

การสร้างและวิเคราะห์แบบจำลอง 3 มิติของทารกในครรภ์
จากข้อมูลเครื่องถ่ายภาพเหนือเสียง



นายกฤษฎิภาณ โสภาบุตร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2555

THREE – DIMENSIONAL MODEL RECONSTRUCTION OF FETAL ULTRASONOGRAPHY



Kritpanu Sopabutr

**A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Engineering in Computer Engineering**

Suranaree University of Technology

Academic Year 2012

การสร้างและวิเคราะห์แบบจำลอง 3 มิติของทารกในครรภ์
จากข้อมูลเครื่องถ่ายภาพเหนือเสียง

มหาวิทยาลัยเทคโนโลยีสุรนารี
ตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

อนุมัติให้บัณฑิตวิทยาลัยรับนี้เป็นส่วนหนึ่งของการศึกษา

คณะกรรมการสอบวิทยานิพนธ์

(รศ. ดร.กิตติศักดิ์ เกิดประสพ)

ประธานกรรมการ

(ผศ. ดร.ประเมศวร์ ห่อแก้ว)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

(ผศ. ดร.พิชโยทัย มัทธนาภิวัฒน์)

กรรมการ

(ศ. ดร.ชูกิจ ลิ้มปีจันทร์)

รองอธิการบดีฝ่ายวิชาการ

(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

กฤษฎีกาญ โสภานุตร : การสร้างและวิเคราะห์แบบจำลอง 3 มิติของทารกในครรภ์จาก
ข้อมูลเครื่องถ่ายภาพเหนือเสียง (THREE – DIMENSIONAL MODEL
RECONSTRUCTION OF FETAL ULTRASONOGRAPHY) อาจารย์ที่ปรึกษา :
ผู้ช่วยศาสตราจารย์ ดร.ปรเมศวร์ ห่อแก้ว, 84 หน้า.

ในอดีตเครื่องมือวินิจฉัยทางการแพทย์ เช่น เครื่องเอกซเรย์ เครื่อง CT-สแกน หรือเครื่อง
ถ่ายภาพเหนือเสียง (Ultrasound) ล้วนแต่แสดงผลลัพธ์ในรูปแบบของภาพ 2 มิติ สีขาว – ดำทั้งสิ้น
โดยเป็นภาพที่มีลักษณะค่อนข้างซับซ้อน จึงทำให้การอธิบายผลการวินิจฉัยของแพทย์ให้แก่ผู้ป่วย
และญาติเป็นไปอย่างลำบาก แต่ในปัจจุบันมีการนำข้อมูลทางการแพทย์ผสานกับเทคโนโลยี
คอมพิวเตอร์ สามารถแสดงผลลัพธ์ออกมาในรูปแบบของภาพ 3 มิติ ทำให้การตรวจสอบและ
วินิจฉัยโรคต่าง ๆ อวัยวะภายใน ความผิดปกติหรือเพศของทารกในครรภ์มีประสิทธิภาพมากขึ้น
แต่อย่างไรก็ตามเครื่องมือเหล่านี้ก็มีราคาค่อนข้างสูง จึงทำให้โรงพยาบาลที่มีขนาดเล็กไม่สามารถ
สั่งซื้อเครื่องมือเหล่านี้ได้ เพื่อแก้ปัญหาภายในงานวิจัยจึงนำความรู้ต่าง ๆ ด้าน Image Processing
และเทคนิคการเขียนโปรแกรม สร้างแบบจำลอง 3 มิติจากข้อมูลภาพทางการแพทย์ เพื่อง่ายต่อการ
วิเคราะห์พิจารณา และขณะเดียวกันยังเป็นการลดต้นทุนจากอุปกรณ์ที่มีอยู่แล้วอีกด้วย ซึ่งข้อมูล
ที่แสดงในงานวิจัยบางส่วนคือข้อมูลที่สร้างขึ้นเอง และข้อมูลของทารกในครรภ์

กระบวนการทำงานของระบบเริ่มจากการนำชุดข้อมูลของภาพ 2 มิติ มาทำการแยก
องค์ประกอบของภาพด้วย Greedy Active Contour หรือ Snake Active Contour เพื่อแยกวัตถุที่สนใจ
ออกจากภาพ ชุดข้อมูลคอนทัวร์ที่ได้มานี้จะใช้ในการสร้างแบบจำลอง 3 มิติ โดยการขยายพื้นผิวด้วย
ค่าจริงของฟังก์ชันทรงกลมแบบฮาร์โมนิก (Spherical harmonics) เพื่อความถูกต้องของแบบจำลอง 3
มิติที่ถูกสร้างขึ้น และเพิ่มประสิทธิภาพของระบบ หากบางส่วนของแบบจำลองมีความผิดพลาดหรือ
ภาพมีความคลาดเคลื่อน ระบบจะทำการปรับพื้นผิวใหม่เฉพาะส่วนที่มีความผิดพลาดเท่านั้น และจะ
ไม่ทำการสร้างพื้นผิวใหม่ทั้งหมด (Interactive Surface Editing)

KRITPANU SOPABUTR : THREE – DIMENSIONAL MODEL

RECONSTRUCTION OF FETAL ULTRASONOGRAPHY.

THESIS ADVISOR : ASST. PROF. PARAMATE HORKAEW, Ph.D., 84 PP.

IMAGE SEGMENTATION/ACTIVE CONTOUR/SURFACE RECONSTRUCTION/
SPHERICAL HARMONICS/INTERACTIVE SURFACE EDITING

In the past, diagnostic medical instruments such as X-Ray CT scanner and ultrasound were all monitoring in 2D (white-black). As such, it is relatively complicated for a doctor to communicate with their patients. Meanwhile, medical informatics and computer technology have been merged and able to display 3D anatomical models. As a result, internal organs, pathologies, abnormalities of the fetus or gender infant, for instances, could be monitored and diagnosed more efficient. These tools, however, come with a high price tag, prohibiting them from being acquired by smaller hospitals. To elevate this issue, this research adopting various image processing and Programming techniques to reconstruct a 3D model from multiple medical images, enhancing its visual comprehension, while maintaining the lower cost of conventional equipment. We also demonstrate our schemes with some synthetic data and in vivo fetal echocardiography.

This involving process consists of starts with a series of 2D segmentation using a greedy active contour or Snake to extract object of interest. A collection of segmented contours were then used to create respective 3D model using real value Spherical Harmonic expansion of a surface. Should more faithful surface was required, interactive surface editing was provided to efficiently correct object – image

discrepancies, after which the 3D model was locally adjusted without having to rebuild whole surface.



School of Computer Engineering

Academic Year 2012

Student's Signature _____

Advisor's Signature _____

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลงด้วยดี ผู้วิจัยขอกราบขอบพระคุณ บุคคล และกลุ่มบุคคลต่าง ๆ ที่ได้กรุณาให้คำปรึกษา แนะนำช่วยเหลืออย่างดียิ่ง ทั้งในด้านวิชาการ และด้านการดำเนินงานวิจัย ดังนี้

ผู้ช่วยศาสตราจารย์ ดร.ปรเมศวร์ ห่อแก้ว อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่มีเมตตาให้การอบรมสั่งสอน ชี้แนะ ช่วยเหลือในการทำการศึกษาวิจัย ตลอดจนให้คำแนะนำในการเขียน และตรวจแก้ไขวิทยานิพนธ์จนเสร็จสมบูรณ์

ดร.ดวงรัตน์ แก่นสวัสดิ์ นักวิจัยที่ปรึกษาโครงการทุนสถาบันบัณฑิตวิทยาศาสตร์และเทคโนโลยีไทย ที่มอบโอกาสดี ๆ รวมทั้งให้คำแนะนำทางด้านการวิชาการตลอดมา

อาจารย์ประจำสาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารีทุกท่าน ที่ให้การอบรมสั่งสอนและดูแลเสมอมา

คุณสมชาย สุขอินทร์ คุณวีรศักดิ์ ช่างหล่อ และบัณฑิตศึกษาสาขาวิศวกรรมคอมพิวเตอร์ทุกท่าน ที่ให้คำปรึกษา ดูแลและช่วยเหลือด้วยดีเสมอมา

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์เล่มนี้ ผู้วิจัยขอมอบให้กับบิดา มารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่าน ที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ผู้วิจัยตลอดมา จนทำให้ประสบความสำเร็จในชีวิต

กฤษฎิ์ภาณุ โสภานุตร

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย)	ก
บทคัดย่อ (ภาษาอังกฤษ)	ข
กิตติกรรมประกาศ	ง
สารบัญ	จ
สารบัญตาราง	ช
สารบัญรูป	ฉ
บทที่	
1 บทนำ	1
1.1 ความสำคัญและที่มาของปัญหาการวิจัย	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของโครงการ	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ และหน่วยงานที่จะนำผลการวิจัยไปใช้ประโยชน์	3
1.5 โครงสร้างวิทยานิพนธ์	4
2 ทัศนวิสัยการมองเห็นและงานวิจัยที่เกี่ยวข้อง	5
2.1 กล่าวนำ	5
2.2 การปรับพื้นผิวบางจุด (Interactive Surface Editing)	7
2.3 การแยกองค์ประกอบภาพ (Image Segmentation)	7
2.4 การสร้างพื้นผิวจากจุด (Surface Reconstruction from Point Clouds)	14
2.4.1 Neighborhoods	14
2.4.2 เวกเตอร์ตั้งฉาก (Normal Vectors)	15
2.4.2.1 Principal Component Analysis	16
2.4.2.2 วิธีกำลังสองน้อยที่สุดของระนาบสามมิติ (Plana Fitting 3D : Least square plan)	17

สารบัญ (ต่อ)

หน้า

2.4.3	Minimum spanning tree.....	19
2.4.3.1	Prim's Algorithm.....	19
2.4.3.2	Dijkstra algorithm.....	20
2.4.4	การสร้างรูปสามมิติ (Surface Reconstruction).....	23
2.5	การสร้างขยายจุดของพื้นผิวทรงกลม (Spherical Expansion).....	26
2.5.1	การขยายจุดของพื้นผิวทรงกลม (Spherical Expansion).....	27
2.6	สรุปงานวิจัยที่เกี่ยวข้อง.....	28
3	วิธีดำเนินการวิจัย.....	29
3.1	กล่าวนำ.....	29
3.2	ระเบียบวิธีวิจัย.....	30
3.3	ภาพรวมของระบบที่ทำการพัฒนา.....	31
3.3.1	เครื่องมือที่ใช้ในการพัฒนา.....	31
3.3.2	โครงสร้างของระบบ.....	32
3.4	การแยกองค์ประกอบของภาพด้วยแอคทีฟคอนทัวร์ (Active Contour).....	46
3.5	การสร้างพื้นผิวสามมิติด้วยจุด (Surface Reconstruction from point cloud) และการขยายพื้นผิวด้วยค่าจริงของฟังก์ชันทรงกลมแบบฮาร์โมนิก (Spherical Harmonics).....	50
3.5.1	การสร้างพื้นผิวสามมิติด้วยจุด.....	50
3.5.2	การขยายพื้นผิวด้วยค่าจริงของฟังก์ชันทรงกลมแบบฮาร์โมนิก.....	54
3.6	การปรับพื้นผิวเฉพาะบางจุด (Interactive Surface Editing).....	58
4	การทดสอบและผลการทดสอบ.....	61
4.1	กล่าวนำ.....	61
4.2	การวัดเวลาการสร้างแบบจำลองตามลำดับ.....	61
4.3	วัดค่าความผิดพลาด (Error) ของแบบจำลองสามมิติ.....	63
4.4	ทดสอบค่าความผิดพลาดจากผู้ทดสอบ 1 คนทำการทดสอบหลายครั้ง (Intra Observer Variability).....	67

สารบัญ (ต่อ)

หน้า

4.5	ทดสอบค่าความผิดพลาดจากผู้ทดสอบหลายคน ทำการทดสอบ คนละ 1 ครั้ง (Inter Observer Variability).....	69
5	สรุปผลการวิจัยและข้อเสนอแนะ.....	75
5.1	สรุปขั้นตอนการทำงานของระบบสร้างแบบจำลองสามมิติ.....	75
5.2	ข้อจำกัดของงานวิจัย.....	76
5.3	ข้อเสนอแนะในงานวิจัยต่อไป.....	77
	รายการอ้างอิง.....	78
	ภาคผนวก	
	ภาคผนวก ก. โค้ดจำลองการสร้างแบบจำลอง 3 มิติ (Pseudo code 3D surface reconstruction).....	80
	ประวัติผู้เขียน.....	84



สารบัญตาราง

ตารางที่	หน้า
2.1 ตารางการคำนวณ Dijkstra's Algorithm เริ่มต้น	21
2.2 ตารางแสดงการคำนวณของของน้ำหนักและ $d(v)$ รอบที่ 1	22
2.3 ตารางแสดงการคำนวณของของน้ำหนักและ $d(v)$ รอบที่ 6	23
2.4 ตารางแสดงการคำนวณของ Dijkstra's Algorithm เมื่อไปที่โหนด G	23
4.1 ตารางผลการทดสอบระบบด้วยเวลาประมวลผลการสร้างพื้นผิวแบบจำลอง	61
4.2 ตารางวัดค่าความผิดพลาดของการสร้างแบบจำลอง 3 มิติ	65
4.3 ตารางผลการทดสอบผู้ใช้ 1 คนทำการทดลอง 4 ครั้ง	67
4.4 ตารางผลการทดสอบจากผู้ใช้ 4 คน ทำการทดลองคนละ 1 ครั้ง	69



