไปเปรียบเทียบกับเครื่องมืออื่น ๆ ที่มีมาตรฐานจะได้เส้นทางที่แตกต่างกันและผลลัพธ์ที่ได้ยังไม่ตรงตามความต้องการในการใช้งาน จึงต้องมีการนำมาประยุกต์และปรับปรุงให้ตรงตามความต้องการให้สอดคล้องกับฐานข้อมูลที่มีอยู่

สหกิจศึกษานี้จึงจัดทำขึ้นเพื่อพัฒนาฟังก์ชัน Dijkstra's algorithm ที่ pgrouting จัดหาฟังก์ชันสำหรับสร้างเครื่องมือในการคำนวณระยะทาง ซึ่งการนำมาประยุกต์ใช้นั้นจะนำชุดข้อมูลของความนิยมในการใช้เส้นทางมาเป็นตัวแปรในการตัดสินใจเพื่อหาเส้นทางที่ดีที่สุด

1.2 วัตถุประสงค์ของสหกิจศึกษา

- 1) เพื่อทดสอบประสิทธิภาพของฟังก์ชัน Shortest path dijkstra ว่ามีประสิทธิภาพเท่าใดเมื่อเทียบกับเครื่องมือที่มีประสิทธิภาพสูงกว่า
- 2) เพื่อแก้ไขปรับปรุงฟังก์ชัน Shortest path dijkstra ให้มีประสิทธิภาพตรงตามการใช้งาน
- 3) เพื่อนำแนวทางที่ทำการศึกษาแก้ไขปรับปรุงมาเขียนเป็นโปรแกรมให้ผู้ใช้ได้นำไปใช้งาน
- 4) เพื่อนำชุดข้อมูลของเส้นทางที่นิยมใช้สูงสุดมาประยุกต์ใช้กับเครื่องมือคำนวณหาเส้นทาง

1.3 ขอบเขตของสหกิจศึกษา

- 1) สหกิจศึกษานี้จะครอบคลุมเส้นทางเฉพาะในประเทศไทย
- 2) สหกิจศึกษานี้จะใช้ค่าตัวแปรตามที่มีบนฐานข้อมูล
- 3) ตัวแปรที่นำมาใช้ในการปรับปรุงฟังก์ชันเป็นตัวแปรที่ถูกสำรวจเส้นทางเฉพาะกรุงเทพมหานคร

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ฟังก์ชันที่ทำการพัฒนาปรับปรุงมีประสิทธิภาพตรงตามที่ต้องการ ผู้ใช้สามารถนำตัวระบบไปประยุกต์ใช้ในงานเกี่ยวกับการคำนวณระยะทาง
- 2) ระบบนี้ช่วยเพิ่มมูลค่าของข้อมูลจีพีเอสที่เก็บไว้ ซึ่งเป็นประโยชน์ต่อผู้ให้บริการเก็บข้อมูลจีพีเอสและบริษัทลอจิสติกส์

1.5 ขั้นตอนในการดำเนินงาน

- 1) ทดสอบประสิทธิภาพฟังก์ชัน Shortest Path Dijkstra
- 2) การปรับปรุงพัฒนาฟังก์ชัน Shortest Path Dijkstra
- 3) นำแนวทางในการปรับปรุงไปเขียนโปรแกรมเพื่อสะดวกต่อการใช้งาน

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

Three-tier architecture

เทียร์ (TIER) คือโปรเซส (Process) ที่ใช้ในการจัดการงานใดงานหนึ่ง ดังนั้นในระบบหนึ่งจะมีเทียร์อย่างน้อยที่สุด 1 เทียร์ แต่จะมีจำนวนเท่าใดนั้นขึ้นอยู่กับทรัพยากรของแต่ละเครื่องและลักษณะงานที่ทำ แบ่งการทำงานเป็น 3 ส่วน

2.1 ส่วนการแสดงผล (Display)

กูเกิล โครม (Google Chrome) เป็นซอฟต์แวร์เว็บเบราว์เซอร์แบบโอเพนซอร์ซ พัฒนาโดยกูเกิล ใช้เว็บคิดเป็นเรนเดอร์ริงเอนจินสำหรับวาดหน้าจอ และนำบางส่วนจากเว็บเบราว์เซอร์ไฟร์ฟอกซ์ มาพัฒนาต่อโครมมีให้ดาวน์โหลดเพื่อทดสอบใช้งานสำหรับวินโดวส์ และมีภาษาที่ให้ใช้ได้มากกว่า 50 ภาษา รวมถึงภาษาไทย รุ่นสำหรับ แมคโอเอส และ ลินุกซ์ นั้นกำลังอยู่ระหว่างการพัฒนา ซึ่งจะเปิดให้ทดสอบในอนาคต โครมเป็นตัวพัฒนาจากโค้ดของซอฟต์แวร์ โครเมียม ซึ่งใช้สัญลักษณ์เดียวกัน แต่คนละสี สัญลักษณ์กูเกิล โครม ดังรูปที่ 2.1



รูปที่ 2.1 กูเกิล โครม

2.2 ส่วนการประมวลผล (Processing)

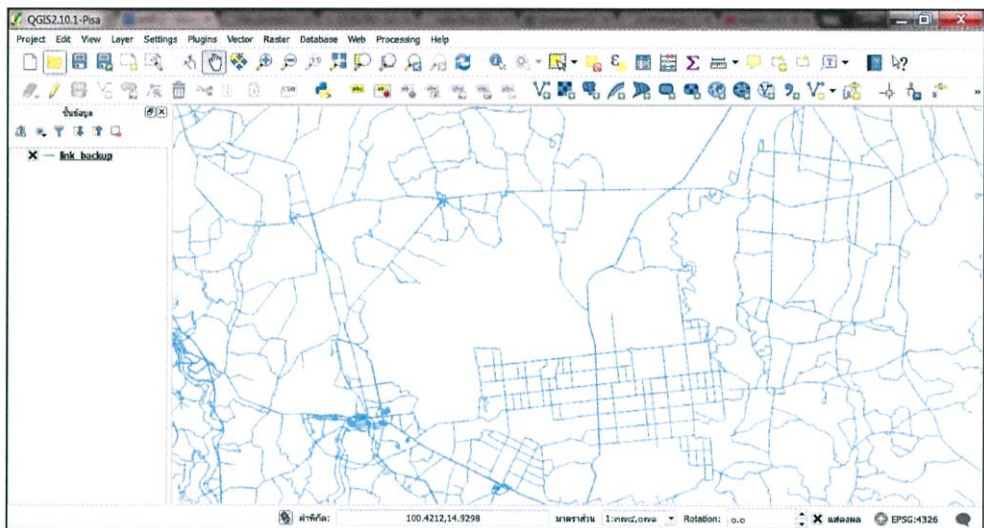
2.2.1 pgRouting

เป็น Open source ที่จัดหาฟังก์ชันสำหรับการใช้งานใน PostgreSQL/PostGIS เพื่อสร้างเครื่องมือในการคำนวณหาระยะทาง ซึ่งไม่ได้ใช้เฉพาะเรื่องระยะบนท้องถนนเท่านั้น แต่ยังสามารถใช้ได้กับข้อมูลต่างๆที่เกี่ยวกับระยะทาง เวลาบนท้องถนน น้ำมัน เงิน เช่น เส้นทางการเดินทางเรือ และระบบเครือข่ายเน็ตเวิร์ค เป็นต้น

นอกจากนี้ pgrouting ยังสามารถช่วยในการวางแผนการเดินทางเพื่อการค้า เช่น การจัดส่งสินค้าหลายๆที่พร้อมกัน ในการเดินทางแต่ละครั้งนั้นก็มีค่าใช้จ่าย pgRouting สามารถช่วยในการวางแผนในเส้นทางที่เหมาะสมเพื่อช่วยในการประหยัดเวลา ประหยัดค่าใช้จ่ายในส่วนต่างๆ

2.2.2 QGIS หรือ Quantum GIS

เป็นเครื่องมือที่ใช้ในการจัดการข้อมูลเชิงพื้นที่ที่จัดเป็น Open source สามารถวิเคราะห์ข้อมูลแบบเวกเตอร์และแบบราสเตอร์ ดังรูปที่ 2.2

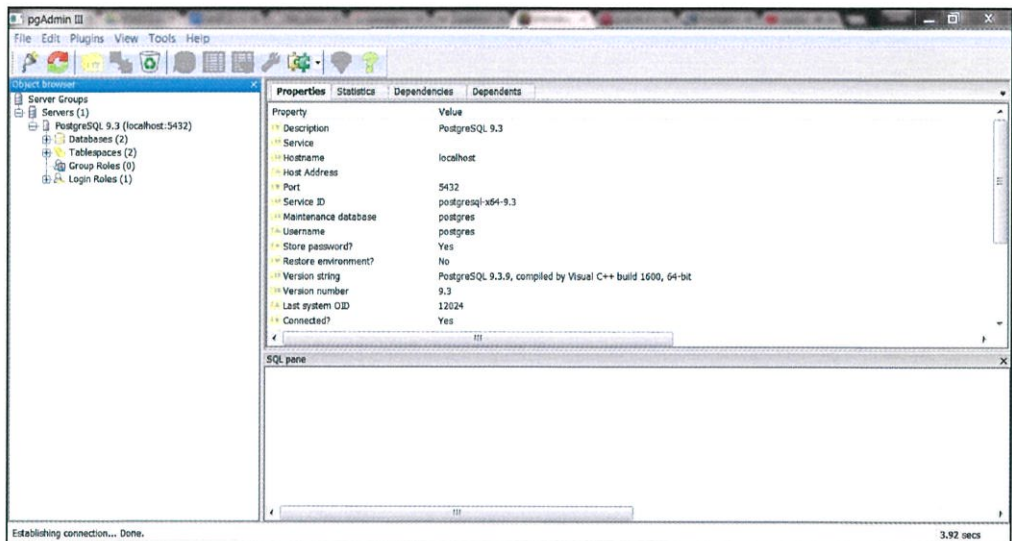


รูปที่ 2.2 หน้าต่างโปรแกรม QGIS หรือ Quantum GIS

2.3 ส่วนของฐานข้อมูล (Database)

2.3.1 postgresql

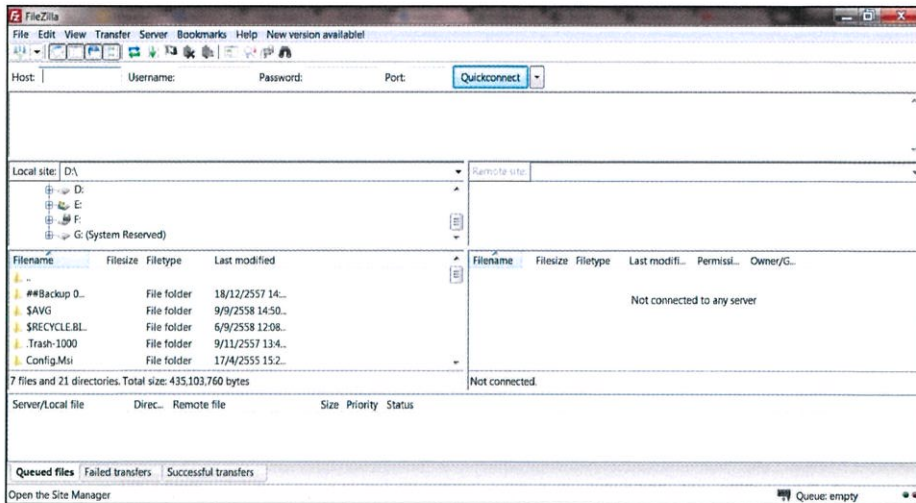
เป็นโปรแกรมจัดการฐานข้อมูลเชิงวัตถุ-สัมพันธ์ (object-relational) แบบ ORDBMS โดยสามารถใช้รูปแบบคำสั่งของภาษา SQL ได้เกือบทั้งหมด นอกจากนี้ยังเป็นระบบฐานข้อมูลที่ทันสมัยที่สุดของ OpenSource ที่สามารถนำไปใช้งานได้ฟรี ได้มีการพัฒนามาจาก POSTGRES 4.2 โดยมหาวิทยาลัยแคลิฟอร์เนีย (Berkeley Computer Science department, University of California.) PostgreSQL สามารถทำงานบนระบบปฏิบัติการได้ทั้ง Linux, UNIX (AIX, BSD, HP-UX, SGI Irix, Mac OS X, Solaris, Tru64) และ Windows ดังรูปที่ 2.3