สารบัญรูป

รูปที่	หน้า
1.1 ลำดับขั้นตอนการสร้างแบบจำลองสามมิติ	2
2.1 โครงสร้างตาข่ายของการขยายผิวทรงกลม	5
2.2 วัตถุต้นแบบ	6
2.3 นำโครงสร้างทั้งสองมาซ้อนกัน	6
2.4 การเทมเพลตใช้ในการหาค่าความแตกต่างความเข้มแสงในภาพ	9
2.5 ขั้นตอนการหาขอบภาพ	9
2.6 ภาพต้นแบบ	10
2.7 หาขอบภาพด้วยโซเบล	10
2.8 ก กำหนดจุดบนคอนทัวร์	11
2.8 ข การเคลื่อนที่เข้าหาขอบ	11
2.9 ตำแหน่งและทิศทางของจุดบนคอนทัวร์	12
2.10 ทิศทางของจุดในแต่ละรอบ	12
2.11 จุดศูนย์กลางแต่ละรอบ	12
2.12 รูปแบบที่ 1 ของการค้นหา	13
2.13 รูปแบบที่ 2 ของการค้นหา	13
2.14 กำหนดจุดบนคอนทัวร์โดยผู้ใช้	14
2.15 ระบบทำการคำนวณหาคอนทัวร์ของวัตถุ	14
2.16 Neighborhoods	15
2.17 Normal Vector	15
2.18 เวกเตอร์ตั้งฉากของแต่ละ Neighborhood	18
2.19 กราฟตั้งต้น	19
2.20 รูปที่คำนวณได้จาก Prim	20
2.21 แสดงตัวอย่าง Dijkstra's Algorithm	21
2.22 แสดงการค้นหาจาก A ไป B	22
2.23 แสดงการค้นหาจาก F ไป G	22

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.24 จุคบนระนาบแต่ละระนาบ	24
2.25 รูปแบบการสร้างภาพสามมิติ	24
2.26 การใส่ตัวชี้	25
2.27 รูปการสร้างจุดบนพื้นผิว	25
2.28 การแยกองค์ประกอบของภาพในขั้นตอนการหาคอนทัวร์	25
2.29 การแยกองค์ประกอบของภาพในขั้นตอนการหาเวกเตอร์ตั้งฉาก	26
2.30 การสร้างแบบจำลองสามมิติจากจุดด้วยการสร้างแบบลูกบาศก์	26
2.31 จำนวนรอบของแต่ละ order	28
3.1 ก การแยกองค์ประกอบของภาพในขั้นตอนการหาคอนทัวร์	29
3.1 ข การแยกองค์ประกอบของภาพในขั้นตอนการหาเวกเตอร์ตั้งฉาก	30
3.2 การซ้อนทับกันของคอนทัวร์และแบบจำลอง	30
3.3 ระบบแยกองค์ประกอบภาพ	32
3.4 Use case diagram ของระบบ	33
3.5 คลาสที่ใช้สำหรับจัดการข้อมูลรูป	34
3.6 OpenFileDialog	34
3.7 เมนูการหาองค์ประกอบของภาพและสร้างแบบจำลองสามมิติ	35
3.8 คลาสสำหรับการแยกองค์ประกอบของภาพ	35
3.9 ตัวอย่างภาพการหาคอนทัวร์ที่ได้จากการเพิ่มและปรับจุด	36
3.10 คลาสสำหรับจัดการการเก็บข้อมูลต่าง ๆ บนคอนทัวร์	36
3.11 ปุ่มที่สร้างจากคลาสต่าง ๆ	37
3.12 คลาสสำหรับจัดการในส่วนองแบบจำลองสามมิติ	37
3.13 หาคอนทัวร์เสร็จสิ้นระบบจะสร้างปุ่ม Create Model	38
3.14 สร้างแบบจำลองสามมิติ	39
3.15 ไม่มีการแก้ไขใด ๆ บนระบบ	40
3.16 มีการแก้ไขจุดบนคอนทัวร์ที่ทำได้	41
3.17 มีการลบจุดทั้งหมดบนคอนทัวร์ออก	42

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.18 มีการแก้ไขตัวแบบจำลองสามมิติ.....	43
3.19 AddPoint.....	45
3.20 ข้อความบอกผู้ใช้งานที่สามารถหาคอนทัวร์ได้แล้ว พร้อมกับจำนวนจุดข้อมูล.....	45
3.21 มีข้อมูลที่ไม่ยึดติดกับขอบรูป.....	45
3.22 แก้ไขข้อมูลจากปุ่ม Editpoint.....	46
3.23 ผลลัพธ์การรันโปรแกรม.....	46
3.24 รูปทรงปกติ.....	47
3.25 ลักษณะของภาพที่มีโค้งเว้าลึก.....	48
3.26 ภาพต้นแบบ.....	48
3.27 หาขอบด้วยโซเบล (Sobel).....	48
3.28 หาขอบด้วย (Laplacian).....	49
3.29 ภาพต้นแบบ (Canny).....	49
3.30 คอนทัวร์ของภาพ.....	51
3.31 Neighborhood.....	51
3.32 เวกเตอร์ตั้งฉาก PCA.....	52
3.33 เวกเตอร์ตั้งฉาก Least Square.....	52
3.34 การสร้างพื้นผิวด้วยลูกบาศก์.....	53
3.35 หาจุดตัดของพื้นผิวบนลูกบาศก์.....	53
3.36 การสร้างพื้นผิวด้วยลูกบาศก์.....	54
3.37 การแปลงพิกัดทรงกลมและพิกัดฉาก.....	55
3.38 คอนทัวร์ 3 มิติ.....	55
3.39 ก พื้นผิวลำดับที่ 2.....	57
3.39 ข โครงตาข่ายลำดับที่ 2.....	57
3.40 พื้นผิวลำดับที่ 5.....	58
3.41 พื้นผิวลำดับที่ 10.....	58
3.42 แผนภาพการทำงานของขั้นตอนการปรับพื้นผิวบางจุดหรือบางบริเวณ.....	59

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.43 โคร่งดาข่ายรูปทรง.....	59
3.44 แก้วไขคอนทัวร์.....	59
3.45 แบบจำลองสามมิติเดิม.....	60
3.46 แบบจำลองสามมิติที่ถูกสร้างใหม่.....	60
4.1 กราฟเวลาประมวผลการสร้างพื้นผิวรูปทรงกลม.....	62
4.2 กราฟเวลาประมวผลการสร้างพื้นผิวรูปฟักทอง.....	62
4.3 กราฟเวลาประมวผลการสร้างพื้นผิวรูปทารก.....	62
4.4 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติทรงกลม.....	66
4.5 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติรูปฟักทอง.....	66
4.6 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติรูปทารก.....	66
4.7 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติทรงกลม.....	68
4.8 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติรูปฟักทอง.....	68
4.9 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติรูปทารก.....	68
4.10 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติทรงกลม.....	70
4.11 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติรูปฟักทอง.....	70
4.12 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติรูปทารก.....	70
4.13 แบบจำลองสามมิติของรูปทรงต่าง ๆ.....	72
4.14 แบบจำลองลำดับที่ 2.....	73
4.15 แบบจำลองลำดับที่ 5.....	73
4.16 การซ้อนทับของแบบจำลองลำดับที่ 2.....	73
4.17 การซ้อนทับของแบบจำลองลำดับที่ 3.....	74

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหาการวิจัย

เนื่องจากทุกวันนี้วิวัฒนาการของเทคโนโลยีทางการแพทย์และคอมพิวเตอร์ได้ถูกพัฒนาไปอย่างรวดเร็ว ทำให้การตรวจอวัยวะภายในเพื่อวินิจฉัยหาโรคของผู้ป่วย หรือการดูเศษ ความผิดปกติของทารกในครรภ์มารดา รวมไปถึงการติดตามผลการรักษาทำได้มีประสิทธิภาพมากขึ้น โดยแต่เดิมเครื่องมือที่ใช้ในการตรวจวินิจฉัยข้อมูลดังกล่าว ไม่ว่าจะเป็นเครื่องเอกซเรย์ เครื่อง CT-สแกน หรือถ่ายภาพเหนือเสียง (Ultrasonography) ส่วนมากแสดงผลลัพธ์ออกมาในรูปแบบของภาพ 2 มิติ สีขาว-ดำเท่านั้น ทำให้ผู้ป่วย หรือญาติรวมทั้งแพทย์ผู้วินิจฉัยโรคที่ยังไม่ค่อยมีประสบการณ์ทำการวินิจฉัยโรคจากภาพเหล่านี้ได้ค่อนข้างลำบาก และอาจทำให้ความถูกต้อง แม่นยำในการวินิจฉัยโรคผิดพลาดได้อีกด้วย แต่ปัจจุบันนี้ด้วยเทคโนโลยีทางคอมพิวเตอร์ที่พัฒนาไปอย่างรวดเร็ว ทำให้มีการสร้างเครื่องมือ ซึ่งแต่เดิมเป็นเพียงการแสดงผลภาพ 2 มิติ มาเป็นภาพเคลื่อนไหวแบบ Real Time และจนมาเป็นการแสดงผลภาพแบบ 3 มิติ ซึ่งช่วยให้การวินิจฉัยมีประสิทธิภาพมากยิ่งขึ้น โดยเฉพาะอย่างยิ่งในการตรวจหาความผิดปกติของทารกในครรภ์ หรือการตรวจหาความผิดปกติของอวัยวะต่าง ๆ ภายในช่องท้อง แต่ยังคงมีความสะดวกมากขึ้นเท่าไร ต้นทุนเองก็ย่อมต้องสูงไปด้วยเช่นกัน ดังนั้นถ้าไม่ใช่โรงพยาบาลของรัฐที่มีขนาดใหญ่ หรือโรงพยาบาลของเอกชนที่มีรายได้ค่อนข้างสูง ก็ไม่สามารถที่จะสั่งซื้อเครื่องหรือนำเข้าเครื่องมือเหล่านี้จากต่างประเทศที่ยังคงมีราคาสูงกว่าเครื่อง Ultrasonography ซึ่งมีใช้อยู่แล้วโดยทั่วไปในโรงพยาบาลได้ จึงเกิดเป็นความสำคัญและที่มาของงานวิจัยเล่มนี้

จากปัญหาดังกล่าว จึงได้เกิดแนวคิดที่จะพัฒนาโปรแกรมโดยอาศัยเทคนิคทางด้านการประมวลผลภาพ และคอมพิวเตอร์กราฟิกประกอบกับข้อมูลจากแพทย์ผู้เชี่ยวชาญ มาใช้กับภาพ Ultrasound ที่มีอยู่แล้วในรูปแบบ 2 มิติ จากนั้นอาศัยการแยกองค์ประกอบของภาพ (Image Segmentation) มาทำการขึ้นรูปเป็นภาพ 3 มิติ (Surface reconstruction) ซึ่งมีลำดับขั้นตอนการพัฒนา ดังรูปที่ 1.1



รูปที่ 1.1 ลำดับขั้นตอนการสร้างแบบจำลองสามมิติ

จากรูปที่ 1.1 เป็นการแสดงขั้นตอนต่าง ๆ ในงานวิจัยชิ้นนี้ โดยส่วนแรกจะเริ่มจากการนำข้อมูลดั้งต้นหรือภาพ Ultrasound มาทำการหาขอบเขตวัตถุที่สนใจ เช่น รูปทารก ผู้ใช้จะต้องทำการวาดขอบเขตโดยคร่าว ๆ รอบวัตถุนั้นหรือทารกในภาพ จากนั้นระบบจะทำการหาขอบเขตของวัตถุต่อไป และสร้างเป็นภาพจำลองสามมิติขึ้นมาเอง ซึ่งเรียกว่าระบบคอมพิวเตอร์กึ่งอัตโนมัติ (Semi-Automatic) จะช่วยให้แพทย์ผู้เชี่ยวชาญสามารถวินิจฉัยความผิดปกติของทารกในครรภ์ หรืออวัยวะภายในของทารกได้อย่างถูกต้องและแม่นยำยิ่งขึ้น รวมไปถึงการลดต้นทุนการสั่งซื้อเครื่องมือที่มีราคาแพงจากต่างประเทศ เช่น Ultrasound และ MRI

1.2 วัตถุประสงค์ของงานวิจัย

งานวิจัยนี้มีวัตถุประสงค์หลักเพื่อศึกษาและพัฒนาระบบแสดงผลภาพจากเครื่อง Ultrasound ซึ่งแต่เดิมเป็นเพียงการแสดงผลแบบ 2 มิติ มาเป็นการแสดงผลภาพแบบ 3 มิติ ซึ่งแบ่งย่อยได้ดังนี้

1.2.1 เกิดการบูรณาการทางด้านวิศวกรรมและด้านการแพทย์และสาธารณสุข เพื่อพัฒนาโปรแกรมเป็นประโยชน์สำหรับการแพทย์ด้านเวชศาสตร์ Ultrasound และ MRI

1.2.2 ลดการนำเข้าเครื่อง Ultrasound แบบใหม่จากต่างประเทศที่มีราคาสูงกว่าเครื่อง Ultrasound ที่มีอยู่แล้ว

1.2.3 เพิ่มความถูกต้อง แม่นยำในการวินิจฉัยโรคของแพทย์ผู้เชี่ยวชาญในการวินิจฉัยโรค

1.2.4 เพื่อเป็นต้นแบบแก่ผู้ที่พัฒนาต่อไป

1.3 ขอบเขตของโครงการ

งานวิจัยทางการแพทย์หลายชิ้นได้รายงานกระบวนการในการตรวจสอบความผิดปกติของทารกในครรภ์ โดยใช้เทคโนโลยีการสร้างภาพแบบจำลอง 3 มิติ จาก Ultrasonography ด้วยวิธีการนี้แพทย์สามารถระบุความผิดปกติของอวัยวะของทารก ได้ดีขึ้นกว่าวิธีการ 2 มิติแบบดั้งเดิม อย่างไรก็ตามในปัจจุบัน การวินิจฉัยอาการผิดปกติดังกล่าว ยังอาศัยเพียงการมองภาพ 3 มิติของผู้เชี่ยวชาญ (Radiologist) ด้วยตาเปล่าซึ่งอาจก่อให้เกิดการเบี่ยงเบนระหว่างผู้สังเกตและภายในผู้สังเกตคนเดียวกัน (Inter- and Intra- observer Variability) งานวิจัยชิ้นนี้จึงมีวัตถุประสงค์เพื่อแก้ไขข้อจำกัดดังกล่าว โดยใช้เทคนิคการสร้างแบบจำลองเสริมเสมือนจริง (Augmented Virtual Reality) ซ้อนทับลงบนข้อมูลภาพ 3 มิติเดิม เพื่อให้แพทย์ผู้เชี่ยวชาญสามารถจำแนกขอบเขต และลักษณะโครงสร้างของอวัยวะได้เที่ยงตรง และแม่นยำขึ้น สำหรับขั้นตอนวิธีที่จะใช้ร่วมงานวิจัยแจกแจงได้ดังต่อไปนี้

- 1.3.1 ศึกษาการสร้างภาพ 3 มิติ งานวิจัยที่เกี่ยวข้อง และขั้นตอนการสร้างแบบจำลอง 3 มิติ
- 1.3.2 พัฒนาระบบขั้นตอนวิธีการสร้างแบบจำลอง 3 มิติ แบบกึ่งอัตโนมัติ
- 1.3.3 พัฒนาระบบวิธีวิเคราะห์คุณลักษณะของโครงสร้างอวัยวะใน 3 มิติ
- 1.3.4 พัฒนาระบบเพื่อแสดงโครงสร้างอวัยวะในรูปแบบ 3 มิติ และสามารถปรับปรุงแก้ไขเพียงบางส่วนได้ เพื่อความรวดเร็วในการแสดงผลของโมเดล
- 1.3.5 ทดสอบระบบพัฒนาขึ้นในการวิเคราะห์โครงสร้างสามมิติของวัตถุในภาพ
- 1.3.6 สรุปผลโครงการวิจัยและจัดทำวิทยานิพนธ์

1.4 ประโยชน์ที่คาดว่าจะได้รับ และหน่วยงานที่จะนำผลการวิจัยไปใช้ประโยชน์

โครงการวิจัยที่เสนอเป็นงานวิจัยสำหรับสนับสนุนงานทางการแพทย์ ผลลัพธ์ที่ได้จะอยู่ในลักษณะของโปรแกรมต้นแบบซึ่งเป็นการนำภาพจากเครื่อง Ultrasound แบบ 2 มิติ มาทำการจำลองเป็นแบบ 3 มิติ แบบ real-time เพื่อใช้เพิ่มประสิทธิภาพในการวินิจฉัยความผิดปกติของทารกในครรภ์ ให้มีความถูกต้องและแม่นยำยิ่งขึ้น สำหรับประโยชน์ที่คาดว่าจะได้รับคือ

- 1.4.1 เกิดการบูรณาการทางด้านวิศวกรรมและด้านการแพทย์และสาธารณสุข เพื่อพัฒนาโปรแกรมที่เป็นประโยชน์สำหรับการแพทย์ด้านเวชศาสตร์ทารกในครรภ์
- 1.4.2 ลดการนำเข้าเครื่อง Ultrasound แบบใหม่จากต่างประเทศที่มีราคาสูงกว่าเครื่อง Ultrasound ที่มีอยู่แล้ว

โดยสรุปหน่วยงานที่สามารถนำผลการวิจัยไปใช้ประโยชน์ สามารถแจกแจงได้ดังต่อไปนี้

- 1) โรงพยาบาล สถาบัน หน่วยงานด้านสาธารณสุข และวิทยาศาสตร์สุขภาพ
- 2) มหาวิทยาลัย และสถาบันการศึกษา ที่เปิดสอนด้านเวชศาสตร์ทหารกในครรภ์ หรือสาขาใกล้เคียง
- 3) คลินิกเวชศาสตร์ทหารกในครรภ์ หรือสาขาใกล้เคียง

1.5 โครงสร้างวิทยานิพนธ์

รายละเอียดเกี่ยวกับงานวิจัยทั้งหมดจะกล่าวถึงในบทถัดไปดังนี้ บทที่ 2 จะเป็นการกล่าวถึงวรรณกรรมและรายละเอียดงานวิจัยที่เกี่ยวข้องกับงานวิจัยนี้ โดยจะกล่าวถึงขั้นตอนวิธีการแยกองค์ประกอบของภาพ ขั้นตอนการจำลองภาพและพื้นผิวสามมิติ วิธีการขยายจุดของพื้นผิวของทรงกลมแบบฮาร์โมนิก และวิธีการซ้อนทับระหว่างพื้นผิว บทที่ 3 จะเป็นการกล่าวถึงรายละเอียดขั้นตอนในการดำเนินงานวิจัย โดยแบ่งออกเป็น ขั้นตอนการแยกองค์ประกอบของภาพ ขั้นตอนการสร้างพื้นผิวโดยขยายด้วยค่าจริงของฟังก์ชันทรงกลมแบบฮาร์โมนิก บทที่ 4 เป็นผลการทดสอบขั้นตอนวิธีทั้งหมดที่กล่าวมา โดยจะแสดงการทดสอบในรูปแบบของรูปภาพ ตารางหรือกราฟ บทที่ 5 เป็นการสรุปและอภิปรายผลการทดลอง ปัญหาและอุปสรรคที่พบ และข้อเสนอแนะในการวิจัยต่อไป

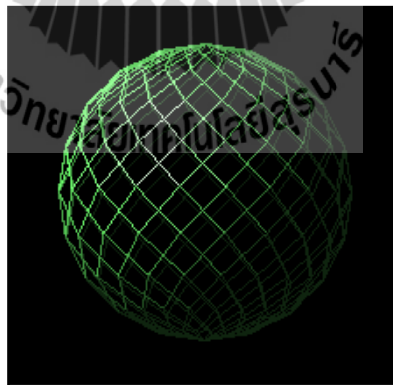


บทที่ 2

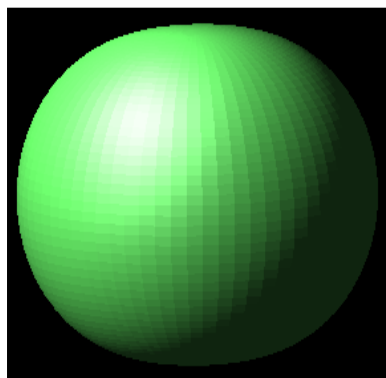
ปรัทัศนัวรรณกรรมและงานวิจัยที่เกี่ยวข้อง

2.1 กล่าวนำ

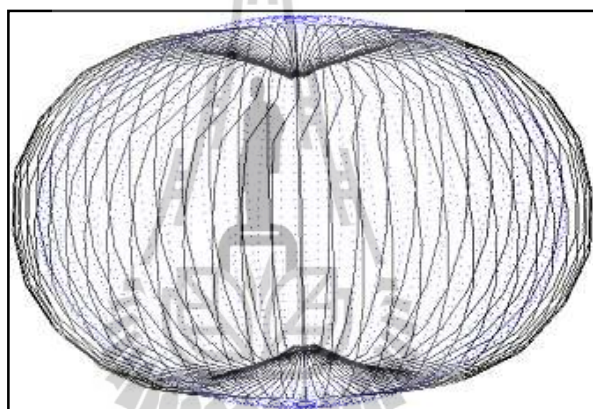
วิทยานิพนธ์ได้นำเสนอเทคนิคขั้นตอนการนำภาพจากเครื่องถ่ายภาพเหนือเสียง หรือเครื่อง Ultrasound ในรูปแบบของภาพสองมิติ มาสร้างเป็นภาพจำลองสามมิติแบบ real-time ประกอบไปด้วย หัวข้อที่ 2.2 กล่าวถึงการปรับจุดบางจุดบนพื้นผิว (Interactive Surface Editing) เพื่อให้สามารถสร้างภาพจำลองได้อย่างรวดเร็วกว่าการปรับจุดทั้งหมด หัวข้อที่ 2.3 คือการแยกองค์ประกอบของภาพ (Image Segmentation) เป็นการนำภาพปกติทั่วไป หรือภาพที่ได้จากเครื่อง Ultrasound มาทำการแยกวัตถุที่สนใจออกจากองค์ประกอบทั้งหมดภายในภาพ ด้วยวิธีการกึ่งอัตโนมัติ เป็นการที่ผู้ใช้งานต้องกำหนดค่าเริ่มต้นให้แก่ระบบก่อน จากนั้นระบบจึงจะทำการคำนวณค่าให้โดยอัตโนมัติ หัวข้อที่ 2.4 การสร้างพื้นผิวจากกลุ่มจุด (Surface Reconstruction from Point Clouds) หัวข้อที่ 2.5 การสร้างพื้นผิวขยายจุดของพื้นผิวทรงกลม (Spherical Expansion) และหัวข้อที่ 2.6 กล่าวถึงงานวิจัยอื่น ๆ ที่เกี่ยวข้องกับวิทยานิพนธ์



รูปที่ 2.1 โครงสร้างตาข่ายของการขยายผิวทรงกลม



รูปที่ 2.2 วัตถุต้นแบบ



รูปที่ 2.3 นำโครงสร้างทั้งสองมาซ้อนกัน

จากรูปที่ 2.1 จะเห็นโครงสร้างตาข่ายจากขั้นตอนการแยกองค์ประกอบของภาพ หรือคอนทัวร์ รูปที่ 2.2 คือพื้นผิวที่ถูกสร้างขึ้นจากการขยายจุดของพื้นผิวทรงกลม เมื่อนำทั้ง 2 ขั้นตอนมารวมกัน คือการเอาโครงสร้างของวัตถุจริง กับโครงสร้างของวัตถุที่เกิดจากการขยายจุดของพื้นผิวทรงกลมมาซ้อนทับกันดังรูปที่ 2.3 ทำให้ทราบตำแหน่งจริงของวัตถุ ฉะนั้นเมื่อเกิดการผิดพลาดบนพื้นผิว หรือต้องการปรับพื้นผิวของวัตถุใหม่ ก็สามารถปรับที่ตำแหน่งนั้นได้ทันที ไม่ต้องสร้างแบบจำลองขึ้นมาใหม่ทั้งหมด เพื่อเป็นการลดระยะเวลาการประมวลผลนั่นเอง

2.2 การปรับพื้นผิวบางจุด (Interactive Surface Editing)

เนื่องจากบางครั้งแบบจำลองสามมิติที่ได้มาอาจไม่ตรงกับตัวต้นแบบ ดังนั้นจึงต้องมีการปรับพื้นผิวในจุดที่ไม่ตรง แต่ปัญหาคือ การประมวลผลภาพขึ้นมาใหม่ในทุกครั้งต้องใช้เวลา และทรัพยากรของเครื่องค่อนข้างสูง บางครั้งเครื่องอาจค้างไป ดังนั้นมีวิธีอะไรที่จะช่วยแก้ไขปัญหานี้ จุดนี้ คำตอบคือ เราสามารถปรับพื้นผิวเพียงบางจุดได้ เมื่อเราทำการสร้างแบบจำลองสามมิติออกมา ซึ่งปกติแล้วพื้นผิวเหล่านี้จะเกิดจากข้อมูลต้นแบบซึ่งมีลักษณะเป็นจุดคล้ายรูปทรงของวัตถุที่เราสนใจ บางครั้งอาจมองว่าเป็นโครงข่ายของจุดจำนวนมากก็ได้ (Yujian Gao, Aimin Hao, Qinpeng Zhao, Neil A. Dodgson, 2009) แต่ยังไม่มีการสร้างพื้นผิว ด้วยลักษณะดังกล่าว จึงสามารถที่จะทราบได้ว่าพื้นผิวบริเวณที่ต้องการเปลี่ยนแปลงนั้นเกิดมาจากจุดบริเวณไหน เมื่อทราบว่าเป็นจุดใดก็สามารถกำหนดหาพื้นผิวขึ้นมาใหม่บริเวณนั้นได้ จึงไม่ต้องทำการสร้างขึ้นมาใหม่ทั้งหมด ช่วยลดระยะเวลาในการประมวลผลด้วย