รูปที่ 2.3 หน้าต่างโปรแกรม postgresql

2.3.2 filezilla

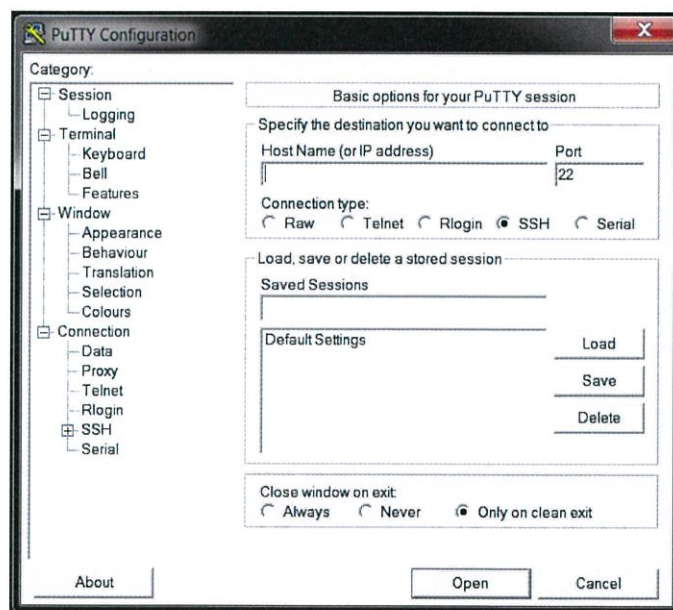
เป็นโปรแกรมที่ใช้ในการติดต่อกับ FTP server เพื่อทำการอัปโหลดและดาวน์โหลดไฟล์ โปรแกรมนี้ถือว่ามีความปลอดภัยเพราะมีการรองรับการถ่ายโอนไฟล์ผ่าน SSH (SFTP) ในการถ่ายโอนไฟล์สามารถ drag and drop ไฟล์ระหว่าง 2 ฝั่งข้อมูลได้อย่างสะดวกและง่ายต่อการใช้งาน ดังรูปที่ 2.4



รูปที่ 2.4 หน้าต่างโปรแกรม filezilla

2.3.3 putty

เป็น terminal เอาไว้เชื่อมต่อข้อมูล ส่งชุดคำสั่งแบบ Command Line จากเครื่องลูก Client เพื่อที่จะเข้าไปจัดการในเครื่องแม่ Server ขนาดเล็ก ดังรูปที่ 2.5



รูปที่ 2.5 หน้าต่างโปรแกรม putty

2.4 ฟังก์ชัน ไดจ์สตรา (Dijkstra's algorithm)

Dijkstra's algorithm ถูกคิดค้นขึ้นโดยนักวิทยาการคอมพิวเตอร์ชาวดัตช์นามว่า แอ็ดส์เคอร์ ไดจ์สตรา (Edsger Dijkstra) ในปี 1959 เพื่อแก้ไขปัญหาวิถีสั้นสุดจากจุดหนึ่งใด ๆ สำหรับกราฟที่มีความยาวของเส้นเชื่อมไม่เป็นลบ สำหรับขั้นตอนวิธีนี้จะหาระยะทางสั้นที่สุดจากจุดหนึ่งไปยังจุดใด ๆ ในกราฟโดยจะหาเส้นทางที่สั้นที่สุดไปที่ละจุดอย่างรวดเร็ว ๆ จนครบตามที่ต้องการ

บทที่ 3

วิธีการดำเนินการ

3.1 ทดสอบประสิทธิภาพฟังก์ชัน Shortest Path Dijkstra

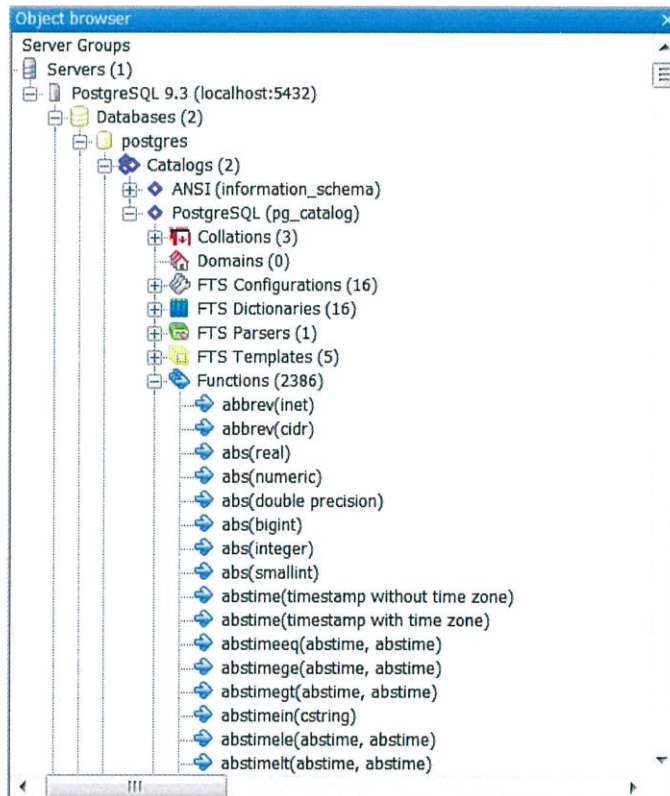
3.1.1 การติดตั้ง pgrouting

ในการติดตั้งฟังก์ชัน Shortest Path Dijkstra นั้นสามารถใช้คำสั่ง SQL ในการติดตั้งฟังก์ชัน โดยการติดตั้งนั้นจะติดตั้งบนโปรแกรม PostgreSQL และไม่เฉพาะฟังก์ชัน Shortest Path Dijkstra เท่านั้นแต่ยังมีฟังก์ชันอื่นๆที่มีให้เลือกใช้ตามประเภทของงานอีกด้วย ดังรูป 3.1

ตารางที่ 3.1 คำสั่ง SQL ที่ใช้ในการติดตั้งฟังก์ชันในการใช้งาน

```
CREATE EXTENSION postgis;
```

```
CREATE EXTENSION pgrouting;
```



รูปที่ 3.1 ฟังก์ชันที่มีเมื่อทำการติดตั้งแล้ว

3.1.2 การจัดเตรียมข้อมูลบนฐานข้อมูล สร้างฐานข้อมูลขึ้นมาชื่อว่า routing บนฐานข้อมูล

3.1.3 การคำนวณฟังก์ชัน Shortest Path Dijkstra

1) เพิ่มคอลัมน์ข้อมูล

ในการคำนวณฟังก์ชันจำเป็นจะต้องเพิ่มคอลัมน์เพื่อใช้ในการรองรับค่าที่ได้จากการคำนวณฟังก์ชัน Shortest path dijkstra

ตารางที่ 3.2 คำสั่ง SQL เพิ่มคอลัมน์ของ source และ target

```
ALTER TABLE link ADD COLUMN "source" integer;
```

```
ALTER TABLE link ADD COLUMN "target" integer;
```

2) ใช้คำสั่งสร้าง Topology

เมื่อทำการเพิ่มคอลัมน์เพื่อใช้ในการรับค่าที่ได้จากการคำนวณฟังก์ชันแล้ว จะเริ่มทำการคำนวณฟังก์ชัน โดยจะใช้คำสั่ง SQL ในการสั่งเริ่มคำนวณฟังก์ชัน ซึ่งค่าเหล่านี้จะบอกลำดับความเชื่อมโยงกันของแต่ละช่วงถนนว่าช่วงถนนเส้นใดติดกับช่วงถนนเส้นใด

ตารางที่ 3.3 คำสั่ง SQL ที่ใช้คำสั่งเพื่อคำนวณฟังก์ชันสร้าง Topology

```
SELECT pgr_createTopology ('routing', 0.00001, 'the_geom', 'gid');
```

3) CREATE INDEX

เมื่อทำการคำนวณฟังก์ชันเสร็จแล้วจะได้ค่าของ source และ target มาแล้วจะทำสร้าง index ข้อมูลของ source และ target

ตารางที่ 3.4 คำสั่ง SQL ทำการ CREATE INDEX

```
CREATE INDEX links_source_idx ON link ("source");
```

```
CREATE INDEX links_target_idx ON link ("target");
```

4) เพิ่มตัวแปร cost และ reverse_cost

เพิ่มตัวแปร cost และ reverse_cost บนฐานข้อมูล ซึ่งค่าตัวแปรเหล่านี้จะบอกถึงถนนเป็นเส้นทาง oneway กำหนดให้ cost=1 และ reverse_cost=-1

ตารางที่ 3.5 คำสั่ง SQL เพื่อเพิ่มคอลัมน์และกำหนดค่าของ cost และ reverse_cost

ALTER TABLE routing ADD COLUMN "cost" integer;
ALTER TABLE outing ADD COLUMN " reverse_cost" integer;
UPDATE link SET cost = 1;
UPDATE link SET reverse_cost = -1;

3.1.4 ทดสอบประสิทธิภาพฟังก์ชัน Shortest Path Dijkstra

การทดสอบประสิทธิภาพของฟังก์ชัน Shortest Path Dijkstra จะนำเส้นทางทางเปรียบเทียบกับเครื่องมือที่มีประสิทธิภาพสูงกว่าอย่างเช่น Google Map เป็นต้น ซึ่งการทดสอบประสิทธิภาพของฟังก์ชันนั้นจะแบ่งเป็นประเภทต่างๆเพื่อการเปรียบเทียบที่ชัดเจนและสามารถแยกแยะเส้นทางที่มีความหลากหลายตามลักษณะภูมิประเทศ ในการทดสอบประสิทธิภาพจะทดสอบกับเครื่องมือ 2 ชนิดคือ Google Map และ รถที่ติดตั้ง GPS

1) Google Map

จะแบ่งประเภทเส้นทางการทดสอบเป็น 5 ประเภทดังนี้

- 1.1) ประเภทเส้นทางโค้ง
- 1.2) ประเภทเส้นทางตรง
- 1.3) ประเภทเส้นทางเลี้ยว
- 1.4) ประเภทเส้นทางระยะยาว
- 1.5) ประเภทเส้นทางยูเทิร์น

ขั้นตอนการทดสอบประสิทธิภาพกับ Google Map

1. ใช้คำสั่ง SQL เพื่อให้ได้ข้อมูลระหว่างจุดเริ่มต้นและจุดสิ้นสุดของเส้นทาง ค่าตัวเลข 30 และ 60 เป็นค่า source และ target ของจุดเริ่มต้นและจุดสิ้นสุด

ตารางที่ 3.6 คำสั่ง SQL ที่ใช้ในการ Query ข้อมูลของเส้นทาง

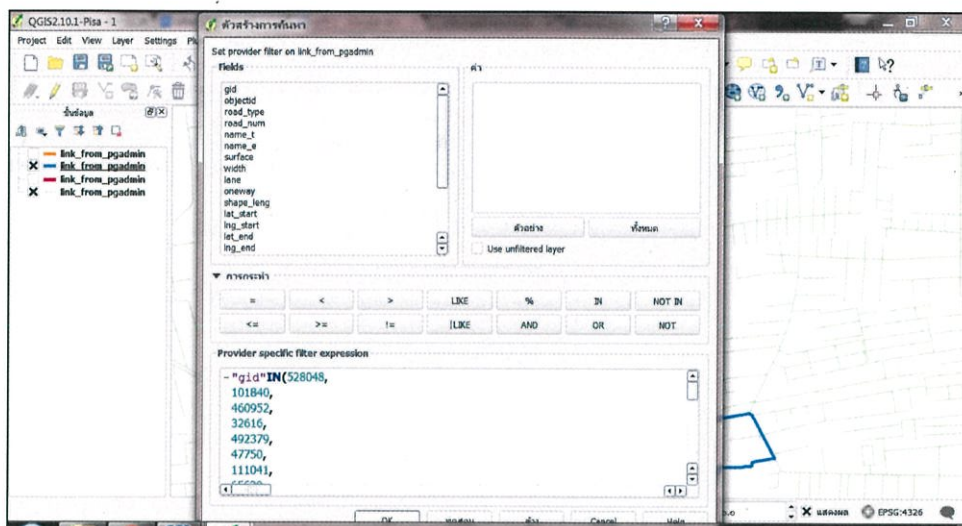
```
SELECT seq, id1 AS node, id2 AS edge, cost FROM pgr_dijkstra
('SELECT gid AS id,
      source::integer,
      target::integer,
      shape_len::double precision AS cost
FROM routing',
30,60, false, false);
```

เมื่อใช้คำสั่งในการ Query ข้อมูลจะได้ข้อมูล 4 คอลัมน์ดังในภาพนั้นคือ seq node edge และ cost ซึ่งข้อมูลที่ได้มาจะแสดงตั้งแต่ตั้งจุดเริ่มต้นจนถึงจุดสิ้นสุดว่ามีจำนวนเท่าไรและอะไรบ้าง ดังรูป 3.2