2.3 การแยกองค์ประกอบภาพ (Image Segmentation)

เป็นวิธีการแบ่งส่วนใดส่วนหนึ่งของภาพที่สนใจออกมาจากภาพที่ต้องการ ซึ่งการแบ่งส่วนภาพนี้โดยส่วนใหญ่แล้วจะเป็นขั้นตอนเบื้องต้นและสำคัญอย่างมากของการประมวลผลภาพทางการแพทย์ เนื่องจากภาพทางการแพทย์ที่ได้จากเครื่องถ่ายภาพแบบต่าง ๆ นั้น โดยปกติมักจะมีองค์ประกอบอื่น ๆ ที่อยู่ใกล้เคียงกับอวัยวะที่ทำถ่ายภาพมา เช่น เนื้อเยื่อ กระดูก อวัยวะข้างเคียง หรือแม้กระทั่งสัญญาณรบกวน (Noise) ที่เกิดขึ้นในขณะถ่ายภาพ (Hill, C. R., Bamber, J. C., and ter Haar, G. R. Physical principles of medical ultrasonics, 2004) ด้วยเหตุนี้การวิเคราะห์เฉพาะอวัยวะที่ต้องการ จึงจำเป็นต้องใช้การแบ่งส่วนภาพมาทำหน้าที่ตัดแยกส่วนที่เราต้องการออกมา ตัวอย่างเช่น การแบ่งส่วนเนื้อสมองจากภาพสมอง การแบ่งส่วนภาพหัวใจห้องล่างซ้ายจากภาพหัวใจ MRI การแบ่งส่วนเฉพาะเส้นโลหิต การแบ่งส่วนข้อกระดูกสันหลังจากภาพลำกระดูกสันหลัง (Julio Carballido-Gamio, Serge J. Belongie, and Sharmila Majumdar, 2004) หรือ การแบ่งส่วนของทารกจาก Ultrasound เป็นต้น การแบ่งส่วนภาพทางการแพทย์มีทั้งการแบ่งส่วนภาพแบบ 2 มิติ และ 3 มิติ ขึ้นอยู่ความจำเป็นและวัตถุประสงค์ของการนำไปวิเคราะห์ โดยวิธีการแบ่งส่วนภาพที่กำลังได้รับความนิยมในงานวิจัยเกี่ยวกับภาพทางการแพทย์ ได้แก่ แอ็กทีฟคอนทัวร์ (Active Contour) เป็นต้น สำหรับการทำคอนทัวร์ หรือการแยกองค์ประกอบของภาพเป็นขั้นตอนที่มีความสำคัญมากที่สุดหนึ่งขั้นตอนในการวิเคราะห์รูปภาพ (Marián Bakoš, 2007) ซึ่งเป้าหมายหลักของมันคือการแบ่งรูปออกเป็น ส่วน ๆ (Image segmentation) ตามที่ได้กล่าวมาแล้ว เพื่อให้ได้ส่วนที่ต้องการหรือสนใจที่จะนำมาวิเคราะห์ ส่วนใหญ่ทำได้ยากเนื่องจากมีความแปรปรวนค่อนข้างมากของรูปร่างวัตถุเอง ความแตกต่างของคุณภาพของรูปภาพ บ่อยครั้งที่รูปภาพมีการรบกวนของ

สัญญาณ และมีส่วนที่มันเกินจำเป็นที่เราไม่ต้องการ ซึ่งก็เป็นปัญหาใหญ่และเป็นเรื่องปกติของการใช้เทคนิคการแยกองค์ประกอบของภาพ การแยกองค์ประกอบของภาพด้วยวิธีนี้ถูกนำมาใช้ในการแก้ปัญหาเหล่านี้โดยใช้กันอย่างกว้างขวาง จนกลายมาเป็นที่นิยมซึ่งรูปจักกันในเรื่องของ Snakes: Active contour models (Micheal Kass, Andrew Witkin and Demetri Tozopoulos, 1988) โดยพื้นฐานของ Snake model จะใช้การลดลงของพลังงานของรูปภาพแบบต่อเนื่องภายใต้อิทธิพลแรงของภาพ (image forces) และแรงจากภายนอก (external constraint forces) ตามสมการ

$$E_{snake}^* = \int_0^1 E_{snake}(v(s))ds \quad (2-1)$$

$$E_{snake}^* = \int_0^1 (E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))) ds \quad (2-2)$$

โดย E_{int} คือ พลังงานภายในรูปภาพเอง
 E_{image} คือ การเปลี่ยนแปลงของแรงจากภาพ เช่น ขอบของรูป เส้นขอบ ความสว่าง
 E_{con} คือ การเปลี่ยนแปลงของแรงจากภายนอก เช่น แรงดึงของเส้นเชื่อมจุด

สมการของพลังงานภายในสามารถเขียนได้ดังนี้

$$E_{int} = (\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}|^2)/2 \quad (2-3)$$

พลังงานจะประกอบไปด้วย 2 เทอมด้วยกัน เทอมแรกคือ $\alpha(s)$ และเทอมที่ 2 คือ $\beta(s)$ เทอมแรกหรือ $\alpha(s)$ ทำหน้าที่เข้าจับขอบวัตถุคล้ายเยื่อหุ้ม และเทอมที่สอง $\beta(s)$ ทำหน้าที่ควบคุมส่วนโค้งเว้าลึก-ตื้นของวัตถุ เช่น เมื่อกำหนดให้ $\beta = 0$ ณ จุดใด ๆ จะทำให้ขอบวัตถุไม่ต่อเนื่องและเป็นสาเหตุทำให้เกิดการหักมุม เป็นต้น

สำหรับในส่วนของ E_{image} เพื่อให้ง่ายต่อการใช้งาน จึงต้องสร้างฟังก์ชันเฉพาะของแต่ละเทอมของพลังงานโดยทำการพิจารณาจากคุณลักษณะที่สำคัญของภาพ ซึ่งจะพิจารณาในส่วนขอบ, เส้น และ เทอมอื่น ๆ เขียนเป็นฟังก์ชันได้ดังนี้

$$E_{image} = W_{line}E_{line} + W_{edge}E_{edge} + W_{term}E_{term} \quad (2-4)$$

ในส่วนของ E_{image} ที่ใช้ในงานวิจัยนี้ จะใช้เฉพาะในส่วนของ $W_{edge}E_{edge}$ เท่านั้น เนื่องจากลักษณะของภาพตัวอย่างที่นำมาทดลองไม่มีความซับซ้อนมากนัก ทำให้ผลรวมพลังงานอีก 2 เทอมที่เหลือมีค่าน้อยมาก จึงไม่ส่งผลกระทบต่อพลังงานรวมของระบบ ซึ่งก็เพียงพอแล้วสำหรับการหาพลังงานในส่วนนี้ การหาขอบภาพ (Edge Detection) เป็นการหาเส้นรอบวัตถุที่อยู่ในภาพ เมื่อทราบเส้นรอบวัตถุ จะสามารถคำนวณหาพื้นที่ (ขนาด) หรือรูป จำชนิดของวัตถุนั้นได้ อย่างไรก็ตามการหาขอบภาพที่ถูกต้องสมบูรณ์ไม่ใช่เป็นเรื่องที่ง่าย โดยเฉพาะอย่างยิ่งการหาขอบของภาพที่มีคุณภาพต่ำ มีความแตกต่างระหว่างพื้นหน้ากับพื้นหลังน้อย หรือมีความสว่างไม่สม่ำเสมอทั่วภาพ ขอบภาพเกิดจากความแตกต่างของความเข้มแสงจากจุดหนึ่งไปยังอีกจุดหนึ่ง หากความแตกต่างมีค่ามาก ขอบภาพก็จะเห็นได้ชัด ถ้าความแตกต่างมีค่าน้อย ขอบภาพก็จะไม่ชัดเจน โดยอัลกอริทึมที่ใช้ในวิทยานิพนธ์คือโซเบล (Sobel Edge Detection) (Emanuele Trucco, Alessandro Verri, 1998) จะเป็นการหาขอบภาพโดยใช้เทมเพลตขนาด 3x3 สองเทมเพลตครอปกับค่าความเข้มแสงจริงของภาพ โดยเทมเพลตแรก X_{diff} จะใช้หาค่าความแตกต่างในแนวนอน และ Y_{diff} จะใช้หาค่าความแตกต่างในแนวตั้ง

-1	0	+1	+1	+2	+1
-2	0	+2	0	0	0
-1	0	+1	-1	-2	-1
Gx			Gy		

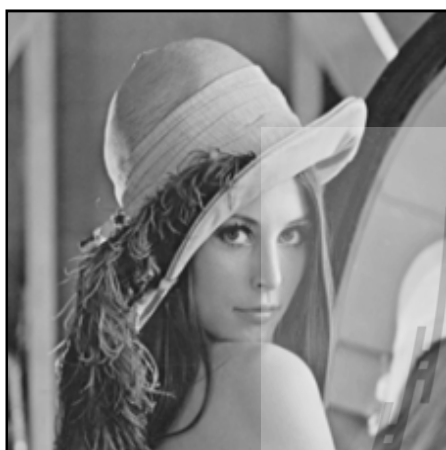
รูปที่ 2.4 เทมเพลตใช้ในการหาค่าความแตกต่างความเข้มแสงในภาพ

อธิบายขั้นตอนการทำได้ดังนี้

ภาพ	$ X_{diff} + Y_{diff} $	เทรตไฮลด์ด้วยค่า 12
0 0 0 0 0 0 2 0 3 3		
0 0 0 1 0 0 0 2 4 2	4 6 4 10 14 12 14 4	0 0 0 0 1 1 1 0
0 0 2 0 2 4 3 3 2 3	6 8 10 20 16 12 6 0	0 0 0 1 1 1 0 0
0 0 1 3 3 4 3 3 3 3	4 10 14 10 2 4 2 4	0 0 1 0 0 0 0 0
0 1 0 4 3 3 2 4 3 2	2 12 12 2 2 4 6 8	0 1 1 0 0 0 0 0
0 0 1 2 3 3 4 4 4 3		

รูปที่ 2.5 ขั้นตอนการหาขอบภาพ

เริ่มจากค่าความเข้มแสงแต่ละพิกัดบนภาพ (Pixel) ดังนั้นภาพหนึ่งภาพจะได้ค่าความเข้มแสงเท่ากับขนาดของภาพ จากนั้นนำเทมเพลตมาครอบแต่ละพิกัดที่สนใจไว้ แล้วทำการคูณสะสมทั้งแปดจุดรอบพิกัดที่สนใจ แล้วหาค่าเฉลี่ยจะได้ค่าความแตกต่างในแนวนอน และแนวตั้งออกมา แล้วนำผลที่ได้มาบวกกันจะได้ขอบของภาพ ส่วนขั้นสุดท้ายคือการปรับภาพให้เป็นเกรย์สเกล ผลลัพธ์ดังรูปที่ 2.7 (Chaganti, Venkata Ravikiran, 2005) จากภาพต้นแบบรูปที่ 2.6

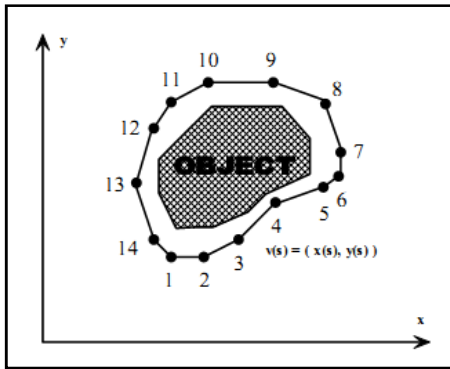


รูปที่ 2.6 ภาพต้นแบบ

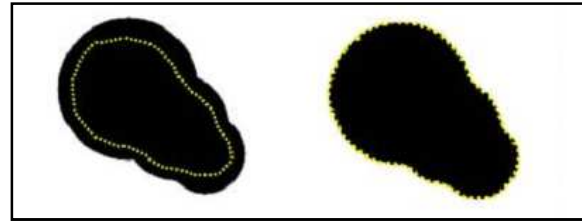


รูปที่ 2.7 หาขอบภาพด้วยโซเบล

นอกจากนี้ยังมีอีกหลายวิธีในการหาขอบเขตของภาพโดยการอ้างวิธีของ Snake Active Contour Model มาใช้ในการหาขอบของวัตถุที่สนใจ พื้นฐานแล้วจะเป็นการปรับค่าของพลังงานของภาพให้ลดลงมากที่สุด (Sandra V. B. Jardim, Mário A. T. Figueiredo) ซึ่งก็มีวิธีเฉพาะในการหาขอบภาพแต่ละภาพ โดยมีความยืดหยุ่นของส่วนโค้งปรับได้โดย (α, β) สามารถปรับเปลี่ยนที่ได้ตลอดเวลาตามแต่ละลักษณะจำเพาะของภาพเพื่อให้ได้ขอบหรือวัตถุที่สนใจใกล้เคียงที่สุด ทั้งหมดขึ้นอยู่กับเซตของจุดที่เชื่อมต่อกันโดยผู้ใช้เป็นผู้กำหนดในขั้นตอนแรก แสดงไว้ในรูปที่ 2.8ก คอนทัวร์จะถูกนิยามไว้โดยเซตของจุดจำนวนมากซึ่งเรียงกันเป็นลำดับรอบวัตถุที่สนใจ การที่จะทำให้เซตของจุดเหล่านี้ยึดติดกับขอบของภาพนั้น จะเป็นการทำงานร่วมกันระหว่างผู้ใช้และระบบของตัวโปรแกรม (Semi-Automatic) โดยผู้ใช้ต้องกำหนดค่าเริ่มต้นให้กับคอนทัวร์ก่อนซึ่งก็คือการวางจุดเหล่านี้นรอบขอบของวัตถุที่เราสนใจ อาจมีการกำหนดไว้ชัด ๆ กับขอบของรูป แสดงดังรูปที่ 2.8ข หลังจากนั้นจะเป็นหน้าที่ของโปรแกรมที่จะคำนวณเพื่อหาขอบรูปทรงของวัตถุที่สนใจออกมา



รูปที่ 2.8ก กำหนดจุดบนคอนทัวร์



รูปที่ 2.8ข การเคลื่อนที่เข้าหาขอบ

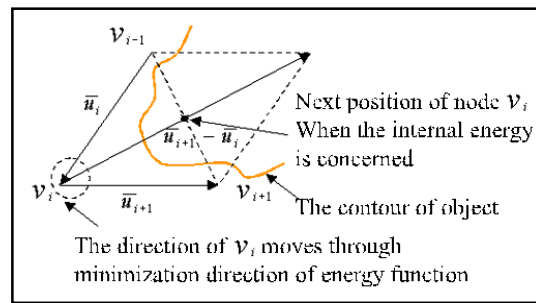
นอกจากนี้ยังมีการนำเสนอฟังก์ชันของพลังงานที่พัฒนามาเพื่อประยุกต์ใช้กับภาพ หรือวัตถุที่มีการเคลื่อนไหว (Tracking) โดยใช้ Greedy algorithm ในการพัฒนา (Jeongju Choi, Jong Shik Kim, 2002) ซึ่งเกิดจากการนำเอาฟังก์ชันของพลังงานเดิมมาประยุกต์ได้แก่สมการที่ 2-1, 2-2 และ 2-3 โดยปรับได้ดังนี้

$$E_{cont} = \alpha |u_i^t - u_i^{t-1}| \quad (2-5)$$

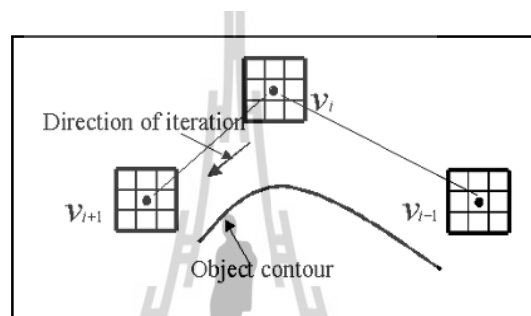
$$E_{cur} = \beta |E_{curi}^t - E_{curi}^{t-1}| \quad (2-6)$$

$$E_{add} = \gamma |v_i - G_j| \quad (2-7)$$

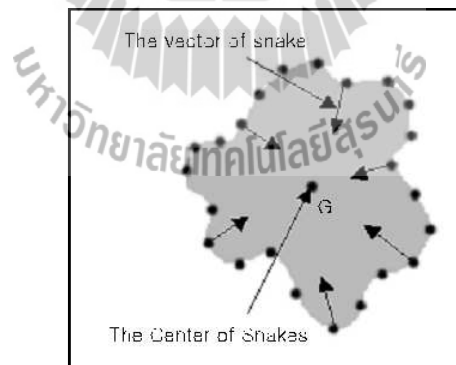
โดยมีการปรับในส่วนแรกคือมีการปรับ E_{int} ใหม่ออกเป็น 3 ส่วน นิยามได้รูปที่ 2.9 และ 2.10 โดย E_{cont} จะเป็นในส่วนของเวกเตอร์ u ซึ่งเป็นเวกเตอร์ที่เกิดจากจุดแต่ละจุดบนคอนทัวร์ที่อยู่กันเป็นลำดับ ส่วน E_{cur} จำทำหน้าที่ควบคุมเกี่ยวกับส่วนโค้งเว้าของวัตถุ และส่วนที่เพิ่มเข้ามาคือ E_{add} เกิดจากการเอาจุดแต่ละจุดบนคอนทัวร์มาลบจุดศูนย์กลางของคอนทัวร์ในแต่ละรอบตามรูปที่ 2.11



รูปที่ 2.9 ตำแหน่งและทิศทางของจุดบนคอนทัวร์



รูปที่ 2.10 ทิศทางของจุดในแต่ละรอบ

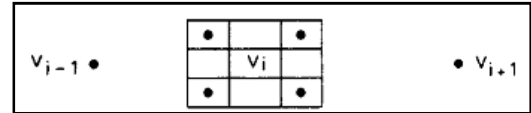
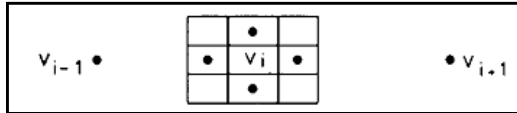


รูปที่ 2.11 จุดศูนย์กลางแต่ละรอบ

ดังนั้นจากสมการที่ปรับใหม่ทั้งหมดเขียนได้ดังนี้

$$E_{int} = \alpha |u_i^t - u_i^{t-1}| + \beta |E_{curi}^t - E_{curi}^{t-1}| + \gamma |v_i - G_j| \quad (2-8)$$

จากสมการด้านบนได้มีการพัฒนาการหาพลังงานของวัตถุว่า ตำแหน่งในรอบถัดไปที่จะทำการเลื่อนเข้าหาวัตถุจะเป็นไปในทิศทางใด เพื่อความรวดเร็วในการค้นหาตำแหน่งใหม่บนคอนทัวร์ (K.M.Lam and H Yan, 1993) จึงสร้างรูปแบบใหม่ตามสมการด้านบนโดยดูจากการเปลี่ยนไปในแต่ละรอบตามเวกเตอร์ u_i และจุดบนคอนทัวร์ v_i นั้นเอง โดยแยกเป็น 2 รูปแบบดังนี้



รูปที่ 2.12 รูปแบบที่ 1 ของการค้นหา

รูปที่ 2.13 รูปแบบที่ 2 ของการค้นหา

และยังมีการสร้างการจำลองโค้ดดังนี้

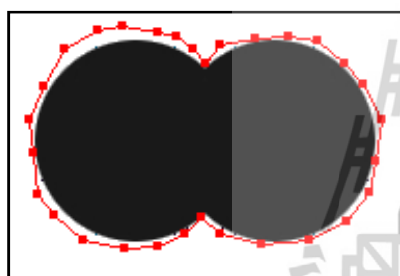
```

jmin = j
Move point vi to location jmin
if jmin not current location
    ptsmoved += 1
    pattern = -pattern
/*process to dermine where to allow corners in the next iteration*/
for i = 0 to n - 1
    if Ecurvaturei < threshold1 then
        ci = |ui / |ui| - ui+1 / |ui+1| |2
for i = 0 to n - 1
    if (ci > ci-1 and ci > ci+1 and ci < threshold2 and mag(vi) >
        threshold3) then
        β = 0
Until ptsmoved < threshold3

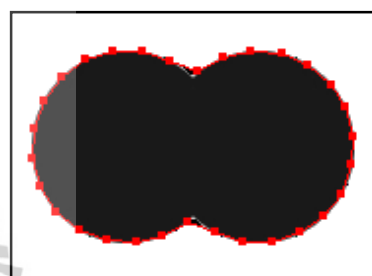
```

พบว่าวิธีดังกล่าวเหมาะสมสำหรับวัตถุที่มีการเคลื่อนที่ที่มีประสิทธิภาพต่ำ ถึงแม้จะมีข้อผิดพลาดมากในคาบแรกก็ตาม และถูกนำเสนอสำหรับการสร้างภาพสามมิติที่มีการเคลื่อนไหว

สรุปคือในส่วนของการแยกองค์ประกอบของภาพนั้นจะเริ่มจากการที่ใช้กำหนดจุดเริ่มต้นรอบวัตถุที่สนใจ หลังจากนั้น โปรแกรมจะทำการแยกองค์ประกอบของวัตถุที่ต้องการออกจากองค์ประกอบรวมของภาพ ด้วยสมการการลดลงของพลังงานในรูปภาพ หรือก็คือ จุดที่ทำการในกำหนดตอนแรกนั้นจะทำการเปลี่ยนพิกัดใหม่ไปเรื่อย ๆ โดยการวิ่งเข้าหาจุดที่มีพลังงานน้อยในแต่ละรอบจนถึงจำนวนรอบที่เรากำหนด การแยกองค์ประกอบของภาพจึงช่วยเพิ่มคุณภาพของภาพให้ดีขึ้นได้โดยตัดองค์ประกอบที่ไม่จำเป็นออกไป ซึ่งก็สามารถทำได้หลายวิธี แต่ยิ่งเพิ่มประสิทธิภาพมากขึ้นเท่าไรจำนวนรอบที่ต้องใช้ในการคำนวณก็ต้องเพิ่มไปด้วย ทำให้ใช้เวลาเพิ่มมากขึ้น แต่ถ้าเทียบกับการคำนวณที่ยุ่งยากก็ไม่ได้เสียประโยชน์อะไรมาก แต่ก็มีข้อเสียอยู่บ้างคือจำเป็นที่ผู้ใช้งานกำหนดจุดเริ่มต้นในตอนแรก จากนั้นจึงจะเป็นหน้าที่ของระบบโดยได้มีการนำมาใช้อย่างแพร่หลายซึ่งมีผลดังรูป



รูปที่ 2.14 กำหนดจุดบนคอนทัวร์โดยผู้ใช้งาน



รูปที่ 2.15 ระบบทำการคำนวณหาคอนทัวร์ของวัตถุ

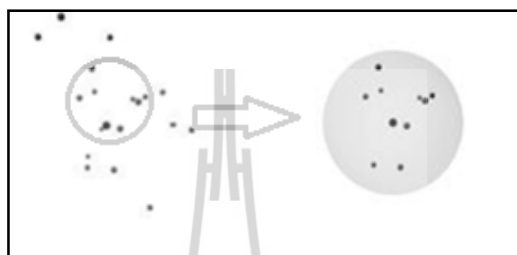
2.4 การสร้างพื้นผิวจากจุด (Surface Reconstruction from Point Clouds)

จากหัวข้อที่ 2.3 เมื่อนำภาพ Ultrasound อยู่ในรูปของภาพสองมิติไปผ่านกระบวนการแยกองค์ประกอบของภาพ ข้อมูลที่ได้มาคือจุดจำนวนมากที่กระจายตัวกันอยู่รอบ ๆ ขอบของวัตถุที่เราสนใจในแต่ละภาพ ซึ่งก็คือคอนทัวร์ของภาพนั้น ๆ ขั้นตอนต่อมาคือการนำจุดที่ได้เหล่านี้มาทำการสร้างพื้นผิว (Bill Toll, Fuhua Cheng, 1999) ให้ได้เป็นภาพจำลองแบบสามมิติด้วยวิธีการต่าง ๆ ด้วยวิธีการดังต่อไปนี้ (Yue Li, Matthew Hielsberg, 2012)

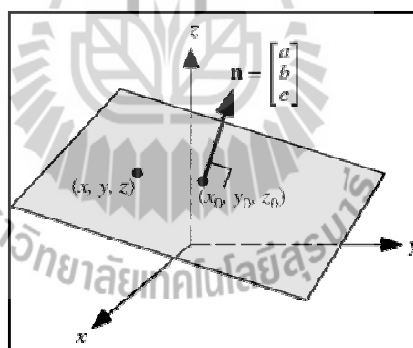
2.4.1 Neighborhoods

สำหรับขั้นตอนนี้เมื่อเรามีข้อมูลเป็นจุดจำนวนมากซึ่งกระจายตัวกันอยู่อย่างอิสระ ไม่มีแบบแผน ดังนั้นการที่จะสร้างพื้นผิวรอบวัตถุที่เราสนใจนั้น ก็คือการแบ่งจุดเหล่านี้ออกเป็นบริเวณย่อย ๆ เหมือนการนำพื้นผิวแผ่นเล็ก ๆ จำนวนมากมาต่อกันสร้างเป็นพื้นผิวของรูปใหญ่ ซึ่งก็

คือรูปในแบบสามมิตินั้นเอง เพราะฉะนั้นสิ่งแรกที่ต้องทำก็คือ การแบ่งจุดที่ได้มาเหล่านี้ออกเป็น บริเวณเล็ก ๆ โดยจำนวนสมาชิกในแต่ละบริเวณนั้น จะถูกกำหนดตามความเหมาะสมสำหรับข้อมูล นั้น ๆ เช่น บริเวณที่มีการกระจายตัวของจุดอยู่อย่างหนาแน่นเราควรเลือกระยะห่างระหว่างสมาชิก เป็นวงแคบ ๆ หรือบางบริเวณที่มีการกระจายตัวไม่ค่อยแน่นหนา ก็อาจมีระยะห่างระหว่างสมาชิก กว้างขึ้นไป จำนวนสมาชิกอาจน้อยหรือมากต่างกันไป เพื่อลดโอกาสที่จะมีข้อผิดพลาดในการสร้าง ระนาบโดยให้ครอบคลุมข้อมูลมากที่สุดดังรูปที่ 2.16