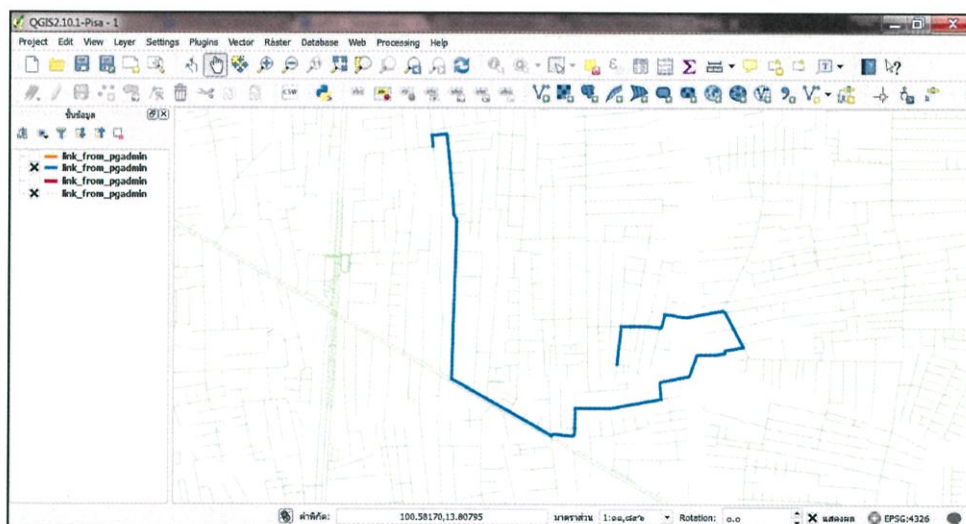
seq	node	edge	cost
integer	integer	integer	double precision
1	30	358259	17.248850476
2	1	381233	211.9692.8083673731
3	2	381234	331.258309.33237455
4	3	465107	323790176.34756051
5	4	465108	33675811.997033776
6	5	53483	483915190.38001732
7	6	53484	358962246.26101605
8	7	475645	33680125.212647707
9	8	475646	362246598.33493486

รูปที่ 3.2 ข้อมูลที่ได้จากการ Query ของจุดเริ่มและจุดสิ้นสุดของเส้นทาง

2. นำข้อมูลที่ได้มาจากการ Query ไปแสดงบนโปรแกรม QGIS เพื่อดูเส้นทางที่ได้ โดยจะนำข้อมูลในส่วนของ edge ไปแสดงซึ่ง edge เป็นค่าที่บอกตำแหน่งของเส้นทาง การใส่ค่าตำแหน่ง (edge) เพื่อดูเส้นทางให้กดไปที่ menu View=>Filter ให้เลือกค่าที่ใส่เป็นค่า gid เพราะค่า edge และ gid เป็นค่าเดียวกัน ที่แสดงตำแหน่งบนเส้นทาง ดังรูป 3.3

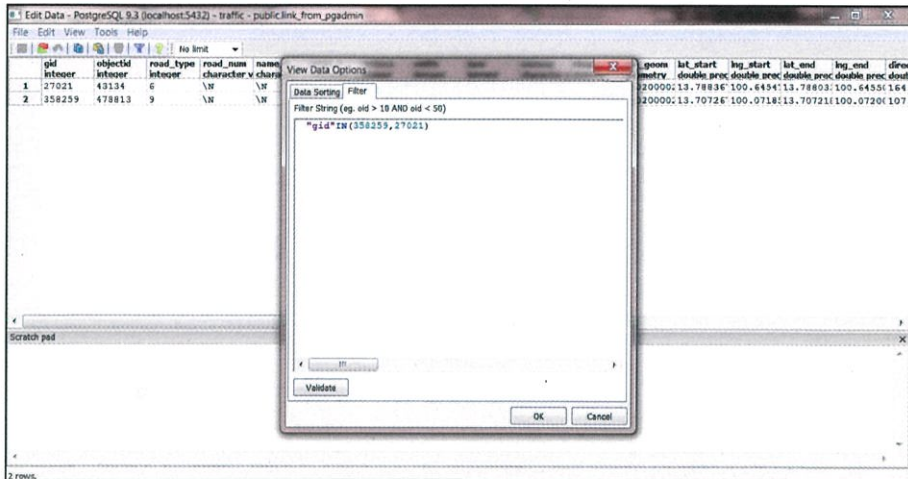


รูปที่ 3.3 การนำข้อมูลเส้นทางมาเปิดดูบนโปรแกรม QGIS



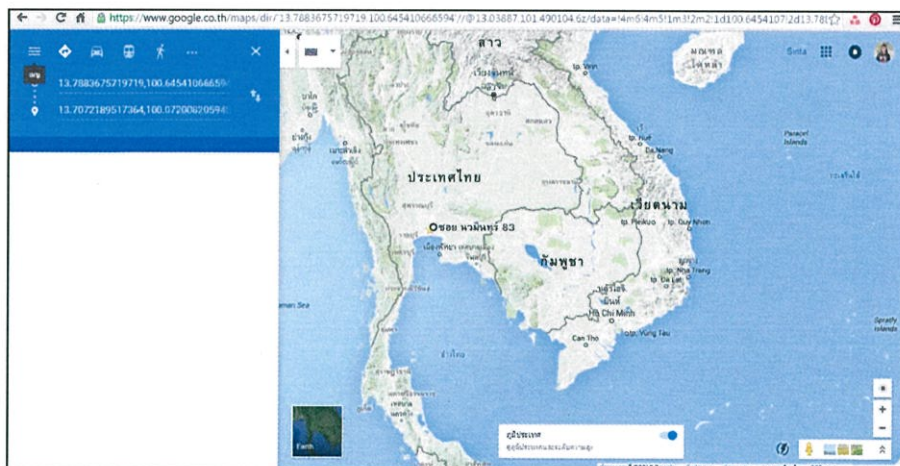
รูปที่ 3.4 เส้นทางที่ได้จากการ Query แสดงบนโปรแกรม QGIS

3. การแสดงเส้นทางบน Google Map ของจุดเริ่มต้นและจุดสิ้นสุดเดียวกัน โดยค่า edge ของจุดเริ่มต้นและจุดสิ้นสุดไปค้นหาบนฐานข้อมูลเพื่อนำเอาค่า lat และ lng ไปค้นหาบน Google Map การค้นหาค่า lat_start lng_start lat_end และ lng_end จากฐานข้อมูล ไปที่ตารางฐานข้อมูล menu Tools=>sort/filter และใช้คำสั่งค้นหาจากข้อมูลค่า edge จุดเริ่มต้นและจุดสิ้นสุด ดังรูป 3.5



รูปที่ 3.5 การนำค่า edge ไปหาค่า lat_start lng_start lat_end และ lng_end

เมื่อได้ค่า lat_start และ lng_start จากฐานข้อมูลแล้ว นำค่าดังกล่าวไปค้นหาบน google map โดยต้องเลือกเมนูเส้นทาง ดังรูปที่ 3.6



รูปที่ 3.6 การนำค่า lat_start lng_start lat_end และ lng_end ไปค้นหาเส้นทางบน google map

2) การเปรียบเทียบเส้นทางบนรถที่ติดตั้ง GPS

ซึ่งจะแบ่งประเภทเส้นทางการทดสอบเป็น 5 ประเภทดังนี้

- 2.1) ประเภททางวงเวียน
- 2.2) ประเภททางเลี้ยว
- 2.3) ประเภททางแยก
- 2.4) ประเภททางตรง
- 2.5) ประเภททางโค้ง

ขั้นตอนการเปรียบเทียบกับเครื่องมือของรถที่ติดตั้ง GPS

- 1.ทำการ Query เส้นทางบน Map และบนฐานข้อมูลโดยจะใช้ช่วงเวลาในการ Query เนื่องจากรถไปวิ่งไประยะเวลาหนึ่งจะได้เส้นทางตามช่วงเวลานั้น ดังรูป 3.7

ตารางที่ 3.7 ตัวอย่างคำสั่งที่ใช้ Query ข้อมูลของรถที่ติดตั้ง GPS

```
SELECT * FROM gps.gpsdata WHERE date_time>'2015-06-08 07:40:00' AND
date_time<'2015-06-08 07:42:00'
```



รูปที่ 3.7 เส้นทางที่ได้จากการ Query ช่วงเวลาของรถที่ติดตั้ง GPS

2. ใช้คำสั่ง SQL เพื่อให้ได้ข้อมูลระหว่างจุดเริ่มต้นและจุดสิ้นสุดของเส้นทาง โดยค่าของ source และ target จะต้องเป็นค่าของจุดเริ่มต้นและจุดสิ้นสุดเดียวกับรถที่ติดตั้ง GPS โดยค่า GPS จุดแรกและจุดสิ้นสุดที่ Query บนฐานข้อมูล จะบอกตำแหน่งว่า link id อะไรจากนั้นนำ link id ที่ได้ไปหา source และ target บนฐานข้อมูล routing ที่สร้างไว้แล้วใช้คำสั่งต่อไปนี้ Query บนฐานข้อมูล routing เพื่อให้ได้เส้นทางที่ได้มาจากฟังก์ชัน

ตารางที่ 3.8 คำสั่ง SQL ที่ใช้ในการ Query ข้อมูลของเส้นทาง

```
SELECT seq, id1 AS node, id2 AS edge, cost FROM pgr_dijkstra ('SELECT
gid AS id,
source::integer,
target::integer,
shape_leng::double precision AS cost
FROM routing',
30,60, false, false);
```

เมื่อใช้คำสั่งในการ Query ข้อมูลเส้นทางจะได้ข้อมูล 4 คอลัมน์ดังในภาพนั้นคือ seq node edge และ cost ซึ่งข้อมูลที่ได้มาจะแสดงตั้งแต่จุดเริ่มต้นจนถึงจุดสิ้นสุดว่ามีจำนวนเท่าไรและอะไรบ้าง ดังรูป 3.8

seq	node	edge	cost
integer	integer	integer	double precision
1	30	358259	17.248850476
2	1381233	211969	2.8083673731
3	2381234	331258	309.33237455
4	3465107	323790	176.34756051
5	4465108	336758	11.997033776
6	553483	483915	190.38001732
7	653484	358962	246.26101605
8	7475645	336801	25.212647707
9	8475646	362246	598.33493486

รูปที่ 3.8 ข้อมูลที่ได้จากการ Query ของจุดเริ่มและจุดสิ้นสุดของเส้นทาง

3.2 การพัฒนาปรับปรุงฟังก์ชัน Shortest Path Dijkstra

Shortest Path Dijkstra เป็นฟังก์ชันที่สามารถเพิ่มตัวแปรที่ใช้ในการคำนวณฟังก์ชันได้ ในการปรับปรุง Shortest Path Dijkstra จะเพิ่มตัวแปรที่เรียกว่า count เป็นค่าที่บอกถึงความนิยมในการใช้เส้นทางนั้นๆ ซึ่งค่าความนิยมในการใช้เส้นทางจะได้มาจากการ Mapmatch

Mapmatch เป็นวิธีการนำจุด GPS ที่กระจาย มาเลือกเฉพาะจุด GPS ที่ตกอยู่บนเส้นทางเท่านั้น จากการ Mapmatch จะได้จุด GPS ที่ตรงตามเส้นทาง ดังรูป 3.11 และ รูปที่ 3.12



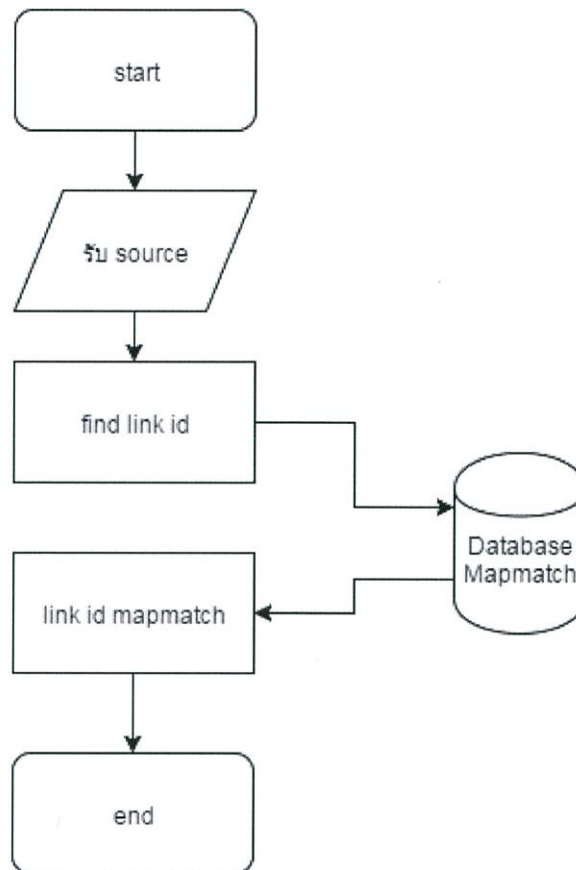
รูปที่ 3.11 GPS ที่ยังไม่ผ่านการ mapmatch