รูปที่ 2.16 Neighborhoods



รูปที่ 2.17 Normal vector

ที่มา : <http://goanna.cs.rmit.edu.au/~gl/teaching/web3d/lectures/web3d/> [20 March 2013]

2.4.2 เวกเตอร์ตั้งฉาก (Normal Vectors)

เมื่อแบ่งสมาชิกออกเป็นกลุ่มข้อมูลย่อย ๆ แล้ว จะสมมติว่าข้อมูลเหล่านี้เสมือนกับ เป็นระนาบเล็ก ๆ จำนวนมากรวมกันอยู่ ดังนั้นภายในระนาบหนึ่งระนาบก็จะมีเวกเตอร์ที่ตั้งฉากอยู่ หนึ่งอันเช่นกัน ดังรูปที่ 2.17 และ 2.18 สำหรับวิธีที่ใช้ในการหาเวกเตอร์ตั้งฉากนั้นจะใช้วิธีการดังนี้

2.4.2.1 Principal Component Analysis

ใช้ในการทำให้เซตข้อมูลง่ายลง ลดปริมาณข้อมูลอย่างฉลาด ทำได้ด้วยการลดจำนวนมิติของข้อมูล (reduce dimensionality) (Herve Abdi, Lynne J. Williams, 2010) และทำให้เสียข้อมูลสำคัญไปเพียงเล็กน้อยด้วยการกำจัดข้อมูลบางส่วนที่ไม่สำคัญลำดับท้าย ๆ ซึ่งก็คือการลดการเก็บข้อมูลที่ซ้ำซ้อน หรือมีความแตกต่างกันน้อย จะเก็บเฉพาะค่าเฉลี่ยของข้อมูลชุดนั้นไว้ค่าเดียวเท่านั้น คือหาค่าเฉลี่ยผ่านข้อมูลทุกจุด หรือให้มากที่สุด เพื่อให้เห็นพิจารณาตัวอย่างง่าย ๆ ว่าเราสามารถเก็บเวกเตอร์ภาพฉายและค่าภาพฉายบนเวกเตอร์นั้น ทำให้เราได้ระบบพิกัดจุดที่ง่ายกว่าเดิม โดยอาศัยวิธีการทางสถิติคือ ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation, SD) เป็นการวัดการกระจายตัวของข้อมูล ยิ่งชุดข้อมูลกระจายห่างออกจากค่าเฉลี่ยมากเท่าใด ส่วนเบี่ยงเบนมาตรฐานก็ยิ่งมีค่ามากขึ้นเท่านั้น โดยมีสูตรคือ (Walpole, R.E., Myers, R.H., 1990)

$$s = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{(n-1)}} \quad (2-9)$$

ค่าความแปรปรวน (Variance) ที่จริงแล้วค่าความแปรปรวนเป็นค่าดั้งเดิมในทางสถิติ ซึ่งมีค่าเป็นกำลังสองของส่วนเบี่ยงเบนมาตรฐาน สูตรคือ (Walpole, R.E., Myers, R.H., 1990)

$$s^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{(n-1)} \quad (2-10)$$

ค่าความแปรปรวนร่วม (Covariance)

$$cov(X, Y) = \sum_{i=1}^n \frac{(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)} \quad (2-11)$$

เนื่องจากโดยปกติเราต้องการวัดความสัมพันธ์ของทุกคู่ตัวแปร (ข้อมูลในทุกมิติ) เราจึงทำการจับคู่ตัวแปรทุกคู่ที่เป็นไปได้ รวมทั้งจับคู่ตัวแปรเดียวกันเพื่อหาความแปรปรวนธรรมดาด้วย ผลของการจับคู่ทุกรูปแบบนำไปสู่การคำนวณที่เป็นระบบในรูปของเมตริกซ์ความแปรปรวนร่วม ดังแสดงให้เห็นจากการจับคู่ของข้อมูลสามมิติด้านล่าง (Rindsay I Smith, 2002)

$$M_{x,y,z} = \begin{bmatrix} C(X, X) & C(X, Y) & C(X, Z) \\ C(Y, X) & C(Y, Y) & C(Y, Z) \\ C(Z, X) & C(Z, Y) & C(Z, Z) \end{bmatrix} \quad (2-12)$$

2.4.2.2 วิธีกำลังสองน้อยที่สุดของระนาบสามมิติ

(Plana Fitting 3D: Least square plan)

การฟิตข้อมูล (Data fitting) คือ การสร้างฟังก์ชันที่เชื่อมโยงค่าของตัวแปรจากข้อมูล “ที่ดีที่สุด” หรือ “เหมาะสมที่สุด” บนหลักเกณฑ์และเหตุผล ซึ่งวิเคราะห์ได้ในทางคณิตศาสตร์

สำหรับเซตของจุดข้อมูลในสามมิติ

$$\{(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n)\} \quad (2-13)$$

ระนาบในสามมิติถูกกำหนดโดย (David Eberly, 1998-2008)

$$z = Ax + By + C \quad (2-14)$$

ซึ่ง $N = \langle A, B, -1 \rangle$ เป็นเวกเตอร์ที่ตั้งฉากกับระนาบ และมีขนาดเท่ากับ 1 นั่นคือ $|N| = \sqrt{A^2 + B^2 + 1} = 1$

และให้ d_i คือระยะทางจากจุด $P_i(x_i, y_i, z_i)$ ถึงระนาบ $z = Ax + By + C$ ถูกกำหนดโดย

$$d_i = \frac{|Ax_i + By_i + C - z_i|}{\sqrt{A^2 + B^2 + 1}} = |Ax_i + By_i + C - z_i| \quad (2-15)$$

เราต้องคำนวณค่าของ $d_1, d_2, \dots, d_n$ และต้องการให้เกิดค่าคลาดเคลื่อนให้น้อยที่สุด โดยเรามีแนวทางในการวัดค่าคลาดเคลื่อนเหล่านี้ในรูปผลบวกได้ และเราเลือกใช้ $\sum_{i=1}^n d_i^2$ ซึ่งมีความเหมาะสมที่น่าจะนำมาเป็นเงื่อนไขในการสร้างค่า $z(x, y, 1)$ (ในที่นี้ $z(x, y, 1)$ คือ ฟังก์ชัน) ซึ่งเป็นฟังก์ชันที่สร้างขึ้นเพื่อเชื่อมโยงค่า x_i, y_i, z_i โดยให้ $\sum_{i=1}^n d_i^2$ มีค่าน้อยที่สุด ทำให้การฟิตข้อมูลด้วยเงื่อนไขนี้เรียกว่า วิธีกำลังสองน้อยที่สุด (Least square)

สำหรับการหาเงื่อนไขที่ทำให้ $\sum_{i=1}^n d_i^2$ นี้มีค่าต่ำที่สุดคือ การหาอนุพันธ์และหาค่าวิกฤต

$$F(A, B, C) = \sum_{i=1}^n d_i^2 = \sum_{i=1}^n (Ax_i + By_i + C - z_i)^2 \quad (2-16)$$

และ $F(A, B, C)$ นี้เป็นฟังก์ชันบวก และเป็นสมการไฮเปอร์โบลอยด์ซึ่งมีจุดต่ำสุดเมื่อ $\nabla F(A, B, C) = (0, 0, 0)$ และสร้างระบบสมการสามตัวแปร A, B, C นั่นคือ

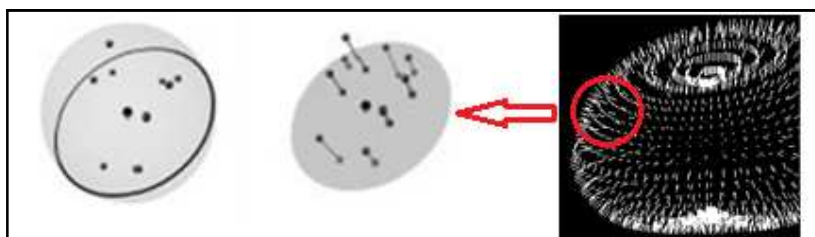
$$(0, 0, 0) = \nabla F = 2 \sum_{i=1}^n [Ax_i + By_i + C - z_i] (x_i, y_i, 1) \quad (2-17)$$

และสามารถเขียนในรูปของเมตริกซ์ได้ว่า

$$\begin{bmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i y_i & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i y_i & \sum_{i=1}^n y_i^2 & \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n y_i & \sum_{i=1}^n 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n x_i z_i \\ \sum_{i=1}^n y_i z_i \\ \sum_{i=1}^n z_i \end{bmatrix} \quad (2-18)$$

คำตอบของระบบสมการนี้จะทำให้เราสามารถคำนวณ A, B, C สำหรับระนาบ $z = Ax + By + C$

ทั้งสองวิธีข้างต้นทำให้ได้ผลลัพธ์ดังรูป



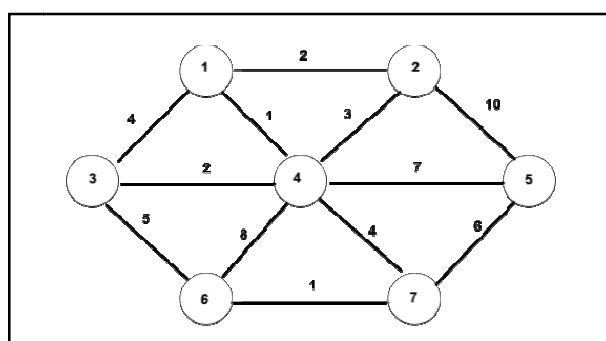
รูปที่ 2.18 เวกเตอร์ตั้งฉากของแต่ละ Neighborhood

2.4.3 Minimum spanning tree

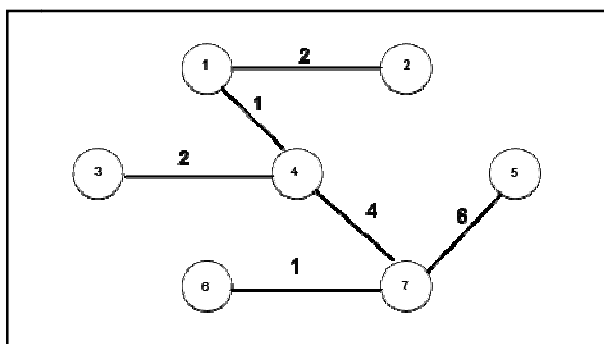
สำหรับขั้นตอนการหา Minimum spanning tree นี้ มีความสำคัญในการสร้างพื้นผิวของแบบจำลองสามมิติคือ เมื่อได้เวกเตอร์ตั้งฉากจำนวนมากจากกระบวนการก่อนหน้านี้ไม่ว่าจะเป็นการทำ PCA หรือ Least square เมื่อได้เวกเตอร์ตั้งฉากมาจะทำให้ทราบว่าพื้นผิวที่จะถูกสร้างขึ้นนั้นจะวางตัวในทิศทางใด เพื่อความเป็นระเบียบ และความสวยงามของพื้นผิวที่จะถูกสร้างขึ้น จึงจำเป็นต้องทำการกลับทิศทางของเวกเตอร์เหล่านี้ให้เป็นไปในทิศทางเดียวกันก่อนคือ พุ่งเข้าหาวัตถุทั้งหมด หรือพุ่งออกจากวัตถุทั้งหมดนั่นเอง ดังนั้นการที่จะค้นหาเส้นทางไปยังทุก ๆ ระบายจึงจำเป็นต้องใช้ Minimum spanning tree เข้ามาช่วยเพราะ เป็นการค้นหาเส้นทางที่สั้นที่สุดไปยังทุก ๆ โหนด โดยกำหนดเรียกใช้โหนดทุกโหนดและทุกเส้นการเชื่อมต่อ มาลำดับความสำคัญของน้ำหนัก โดยเริ่มจากค่าน้อยที่สุด ทำการเชื่อมต่อคู่โหนดนั้น และดำเนินการต่อไปในค่าน้ำหนักที่ต่อกัน แต่ถ้าโหนดใดมีการเชื่อมต่อคู่โหนดแล้วจะไม่เชื่อมต่ออีก ส่วนค่าน้ำหนักที่ใช้ในการค้นหาเส้นทางนั้นจะเปลี่ยนจากระยะทางมาเป็นพลังงานที่ใช้ในการกลับทิศทางของเวกเตอร์ โดยมีอัลกอริทึมที่ใช้ดังต่อไปนี้

2.4.3.1 Prim's Algorithm

แนวคิดของ Prim's algorithm คือเริ่มจากโหนดเริ่มต้นจากนั้นให้ดูโหนดใกล้เคียงที่เชื่อมกันอยู่ว่ามีเส้นทางไหนที่มีน้ำหนักน้อยสุด (ในส่วนนี้ที่ใช้คือพลังงานในการกลับเวกเตอร์) ก็ให้ไปที่โหนดนั้น ๆ ก่อน โดยที่เส้นทางนั้น ๆ จะต้องไม่ไปยังโหนดที่มันเลือกไปแล้ว และจะต้องไปให้ครบทุกโหนดดังรูป (Susanna S. Epp, 1995)



รูปที่ 2.19 กราฟตั้งต้น



รูปที่ 2.20 รูปที่คำนวณได้จาก Prim

2.4.3.2 Dijkstra algorithm

เพื่อแก้ไขปัญหาเส้นทางที่สั้นที่สุดไปยังโหนดใด ๆ สำหรับขั้นตอนวิธีนี้จะหาระยะทางสั้นที่สุดจากจุดหนึ่งไปยังจุดใด ๆ ในกราฟโดยจะหาเส้นทางที่สั้นที่สุดไปที่ละจุดยอดเรื่อย ๆ จนครบตามที่ต้องการ กำหนดให้โหนดหนึ่งเป็นโหนดเริ่มต้น (Initial node) และกำหนดให้ "ระยะทางของโหนด Y" (distance of node Y) หมายถึง ระยะทางจากโหนดเริ่มต้นไปยังโหนด Y ขั้นตอนวิธีของไดคัสตราจะกำหนดค่าระยะทางเริ่มต้นไว้บางโหนดและจะเพิ่มค่าไปที่ละขั้นตอน (Cormen, Thomas H, Leiserson, Charlie E., Rivest Ronald L., Stein Clifford, 2001)

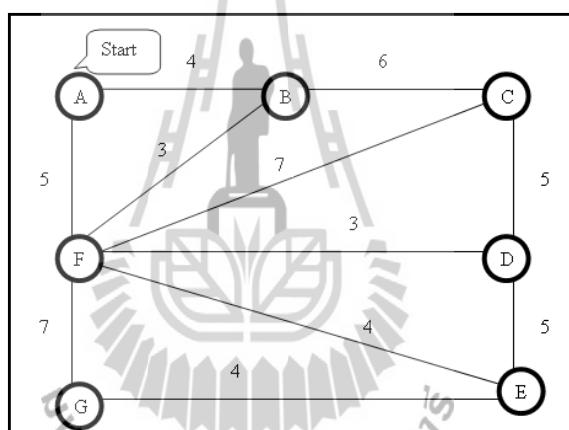
- 1) กำหนดให้ทุกโหนดมีค่าระยะทางตามเส้นเชื่อม โดยให้โหนดเริ่มต้นมีค่าเป็นศูนย์ และโหนดอื่นมีค่าเป็นอนันต์
- 2) ทำเครื่องหมายทุกโหนดยกเว้นโหนดเริ่มต้นว่ายังไม่ไปเยือน (unvisited) ตั้งให้โหนดเริ่มต้นเป็นโหนดปัจจุบัน สร้างเซตของโหนดที่ยังไม่ไปเยือนขึ้นมาเซตหนึ่งซึ่งประกอบด้วยทุกโหนดยกเว้นโหนดเริ่มต้น
- 3) จากโหนดปัจจุบัน พิจารณาโหนดข้างเคียงตามเส้นเชื่อมทุกโหนดที่ยังไม่ไปเยือน และคำนวณระยะทางต่อเนื่องของเส้นเชื่อม ตัวอย่างเช่น ถ้าโหนดปัจจุบันคือ A มีระยะทางของโหนดเป็น 6 และเส้นเชื่อมที่ต่อจาก A ไปยังโหนดข้างเคียง B มีระยะทางเป็น 2 ดังนั้นระยะทางของโหนด B (โดยผ่าน A) จึงเท่ากับ $6+2=8$ เป็นต้น ถ้าระยะทางที่คำนวณได้มีค่าน้อยกว่าค่าระยะทางที่บันทึกอยู่ของโหนดนั้น ให้เขียนทับค่าระยะทางของโหนดดังกล่าว แม้ว่าโหนดข้างเคียงได้ถูกพิจารณาแล้ว แต่ก็ยังไม่ทำเครื่องหมายว่าไปเยือนแล้ว (visited) ในขั้นตอนนี้ โหนดข้างเคียงจะยังคงอยู่ในเซตของโหนดที่ยังไม่ไปเยือนเช่นเดิม

4) เมื่อพิจารณาโหนดข้างเคียงจากโหนดปัจจุบันครบทุกโหนดแล้ว ทำเครื่องหมายโหนดปัจจุบันว่าไปเยือนแล้ว และนำออกจากเซตของโหนดที่ยังไม่ไปเยือน โหนดที่ไปเยือนแล้วนี้จะไม่ถูกนำมาตรวจสอบอีก ค่าระยะทางที่บันทึกอยู่จะสิ้นสุดและมีค่าน้อยสุด

5) โหนดปัจจุบันตัวถัดไปที่ถูกเลือกจะเป็นโหนดที่มีค่าระยะทางน้อยสุดในเซตของโหนดที่ยังไม่ไปเยือน

6) ถ้าเซตของโหนดที่ยังไม่ไปเยือนว่างแล้วให้หยุดการทำงาน ขั้นตอนวิธีเสร็จสิ้น หากไม่ใช่ให้เลือกโหนดที่ยังไม่ไปเยือนที่มีค่าระยะทางน้อยสุดเป็นโหนดปัจจุบัน แล้ววนกลับไปทำขั้นตอนที่ 3

ซึ่งขั้นตอนทั้งหมดถูกอธิบายด้วยรูปที่ 2.21 – 2.23 และตารางที่ 2.1 – 2.4

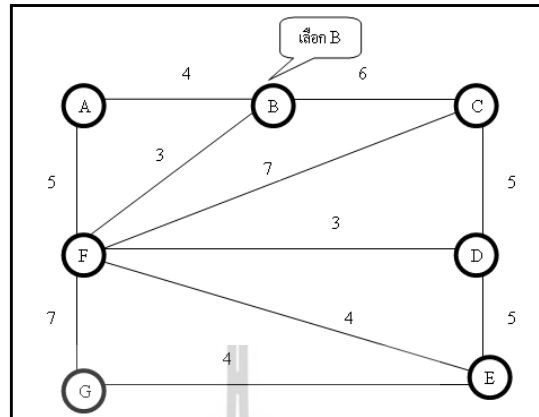


รูปที่ 2.21 แสดงตัวอย่าง Dijkstra's Algorithm

ตารางที่ 2.1 ตารางการคำนวณ Dijkstra's Algorithm เริ่มต้น

โหนด	$d(v)$	$pred(v)$
A	0	0
B	infinity	0
C	infinity	0
D	infinity	0
E	infinity	0
F	infinity	0
G	infinity	0

รอบที่ 1



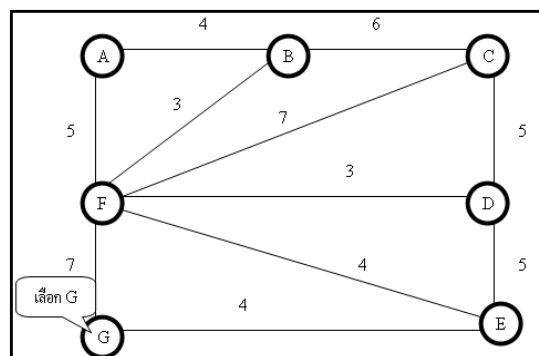
รูปที่ 2.22 แสดงการค้นหาจาก A ไป B

ตารางที่ 2.2 ตารางแสดงการคำนวณของน้ำหนักและ $d(v)$ รอบที่ 1

รอบที่	โหนดที่มีสีแสดล้อมรอบ	โหนดที่เชื่อมกัน	อาร์กที่พิจารณา	$weight(i,j)$	$weight(i,j) + d(i)$	$d(v)$
1	A	B	(A,B)	4	4	4
		F	(A,F)	5	5	5

ในทำนองเดียวโหนด B, C, D, E, F ทำเหมือนกับโหนด A และโหนดสุดท้ายคือ G มีขั้นตอนดังนี้

รอบที่ 6



รูปที่ 2.23 แสดงการค้นหาจาก F ไป G

ตารางที่ 2.3 ตารางแสดงการคำนวณของของน้ำหนักและ $d(v)$ รอบที่ 6

รอบที่	โหนดที่มีสีแดงล้อมรอบ	โหนดที่เชื่อมกัน	อาร์กที่พิจารณา	$weight(i,j)$	$weight(i,j) + d(i)$	$d(v)$
6	A,B,D,E,F,G	G	(E,G)	4	13	12
			(F,G)	7	12	12

ตารางที่ 2.4 ตารางแสดงการคำนวณของ Dijkstra's Algorithm เมื่อไปที่โหนด G

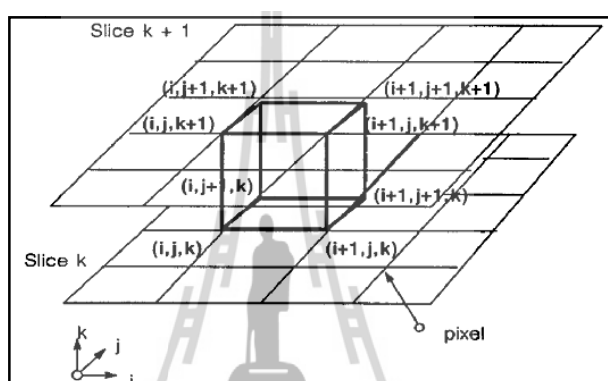
โหนด	$d(v)$	$pred(v)$
A	0	0
B	4	A
C	10	B
D	8	F
E	9	F
F	5	A
G	12	F

2.4.4 การสร้างรูปสามมิติ (Surface Reconstruction)

เนื่องจากข้อมูลที่ใช้ในงานวิจัยนี้เป็นภาพ 2 มิติที่ได้จากเครื่องถ่ายภาพเหนือเสียง และผ่านขั้นตอนของการแยกองค์ประกอบของภาพ จนถึงขั้นตอนการหา Minimum spanning tree ผลที่ได้ก็คือ จุดศูนย์กลางของระนาบเล็ก ๆ จำนวนมากและเวกเตอร์ตั้งฉากกับระนาบเหล่านี้เท่านั้น แต่ในงานวิจัยนี้ต้องการสร้างแบบจำลองสามมิติขึ้นมา เพื่อนำไปช่วยในการวินิจฉัยความผิดปกติของทารกในครรภ์ ดังนั้นจึงใช้วิธีการต่อกันของลูกบาศก์ (Marching Cubes Algorithm) (William E. Lorensen and Harvey E. Cline, 1987) เข้ามาช่วยในการสร้างแบบจำลอง ซึ่งภายในหนึ่งลูกบาศก์จะทำการคำนวณหาระยะที่สั้นที่สุดจากแต่ละมุมไปยังทุก ๆ จุดศูนย์กลางของระนาบทั้งหมด ซึ่งค่าที่ได้เป็นได้ทั้งลบและบวก หากพบว่ามุมที่อยู่ติดกันมุมหนึ่งมีค่าเป็นลบและอีกมุมหนึ่งมีค่าเป็นบวก แสดงว่าที่บริเวณนี้มีการตัดกันของพื้นผิวหรือก็คือมีพื้นผิวอยู่บริเวณลูกบาศก์นั่นเอง เรียกวิธีการนี้ว่า Signed distance function โดยจะทำการค้นหาทุก ๆ ลูกบาศก์สร้างแบบจำลองสามมิติ ซึ่งสามารถอธิบายได้ดังต่อไปนี้

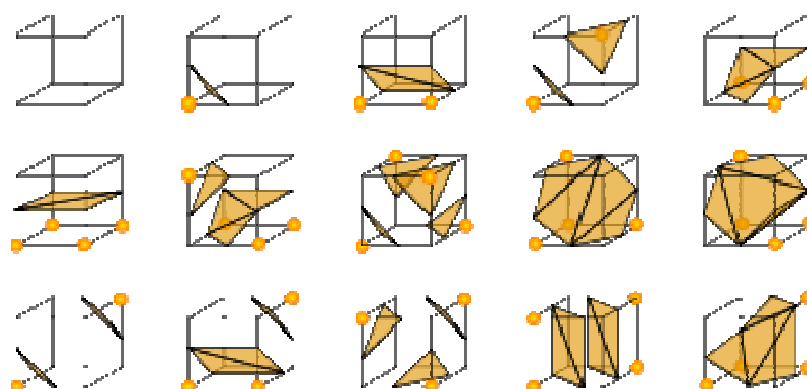
แนวความคิดสร้างรูปด้วยขั้นตอนวิธีการต่อกันของลูกบาศก์ (Sittichai Pomthong, Paramate Horkaew, Saowapak Sotthivirat, 2008) คือ การสร้างตาข่ายสามเหลี่ยม (Triangle Mesh) จากความเข้มของจุดภาพ (Pixel) ในแต่ละแผ่นภาพคั่นรูป แล้วนำจุดภาพไปสร้างตาข่ายสามเหลี่ยม โดยไปเปรียบเทียบรูปแบบในรูปถัดไปจนเกิดเป็นกล่องภาพ (Voxels) แล้วนำตาข่ายสามเหลี่ยมในแต่ละกล่องภาพมาต่อกัน จะได้ภาพสามมิติที่ต้องการ ซึ่งอธิบายด้วยขั้นตอนวิธีได้ดังนี้

- 1) เลือกแผ่นภาพ 2 แผ่นภาพที่อยู่ติดกัน เพื่อพิจารณาทีละ 4 จุดภาพของทั้ง 2 แผ่นภาพ