รูปที่ 3.12 GPS ที่ยังผ่านการ mapmatch แล้ว

3.2.1 เขียนโปรแกรมดึงค่า count

เพิ่มค่า cost ให้กับข้อมูลที่จะนำไปใช้ในการคำนวณค่า count ที่ได้มาจากการเขียนโปรแกรมดึงข้อมูลมาจากการ mapmatch ที่มีอยู่แล้วบน server ดังรูปที่ 3.13



รูปที่ 3.13 flow chart แสดงการดึงข้อมูลของค่า count

3.2.2 ปรับปรุงฐานข้อมูล

สร้าง database ชื่อว่า frequently เพื่อเก็บข้อมูลค่าความนิยมในการใช้เส้นทางนั้นๆไว้ แล้วนำไปใช้ในการตัดสินใจหาเส้นทางที่ดีที่สุดและเพิ่มข้อมูลบนฐานข้อมูล routing ดังรูปที่ 3.14

ตารางที่ 3.9 ส่วนประกอบของฐานข้อมูล frequently

frequently
link_id integer ตัวเลขแทนช่วงถนนที่ตัดเป็นช่วงสั้นๆ
count double precision ค่าความนิยมในการใช้เส้นทางนั้นๆ

	link_id integer	count integer
1	19	6
2	20	2
3	23	4
4	24	7
5	25	2
6	31	4
7	73	25
8	78	5
9	87	36
10	88	35
11	94	130
12	99	5
13	105	9
14	122	3
15	132	2
16	136	27
17	138	4
18	140	4
19	146	6
20	150	4

รูปที่ 3.14 ฐานข้อมูลของ frequently

ค่าความนิยมที่ใช้ได้มาจากรถที่ติดตั้ง GPS เมื่อรถวิ่งตามถนนจะเก็บความถี่ของจุด GPS ไว้ถ้ารถวิ่งถนนเส้นทางใดบ่อยความถี่ของ GPS ในช่วงถนนนั้นๆก็จะมีค่าความถี่ของจุด GPS มาก ดังรูปที่ 3.15



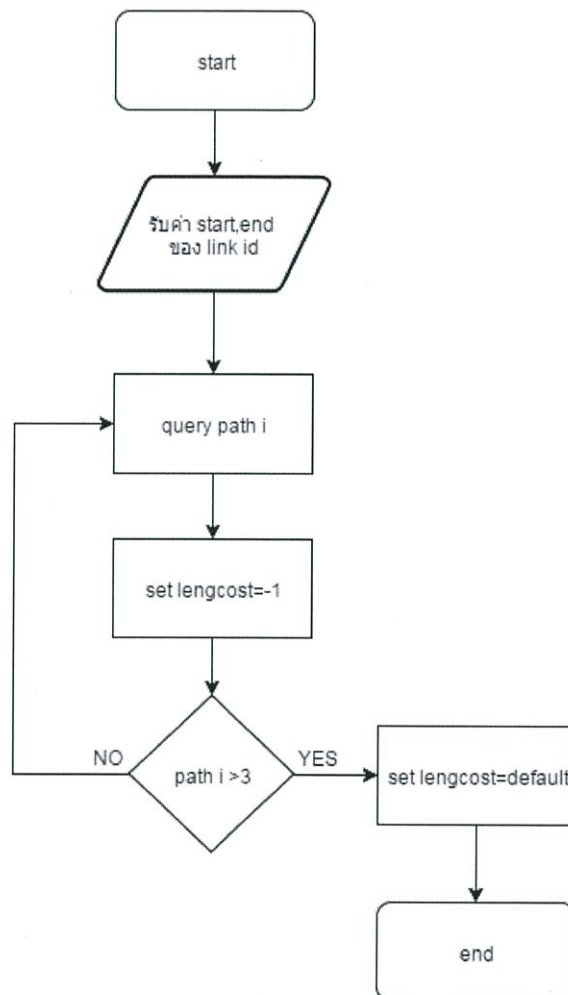
รูปที่ 3.15 ขนาดความถี่ของจุด GPS ตามเส้นทางที่มีการเดินทาง

3.2.3 การทำ Multiple Path

ต้องทำการคำนวณฟังก์ชันใหม่อีกครั้ง ในการทำ Multiple Path จะช่วยในการเพิ่มทางเลือกของเส้นทางว่าจะเลือกใช้เส้นทางใด เพราะการใช้แค่เส้นทางเดียวไม่สามารถบอกได้ว่าเส้นทางนั้นมีประสิทธิภาพมากที่สุด ฟังก์ชันถ้ากำหนดตัวแปรใดๆมีค่าเท่ากับ -1 จะถือว่าข้อมูลในคอลัมน์นั้นๆจะถือว่าไม่มีข้อมูลนั้นฟังก์ชันจะเลยไปใช้ข้อมูลอื่นที่เป็นไปได้ ดังนั้นเมื่อกำหนดให้เส้นทางใดเป็น -1 แล้วฟังก์ชันจะสามารถ Query เส้นทางได้อีกเส้นทาง ทำให้เส้นทางทั้งหมดที่เป็นไปได้สามารถ Query จากฐานข้อมูลจนกว่าการ Query เส้นทางจะไม่มีข้อมูลส่งมานั้นจะถือว่าเส้นทางที่เป็นไปได้มากที่สุด ดังรูปที่ 3.16

ตารางที่ 3.10 คำสั่ง SQL ที่ใช้คำสั่งเพื่อคำนวณฟังก์ชันสร้าง Topology

```
SELECT pgr_createTopology ('routing', 0.00001, 'the_geom', 'gid');
```



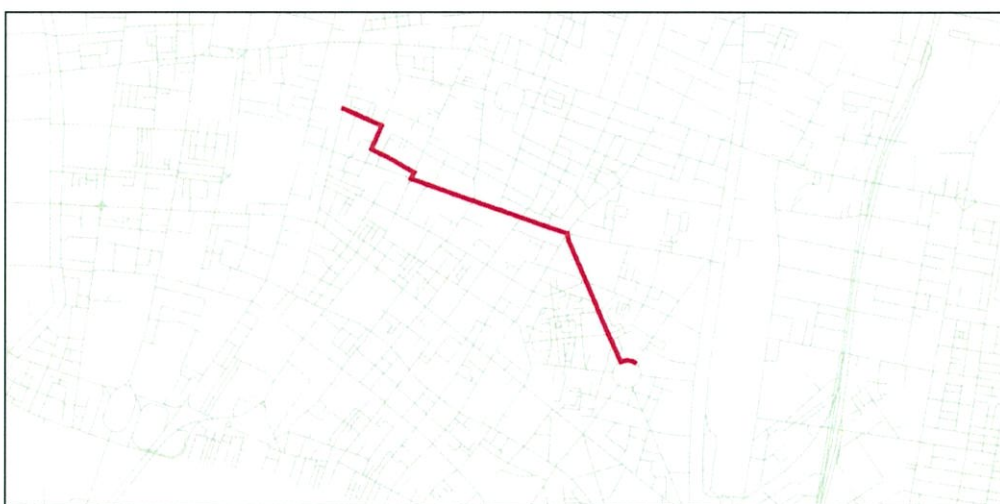
รูปที่ 3.16 flow chart การทำ multiple path

3.2.4 การเลือก Path ที่ได้มาจากการทำ Multiple Path

การเลือกเส้นทางที่ดีที่สุดนั้นจะเลือกจากเส้นทางที่มีค่าความนิยมในการใช้เส้นทางนั้นสูงสุด ซึ่งจะนำค่าของ count ของแต่ละ link id มาบวกกัน แต่ละเส้นทางจะมีค่า count รวมกันไม่เท่ากันจะเลือกเส้นทางค่า count ที่มีค่าสูงสุด ดังรูปที่ 3.17 และ 3.18



รูปที่ 3.17 ตัวอย่าง Multiple path



รูปที่ 3.18 ตัวอย่างเส้นทางที่ดีที่สุดจาก Multiple path

บทที่ 4

ผลการวิจัยและการอภิปรายผล

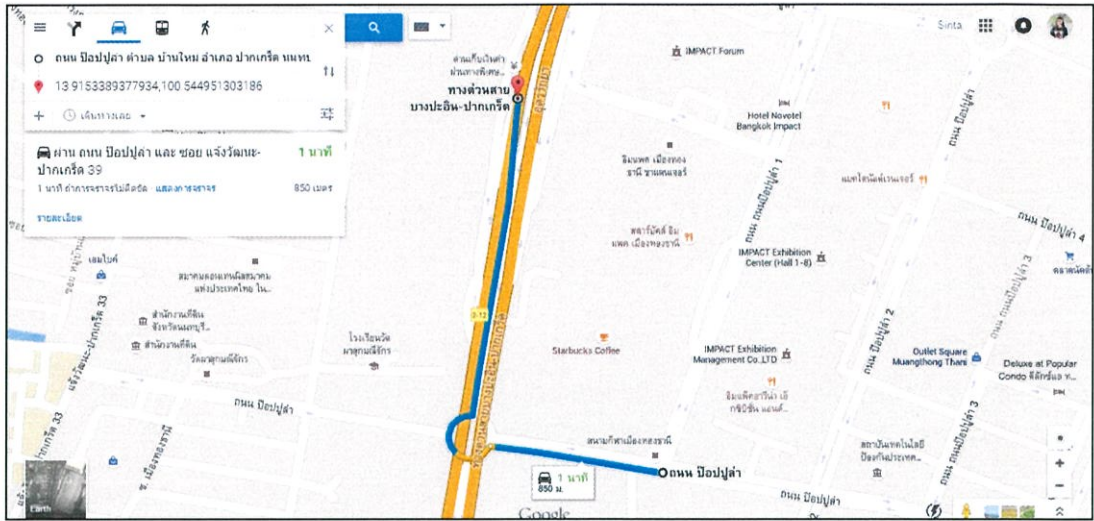
4.1 ประสิทธิภาพที่วัดได้ของเส้นทางแต่ละประเภท

ตารางแสดงการเปรียบเทียบเส้นทางระหว่าง Goole Maps และเส้นทางของข้อมูลบน Database ที่แสดงให้เห็นเปอร์เซ็นต์ความถูกต้อง เมื่อนำจุดเริ่มต้นและจุดสิ้นสุดเดียวกัน

ตาราง 4.1 Accuracy ที่ทำการ Test Routing ระหว่าง Google Maps กับข้อมูลแผนที่บน Database

	ทางโค้ง	ทางตรง	ทางเลี้ยว	ทางระยะยาว	ยูเทิร์น
Routing ตรงกัน	90%	85.72	40%	50%	57.15%
Routing ไม่ตรงกัน	10%	14.28%	60%	50%	42.85%

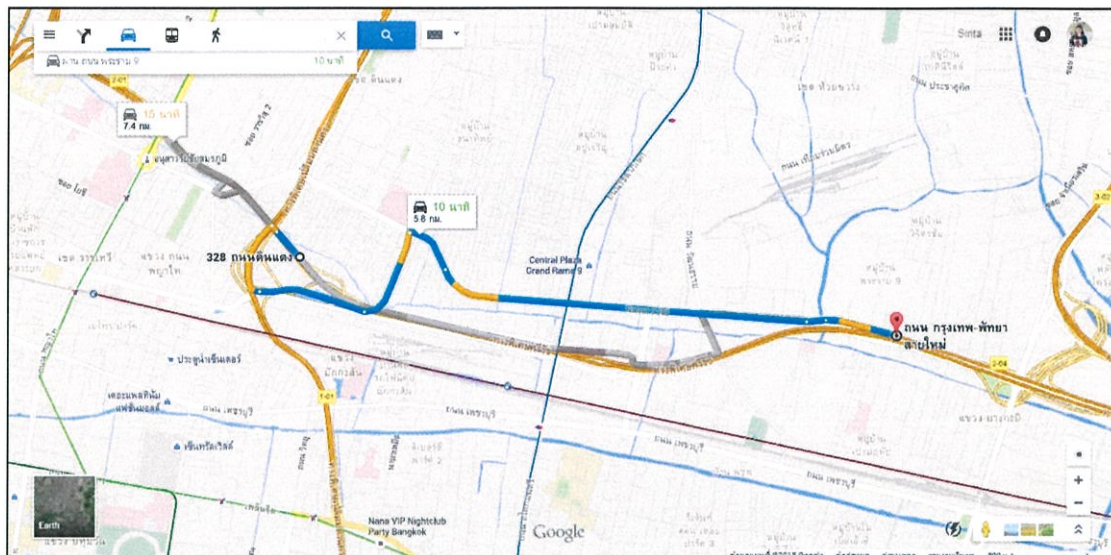
ตัวอย่างของเส้นทางที่ไม่ตรงกันระหว่าง Google Maps และ ข้อมูลบน Database แต่ละประเภท



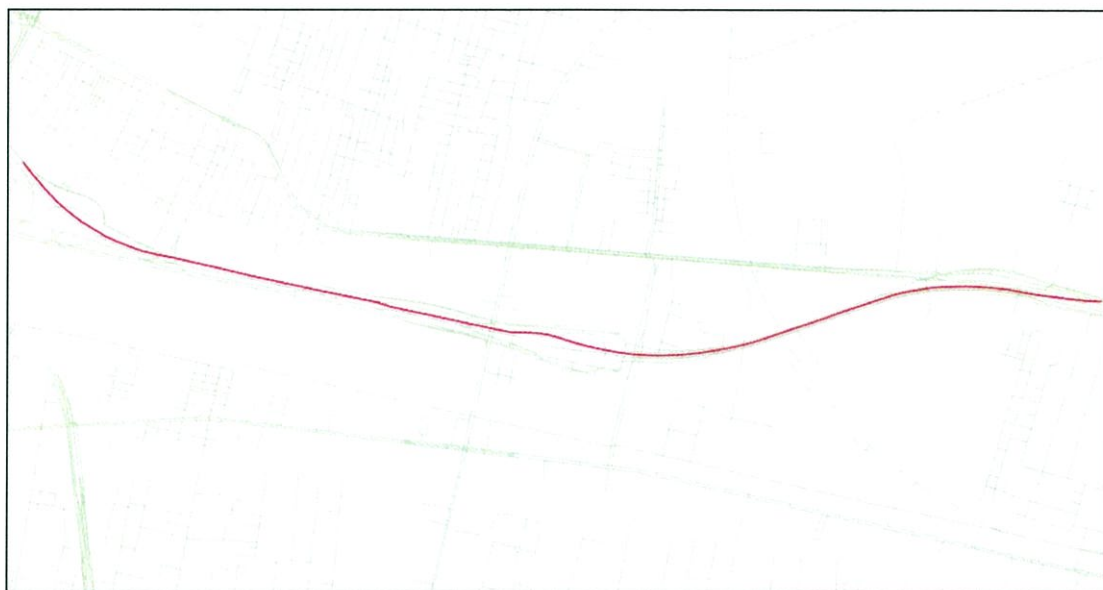
รูปที่ 4.1 เส้นทางบน Google Maps ของประเภททางโค้ง



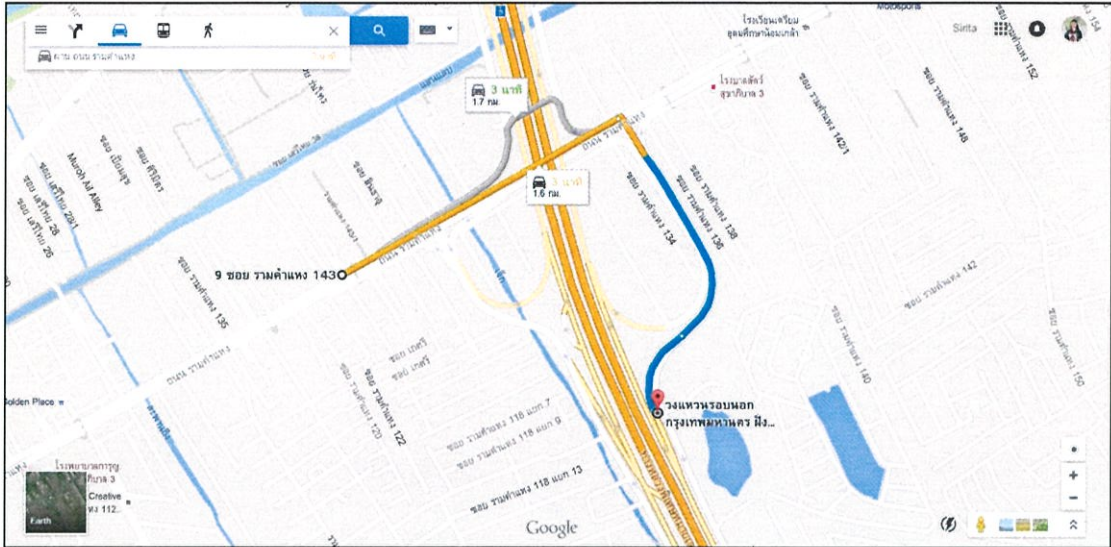
รูปที่ 4.2 เส้นทางบน Database ของประเภททางโค้ง



รูปที่ 4.3 เส้นทางบน Google Maps ของประเภททางตรง



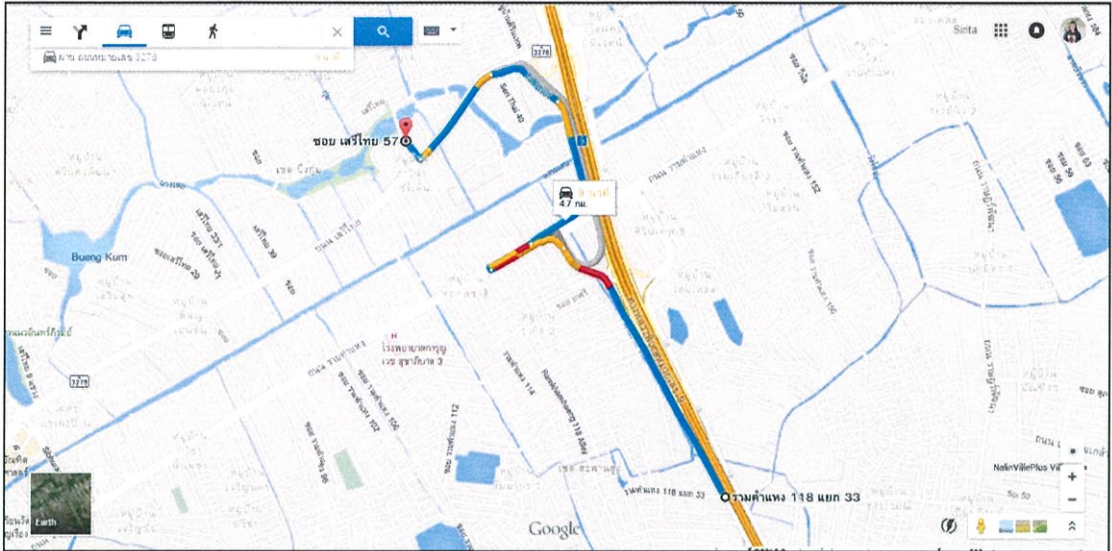
รูปที่ 4.4 เส้นทางบน Database ของประเภททางตรง



รูปที่ 4.5 เส้นทางบน Google Maps ของประเภททางเดี่ยว



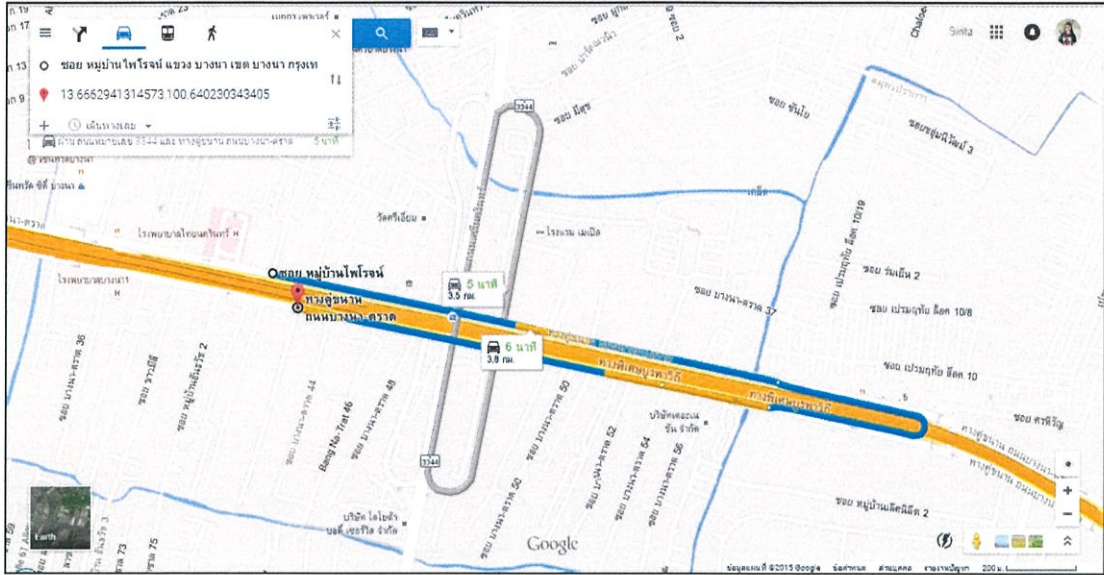
รูปที่ 4.6 เส้นทางบน Database ของประเภททางเดี่ยว



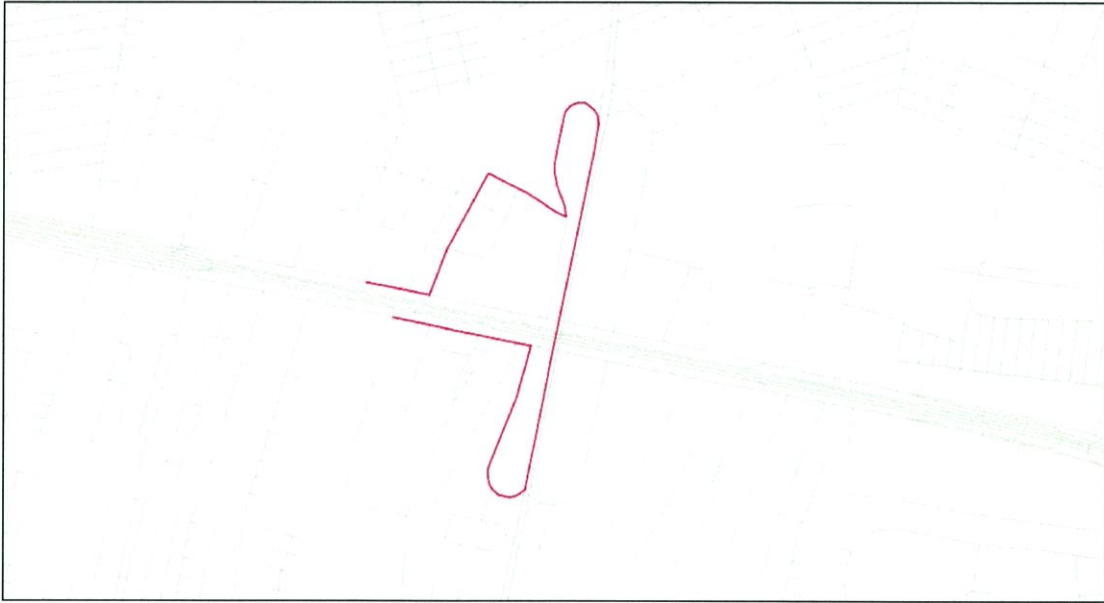
รูปที่ 4.7 เส้นทางบน Google Maps ของประเภททางระยะยาว



รูปที่ 4.8 เส้นทางบน Database ของประเภททางระยะยาว



รูปที่ 4.9 เส้นทางบน Google Maps ของประเภททางยูเทิร์น



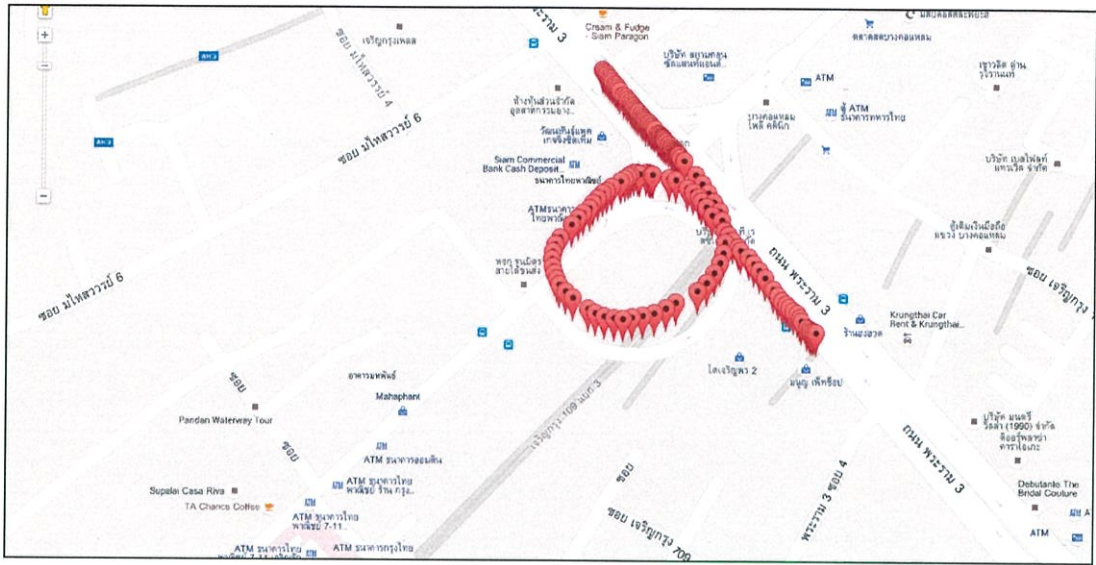
รูปที่ 4.10 เส้นทางบน Database ของประเภททางยูเทิร์น

ตารางแสดงการเปรียบเทียบเส้นทางระหว่างเส้นทางของรถที่ติดตั้ง GPS และเส้นทางของข้อมูลบน Database ที่แสดงให้เห็นเปอร์เซ็นต์ความถูกต้อง เมื่อนำจุดเริ่มต้นและจุดสิ้นสุดเดียวกัน

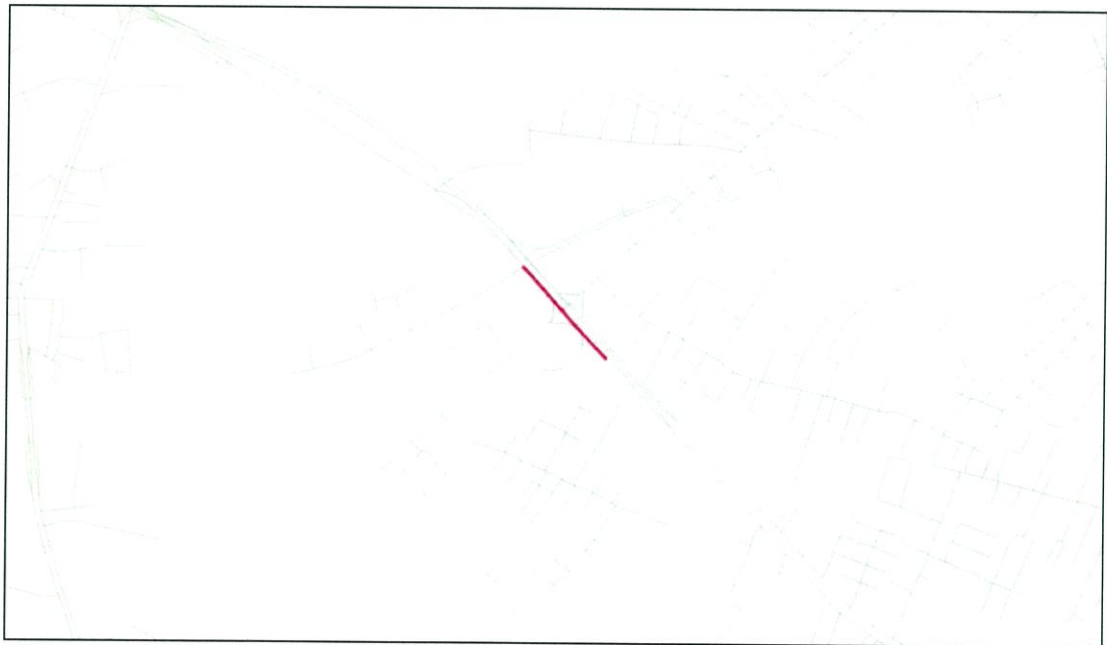
ตาราง 4.2 Accuracy ที่ทำการ Test Routing ระหว่าง เส้นทางเดินรถกับข้อมูลแผนที่บน Database

	ทางตรง	ทางเลี้ยว	ทางโค้ง	ทางแยก	วงเวียน
Routing ตรงกัน	100%	75%	83.33%	100%	-
Routing ไม่ตรงกัน	-	22.2%	16.67%	-	100%

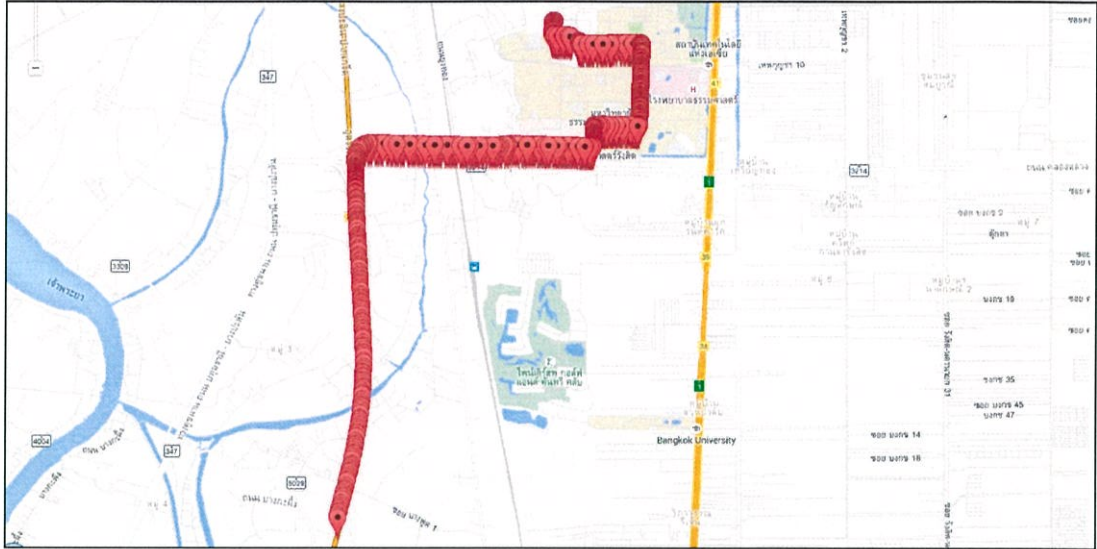
ตัวอย่างของเส้นทางที่ไม่ตรงกันระหว่าง Google Maps และ ข้อมูลบน Database แต่ละประเภท ดังรูปภาพต่อไปนี้



รูปที่ 4.11 ตัวอย่างเส้นทางจากรถที่ติดตั้ง GPS ของประเภททางวงเวียน



รูปที่ 4.12 ตัวอย่างเส้นทางบน Database ของประเภททางวงเวียน



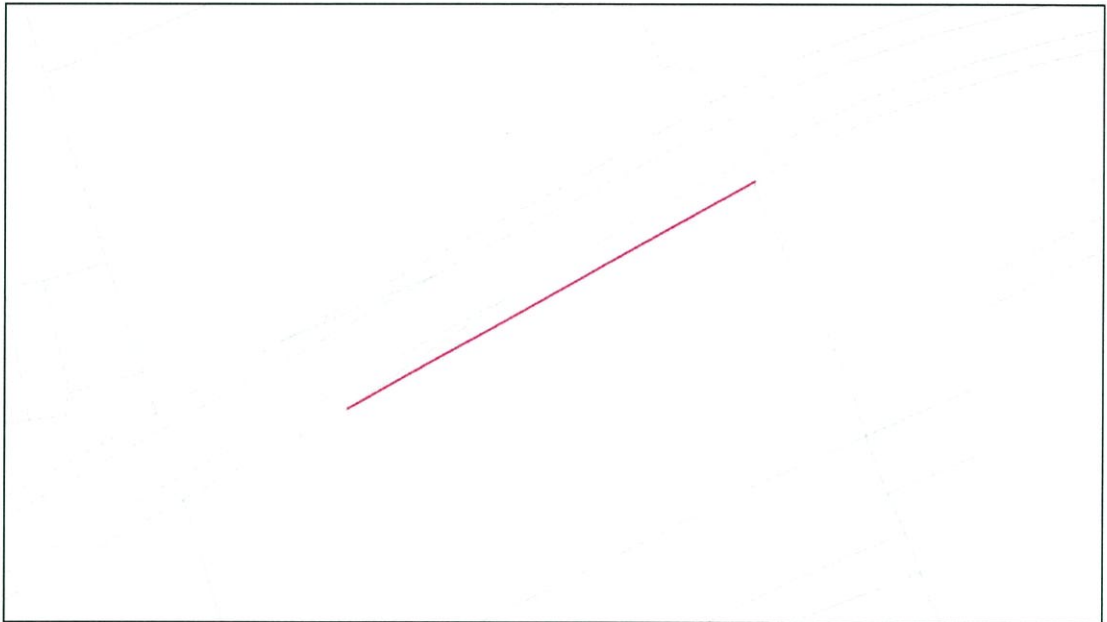
รูปที่ 4.13 ตัวอย่างเส้นทางจากรถที่ติดตั้ง GPS ของประเภททางเลียว



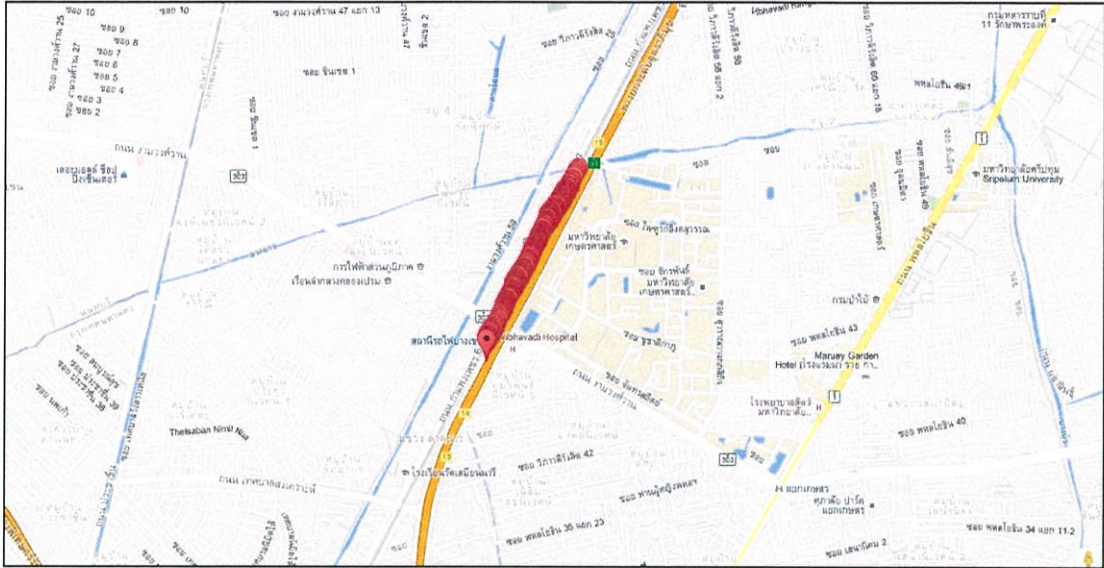
รูปที่ 4.14 ตัวอย่างเส้นทางบน Database ของประเภททางเลียว



รูปที่ 4.15 ตัวอย่างเส้นทางจากรถที่ติดตั้ง GPS ของประเภททางแยก



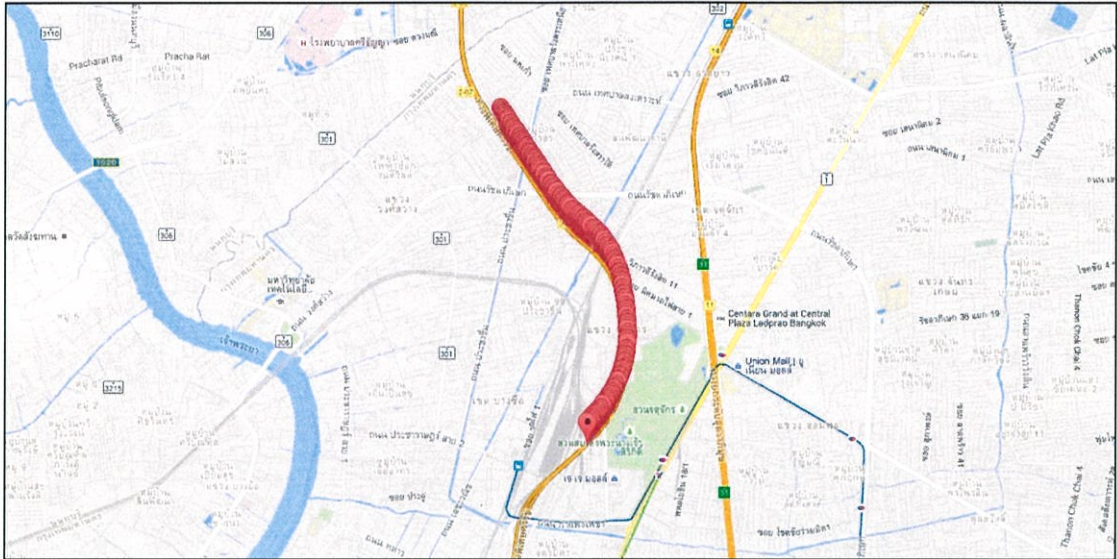
รูปที่ 4.16 ตัวอย่างเส้นทางบน Database ของประเภททางแยก



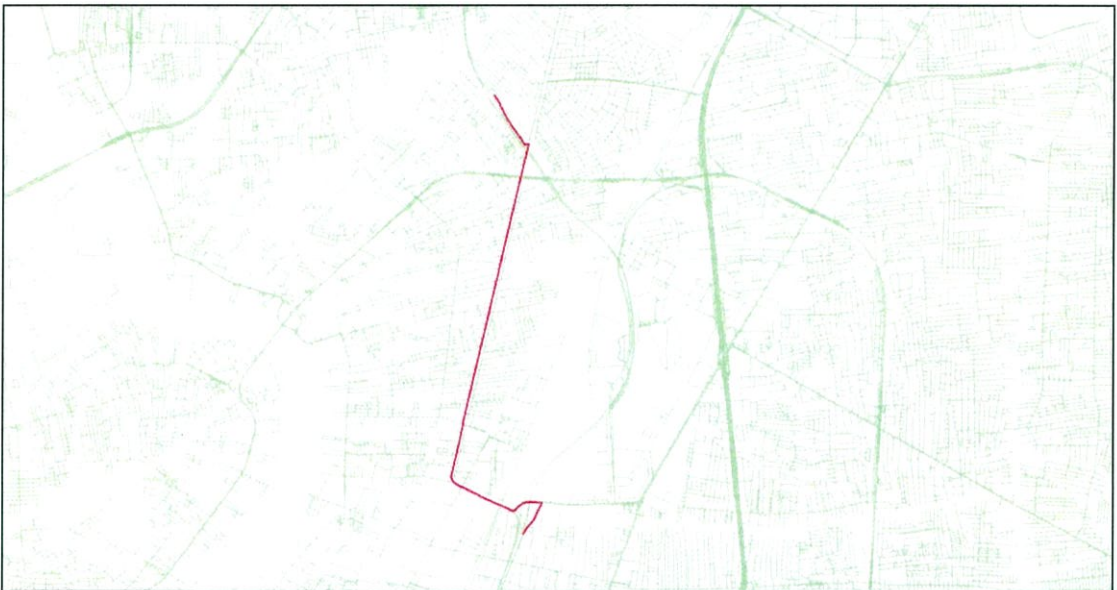
รูปที่ 4.17 ตัวอย่างเส้นทางจากรถที่ติดตั้ง GPS ของประเภททางตรง



รูปที่ 4.18 ตัวอย่างเส้นทางบน Database ของประเภททางตรง



รูปที่ 4.19 ตัวอย่างเส้นทางจากรถที่ติดตั้ง GPS ของประเภททางโค้ง



รูปที่ 4.20 ตัวอย่างเส้นทางบน Database ของประเภททางโค้ง

4.2 สรุปสาเหตุที่ประเภทของเส้นทางมีเปอร์เซ็นต์ของ Routing ไม่ตรงกัน

- วงเวียนผิด 100 เปอร์เซ็นต์ เพราะไม่มีเส้นทางเป็นวงเวียน มีแต่เป็นทางแยก
- ไม่สามารถแยกแยะสถานที่ที่ไม่สามารถใช้เดินรถได้ เช่น สถานที่ราชการ สถานที่ส่วนบุคคล เป็นต้น
- เส้นทางเล็กๆที่รถไม่จำเป็นต้องเข้าไป เช่น ซอยเล็กๆในหมู่บ้านหรือเส้นทางที่ไม่นิยมใช้ในการเดินทาง เป็นต้น

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

การพัฒนาและปรับปรุงฟังก์ชัน Shortest path dijkstra นั้นเพื่อต้องการปรับปรุงฟังก์ชันให้เข้ากับฐานข้อมูลที่มีเพื่อให้เกิดประสิทธิภาพในการใช้งานสูงสุด จากการนำแนวทางการทำ Multiple path มาแก้ปัญหานั้น ทำให้การเลือกเส้นทางที่ดีที่สุดมีขอบเขตที่ชัดเจนและได้แนวทางการแก้ปัญหาเหล่านั้นไปเขียนโปรแกรมเป็น Application Programming Interface หรือ API เพื่อให้ผู้ที่ต้องการนำไปใช้ได้ตามประเภทของงานการจราจร

5.2 ข้อเสนอแนะ

จากการพัฒนาและปรับปรุงฟังก์ชันเพื่อให้ตรงตามการใช้งาน แต่ก็ยังสามารถจะพัฒนาต่อไปได้อีกด้วยการเพิ่มตัวแปรที่ใช้ในการคำนวณฟังก์ชันเพื่อให้มีความหลากหลายและทางเลือกในการเลือกเส้นทางที่ดีที่สุด ไม่ว่าจะเป็นค่าเดินทาง ค่าน้ำมัน หรือตัวแปรต่างๆ ที่เกี่ยวข้องกับการใช้เส้นทาง ซึ่งค่าตัวแปรเหล่านี้สามารถเพิ่มประสิทธิภาพการทำงานของเครื่องมือนี้ได้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

การพัฒนาและปรับปรุงฟังก์ชัน Shortest path dijkstra นั้นเพื่อต้องการปรับปรุงฟังก์ชันให้เข้ากับฐานข้อมูลที่มีเพื่อให้เกิดประสิทธิภาพในการใช้งานสูงสุด จากการนำแนวทางการทำ Multiple path มาแก้ปัญหานั้น ทำให้การเลือกเส้นทางที่ดีที่สุดมีขอบเขตที่ชัดเจนและได้ค้นพบแนวทางการแก้ปัญหาเหล่านั้นไปเขียนโปรแกรมเป็น Application Programming Interface หรือ API เพื่อให้ผู้ที่ต้องการนำไปใช้ได้ตามประเภทของงานการจราจร

5.2 ข้อเสนอแนะ

จากการพัฒนาและปรับปรุงฟังก์ชันเพื่อให้ตรงตามการใช้งาน แต่ก็ยังสามารถจะพัฒนาต่อไปได้อีกด้วยการเพิ่มตัวแปรที่ใช้ในการคำนวณฟังก์ชันเพื่อให้มีความหลากหลายและทางเลือกในการเลือกเส้นทางที่ดีที่สุด ไม่ว่าจะเป็นค่าเดินทาง ค่าน้ำมัน หรือตัวแปรต่างๆที่เกี่ยวข้องกับการใช้เส้นทาง ซึ่งค่าตัวแปรเหล่านี้สามารถเพิ่มประสิทธิภาพการทำงานของเครื่องมือนี้ได้

เอกสารอ้างอิง

[0] โครงการประเมินและรายงานสภาพจราจร ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ. 2015. **ข้อมูลการจราจร**. กรุงเทพฯ สวทช.

ชาติ. 2015. **ข้อมูลการจราจร**. กรุงเทพฯ สวทช.

[1] Daniel Kastl and Frédéric Junod. Sep 06, 2014. **Installation and Requirements**.

[Online]. Available : <http://workshop.pgRouting.org/chapters/installation.html>.

[2] Daniel Kastl and Frédéric Junod. Sep 06, 2014. **Create a Network Topology**.

[Online]. Available : <http://workshop.pgRouting.org/chapters/topology.html>.

[3] Daniel Kastl and Frédéric Junod. Sep 06, 2014. **pgRouting Algorithms**. [Online].

Available : http://workshop.pgRouting.org/chapters/shortest_path.html.

[4] Daniel Kastl. Jul 21, 2014. **Shortest Path search in your Database and more with pgRouting**. [Online].

Available :

<http://www.slideshare.net/kastl/pg-routing-presentation-foss4g-europe>.

[5] Chingchai Humhong. 2014. **Free and Open Source GIS**. [Online].

Available : <http://nu-gis.blogspot.com/2014/08/pgRouting.html>

ภาคผนวก

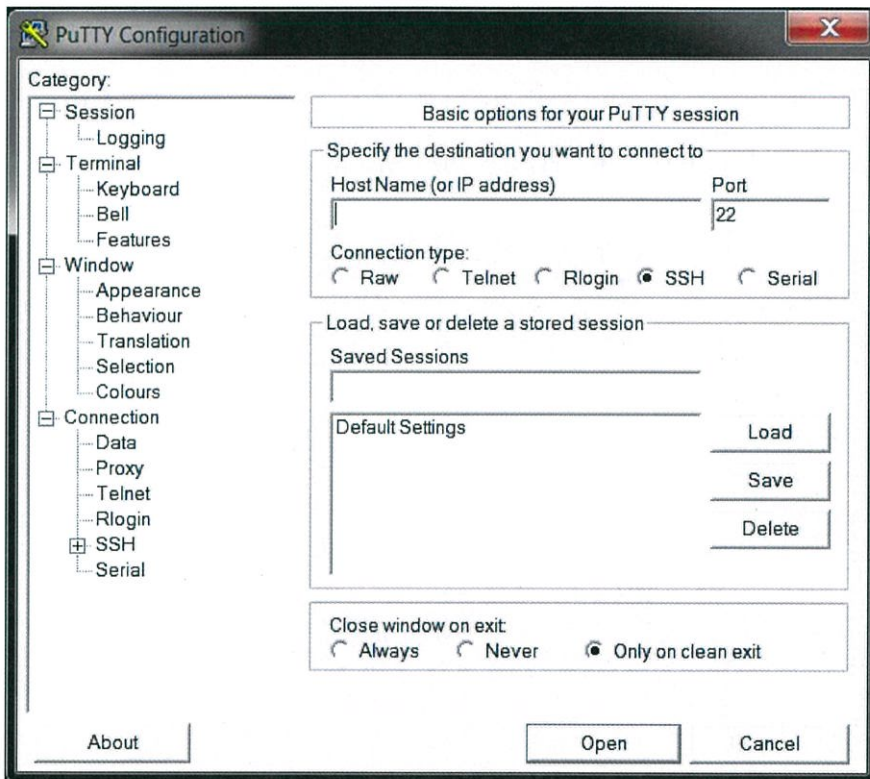
การนำไปใช้บน server

จากการวัดประสิทธิภาพและพัฒนาปรับปรุงฟังก์ชัน Shortest Path Dijkstra จะนำแนวทางข้างต้นมาเขียนเป็นโปรแกรมที่สามารถหาเส้นทางที่ดีที่สุด โดยจะนำไปใช้บน server และพัฒนาให้เป็น Application Programming Interface หรือ API เพื่อเป็นช่องทางหรือตัวกลางที่จะช่วยให้ผู้ใช้ด้านนอกสามารถดึงข้อมูลจากโปรแกรมนี้ออกไปใช้เป็นเครื่องมือในการใช้งานตามประเภทงานที่ต้องการ

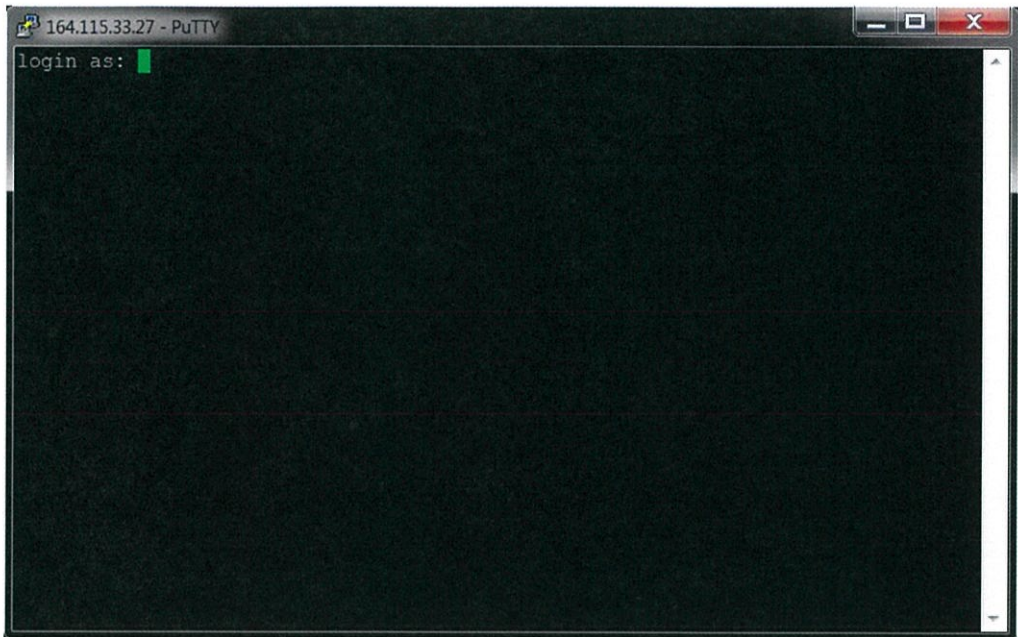
ขั้นตอนการนำไปใช้งานบน server

1. การติดตั้ง pgrouting บน server

ทำการติดตั้ง pgrouting บน server ผ่านโปรแกรม putty โดยจะต้องใส่ IP address ของ server ที่ต้องการใช้งาน ดังรูปที่ 1



รูปที่ 1 หน้าต่างโปรแกรม putty



รูปที่ 2 หน้าต่างโปรแกรม putty เมื่อเข้าสู่ server

2. ติดตั้งฟังก์ชัน Shortest path dijkstra บน server

ในติดตั้งฟังก์ชันบน server จะแตกต่างกับการติดตั้งบนเครื่องคอมพิวเตอร์ จะใช้คำสั่ง linux ในการติดตั้ง pgRouting พร้อมทั้งติดตั้งฟังก์ชัน Shortest Path Dijkstra บน server ดังในตารางต่อไปนี้

ตารางที่ 1 คำสั่งที่ใช้ในการติดตั้ง pgRouting บน server

```
sudo apt-add-repository -y ppa:ubuntugis/ppa
```

```
sudo apt-add-repository -y ppa:georepublic/pgRouting
```

```
sudo apt-get update
```

ตารางที่ 2 คำสั่งติดตั้ง pgRouting package (for Ubuntu 14.04)

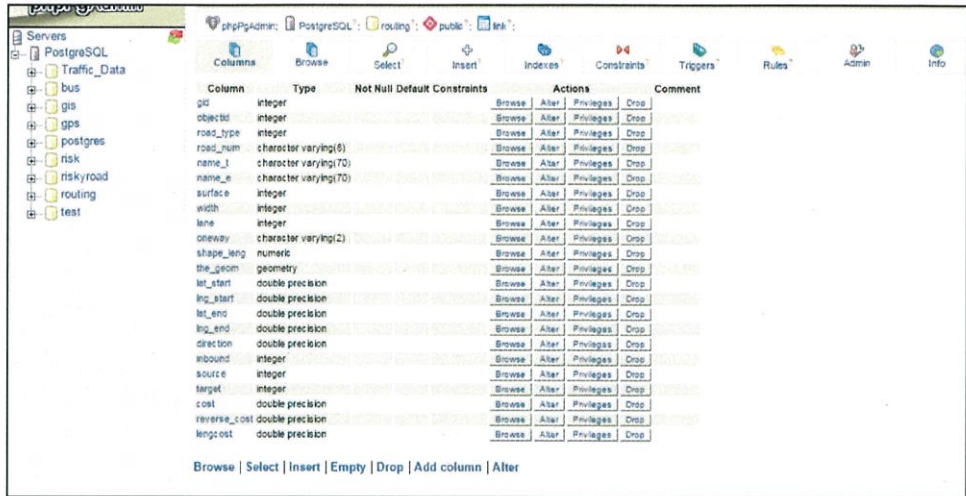
```
sudo apt-get install postgresql-9.3-pgRouting
```

ตารางที่ 3 คำสั่งติดตั้ง osm2pgRouting package

```
apt-get install osm2pgRouting
```

3. ขั้นตอนการเตรียมฐานข้อมูลบน server

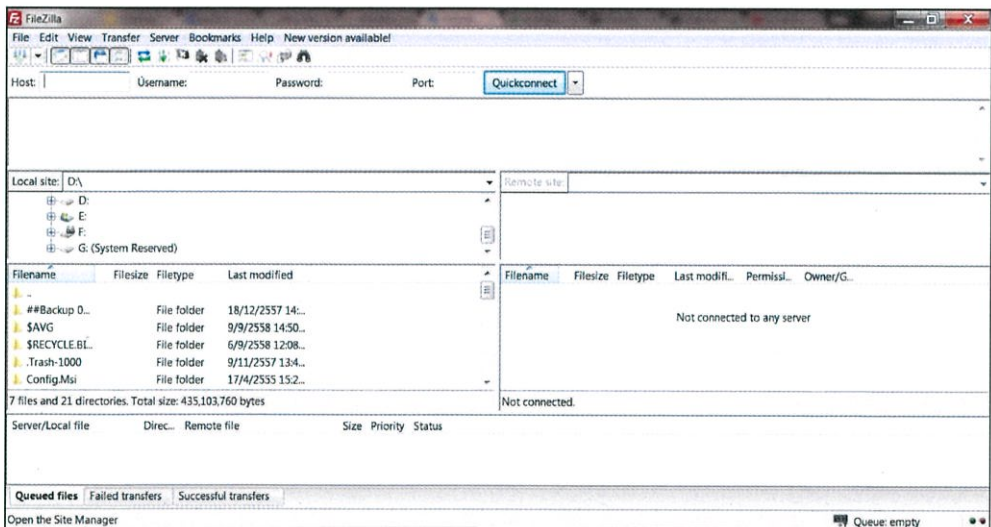
สร้างฐานข้อมูลขึ้นมาชื่อว่า Routing บน server โดยใช้ผ่าน pgAdmin ดังรูปที่ 3



Column	Type	Not Null	Default	Constraints	Actions	Comment
gid	integer				Browse Alter Privileges Drop	
objectid	integer				Browse Alter Privileges Drop	
road_type	integer				Browse Alter Privileges Drop	
road_num	character varying(6)				Browse Alter Privileges Drop	
name_1	character varying(70)				Browse Alter Privileges Drop	
name_2	character varying(70)				Browse Alter Privileges Drop	
surface	integer				Browse Alter Privileges Drop	
width	integer				Browse Alter Privileges Drop	
lane	integer				Browse Alter Privileges Drop	
oneway	character varying(2)				Browse Alter Privileges Drop	
shape_len	numeric				Browse Alter Privileges Drop	
the_geom	geometry				Browse Alter Privileges Drop	
int_start	double precision				Browse Alter Privileges Drop	
int_end	double precision				Browse Alter Privileges Drop	
direction	double precision				Browse Alter Privileges Drop	
inbound	integer				Browse Alter Privileges Drop	
source	integer				Browse Alter Privileges Drop	
target	integer				Browse Alter Privileges Drop	
cost	double precision				Browse Alter Privileges Drop	
reverse_cost	double precision				Browse Alter Privileges Drop	
lengthcost	double precision				Browse Alter Privileges Drop	

รูปที่ 3 องค์ประกอบของฐานข้อมูล routing

ข้อมูลที่จะนำมาใช้บน server จะเป็นไฟล์ csv ข้อมูลในไฟล์เหล่านี้จะถูกค้นด้วยเครื่องหมายจุดภาคเป็นชุดเฉพาะของข้อมูล จะทำการอัปโหลดไฟล์ข้อมูลโดยใช้โปรแกรม filezilar ขึ้นไปบน server ดังรูปที่ 4



รูปที่ 4 หน้าต่างโปรแกรม filezilar ที่ใช้ในการอัปโหลดข้อมูล

4. คำนวณฟังก์ชัน Shortest path dijkstra บน server

การคำนวณฟังก์ชัน Shortest path dijkstra บน server จะใช้คำสั่ง SQL ผ่าน pgAdmin ของ server ดังตารางคำสั่งต่อไปนี้

ตารางที่ 4 คำสั่งเพิ่มคอลัมน์ของ source และ target บน server

```
ALTER TABLE link ADD COLUMN "source" integer;
```

```
ALTER TABLE link ADD COLUMN "target" integer;
```

ตารางที่ 5 คำสั่งที่ใช้ในการคำนวณฟังก์ชันหรือเรียกว่าการสร้าง Topology บน server

```
SELECT pgr_createTopology ('routing', 0.00001, 'the_geom', 'gid');
```

ตารางที่ 6 คำสั่งสร้าง INDEX ของ source และ target

```
CREATE INDEX links_source_idx ON link ("source");
```

```
CREATE INDEX links_target_idx ON link ("target");
```

ตารางที่ 7 เพิ่มคอลัมน์ cost และ reverse_cost กำหนดให้ cost=1 และ reverse_cost=-1

```
ALTER TABLE links ADD COLUMN "cost" integer;
```

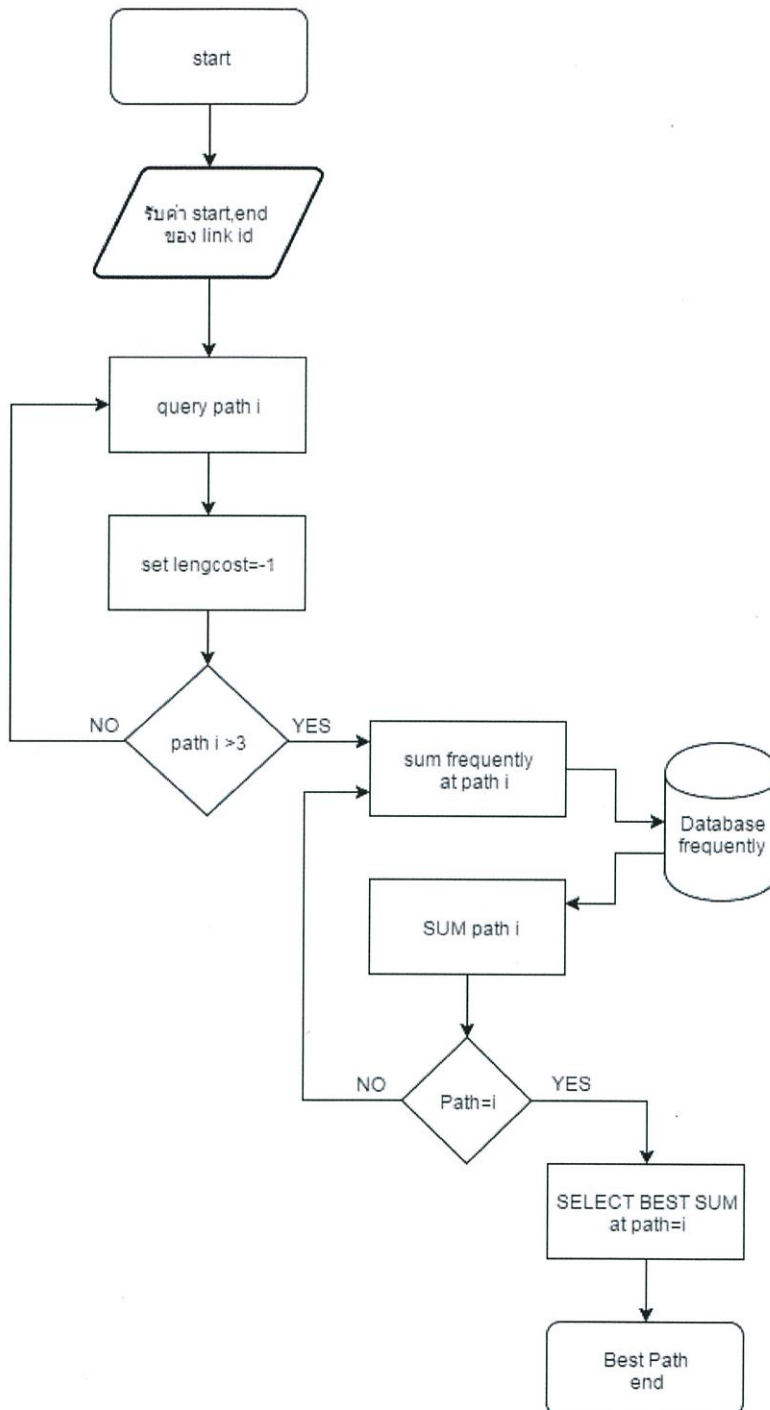
```
ALTER TABLE links ADD COLUMN "reverse_cost" integer;
```

```
UPDATE link SET cost = 1;
```

```
UPDATE link SET reverse_cost = -1;
```

5. เขียนโปรแกรมเป็น API

นำกระบวนการทำ Multiple path มาเขียนเป็นโปรแกรม API เพื่อให้ผู้ใช้สามารถนำฟังก์ชัน Shortest path dijkstra ที่ทำการพัฒนาปรับปรุงให้สอดคล้องตามความต้องการไปใช้ได้อย่างสะดวก กระบวนการทำงานของโปรแกรมมีดังรูปที่ 4.5



รูปที่ 4.5 flow chart การทำงานของโปรแกรม