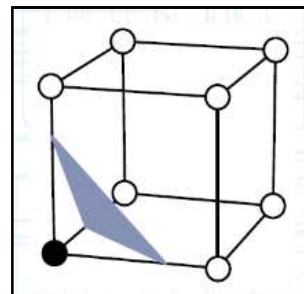
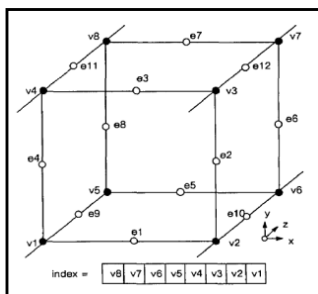
รูปที่ 2.24 จุดบนระนาบแต่ละระนาบ

- 2) พิจารณา 8 จุดภาพว่าตรงกับรูปแบบใดให้สร้างตาข่ายสามเหลี่ยมรูปแบบนั้น



รูปที่ 2.25 รูปแบบการสร้างภาพสามมิติ

3) สร้างตัวชี้ (Index) ในแต่ละกรณี



รูปที่ 2.26 การใส่ตัวชี้

รูปที่ 2.27 รูปการสร้างจุดบนพื้นผิว

4) สร้างจุดบนพื้นผิวจากการตัดกันของแต่ละขอบ (edge)

$$v_i = v_1(1 - u) + v_2 \cdot u \quad (2-19)$$

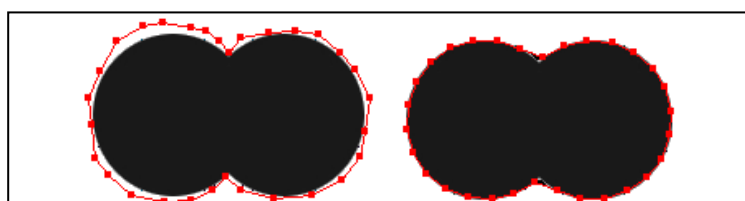
โดยที่

$$u = \frac{v_1 - v_i}{v_1 - v_2} \quad (2-20)$$

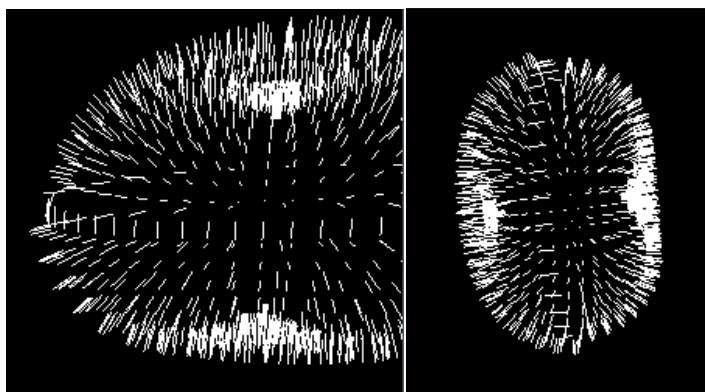
5) คำนวณหาเวกเตอร์ตั้งฉาก (normal vector) ของแต่ละจุดบนลูกบาศก์

6) คำนวณหาเวกเตอร์ตั้งฉาก (normal vector) ของแต่ละจุดบนดาข่ายสามเหลี่ยม

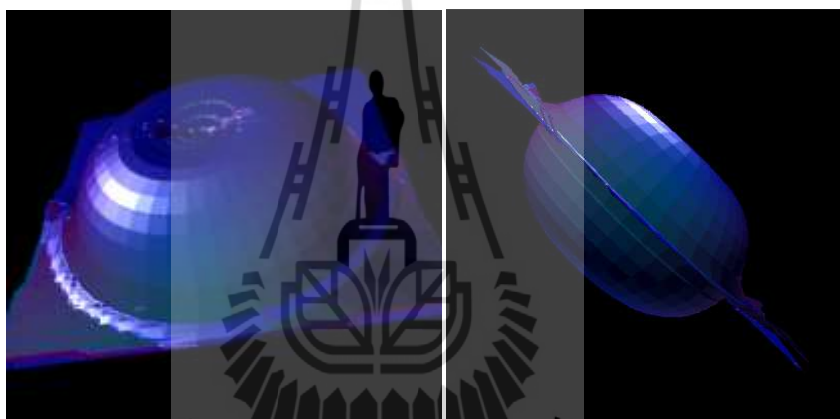
จากขั้นตอนการแยกองค์ประกอบของภาพจนถึงการสร้างพื้นผิวจากจุดทั้งหมดสามารถสรุปได้ดังรูปที่ 2.28-2.30



รูปที่ 2.28 การแยกองค์ประกอบของภาพในขั้นตอนการหาคอนทัวร์



รูปที่ 2.29 การแยกองค์ประกอบของภาพในขั้นตอนการหาเวกเตอร์ตั้งฉาก



รูปที่ 2.30 การสร้างแบบจำลองสามมิติจากจุดด้วยการสร้างแบบลูกบาศก์

2.5 การสร้างขยายจุดของพื้นผิวทรงกลม (Spherical Expansion)

ในกระบวนการสร้างพื้นผิวสามมิติด้วยวิธีการต่อกันของลูกบาศก์ (Marching cubes) เนื่องจากข้อมูลที่ใช้ในการทดสอบมีลักษณะเฉพาะตัวซึ่งสร้างจากคอนทัวร์เพียงระนาบเดียวแต่ทำให้เป็นปริมาตรโดยใช้สมการทางคณิตศาสตร์ช่วย ดังนั้นจึงอาจเกิดการผิดพลาดในบางส่วนทำให้เวกเตอร์ตั้งฉากที่ได้ออกมาไม่ตรงบางส่วน และเมื่อสร้างเป็นพื้นผิวพบว่าพื้นผิวที่ได้ยังคงมีปัญหาไม่เรียบเท่าที่ควร มีส่วนเกินจากเวกเตอร์ตั้งฉาก และจำนวนจุดที่มีมากเกินไป ซึ่งเมื่อผ่านกระบวนการต่าง ๆ แล้วทำให้สูญเสียเวลามากเกินไป วิธีหนึ่งที่จะช่วยให้ปัญหาดังกล่าวหมดไปคือการสร้างพื้นผิวจากการขยายจุดของพื้นผิวทรงกลม ซึ่งวิธีการนี้จะช่วยให้พื้นผิวเรียบขึ้นและลดจำนวนจุดลงโดยที่ยังคงรูปลักษณะเดิมของพื้นผิวไว้ ซึ่งมีวิธีดังนี้

2.5.1 การขยายจุดของพื้นผิวทรงกลม (Spherical Expansion)

การปรับพื้นผิวให้จุดของพื้นผิวน้อยลงและพื้นผิวราบเรียบขึ้นอีกทั้งยังคงรูปเดิมไว้สามารถทำได้โดยการขยายจุดบนทรงกลมที่ได้จากการสร้างตัวแปรของพื้นผิวมากระจายจุดออกด้วยความถี่สม่ำเสมอด้วยฟังก์ชันทรงกลมแบบฮาร์โมนิก (Spherical harmonic Function) ซึ่งขั้นตอนทั้งหมดอธิบายได้ดังนี้ การขยายของพื้นผิวทรงกลมด้วยฟังก์ชันฮาร์โมนิก (W.H. Press, B.P. Flannery, S. A. Teukolsky, and W. Vetterling, 1988) คือการแปลงจุดบนทรงกลมรัศมีหนึ่งหน่วยซึ่งเป็นจุดในพิกัดเชิงฉาก (Cartesian) ด้วยรูปแบบของฟูเรียร์ (Fourier Transform) และจะกระจายจุดต่าง ๆ ในความถี่คงตัวสม่ำเสมอ ด้วยฟังก์ชันทรงกลมฮาร์โมนิก (Spherical Harmonic) โดยความถี่การกระจายจุดจะขึ้นอยู่กับมุม θ และ ϕ บนทรงกลมรัศมีหนึ่งหน่วย และการกลับคืนสู่รูปทรงเดิมจะขึ้นกับลำดับตามอนุกรมฟูเรียร์ ขั้นตอนอธิบายได้ดังนี้

$$x(\theta, \phi) = \begin{pmatrix} x(\theta, \phi) \\ y(\theta, \phi) \\ z(\theta, \phi) \end{pmatrix} \quad (2-21)$$

โดยที่ x, y , และ z เป็นพิกัดเชิงฉาก (Cartesian)

$$\theta \in [0, 2\pi] \text{ และ } \phi \in [0, \pi] \quad (2-22)$$

ฟังก์ชันแปลงพื้นผิวในรูปแบบฟูเรียร์ (Fourier series)

$$x(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l c_l^m Y_l^m(\theta, \phi) \quad (2-23)$$

ฟังก์ชันทรงกลมแบบฮาร์โมนิก (Spherical harmonic Function) ซึ่ง

$$Y_l^m(\theta, \phi) = \sqrt{\frac{2l+1}{4\pi} \frac{(l-|m|)!}{(l+|m|)!}} P_l^{|m|}(\cos\theta) e^{i|m|\phi} \quad (2-24)$$

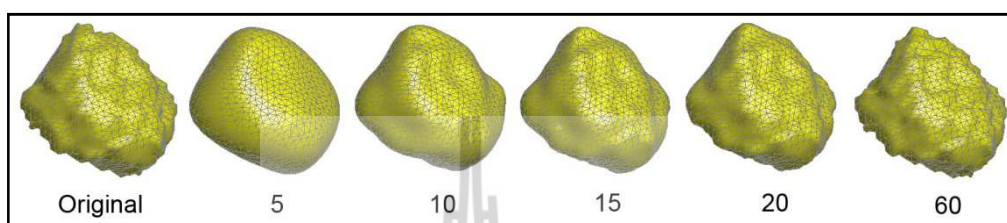
โดยที่ฟังก์ชันโพลิโนเมียล P

$$P_l^m(x) = \frac{(-1)^m}{2^l l!} (1-x^2)^{\frac{m}{2}} \frac{d^{m+1}}{dx^{m+1}} (x^2-1)^l \quad (2-25)$$

สัมประสิทธิ์ c หาได้โดย

$$c_l^m = (Z^T Z)(Z^T V) \quad (2-26)$$

ซึ่ง $Z_{l,m} = y_l^m(\theta_i, \phi_i)$ และ V คือจุดบนพื้นผิวเดิม



รูปที่ 2.31 จำนวนรอบของแต่ละ order

ที่มา : <http://www.stat.wisc.edu/~mchung/research/amygdala/> [20 March 2013]

จากรูปคือภาพการขยายจุดของพื้นผิวทรงกลมในแต่ละระดับ (order) ของอนุกรมฟูเรียร์ (Fourier series) จะเห็นว่าเมื่อลำดับยิ่งสูงพื้นผิวที่ได้จะใกล้เคียงกับพื้นผิวเดิม

2.6 สรุปงานวิจัยที่เกี่ยวข้อง

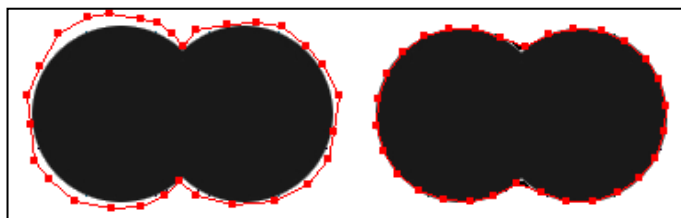
จากหัวข้อที่ 2.2 ถึงหัวข้อที่ 2.5 งานวิจัยที่กล่าวมาข้างต้นทั้งหมดนั้นเป็นงานวิจัยซึ่งใช้กันอย่างแพร่หลายในการประมวลผลภาพ และมีประโยชน์ต่องานวิจัยการสร้างภาพใหม่จากภาพอัลตราซาวด์ โดยขั้นตอนวิธีทั้งหมดที่เกี่ยวข้องกับงานวิจัยได้แก่ การแยกองค์ประกอบของภาพ การสร้างรูปใหม่จากจุด การสร้างรูปสามมิติด้วยการต่อกันของลูกบาศก์ การขยายจุดของพื้นผิวทรงกลม การแยกองค์ประกอบภาพถูกนำมาใช้ในงานวิจัยเนื่องจากภาพถ่ายอัลตราซาวด์ที่ได้มาจากเครื่องถ่ายภาพเหนือเสียงนั้นนอกเหนือจากรูปของทารกในครรภ์แล้ว ยังมีน้ำคร่ำ ความถี่รบกวนอื่น พื้นหลังของภาพ จึงต้องทำการแยกองค์ประกอบของทารกเพื่อนำไปใช้สร้างรูปสามมิติต่อไป ซึ่งการสร้างภาพสามมิติวิธีที่ถูกนำมาใช้กันอย่างแพร่หลายก็คือการต่อกันของลูกบาศก์ (Marching cubes) เนื่องจากเกิดข้อผิดพลาดในขั้นตอนของการหาเวกเตอร์ตั้งฉาก และจำนวนของจุดมีมากทำให้ขึ้นรูปช้า วิธีแก้ไขปัญหาดังกล่าวคือการขยายจุดของพื้นผิวทรงกลม ผลลัพธ์ที่ได้จะเป็นรูปทรง 3 มิติที่ถูกขยายด้วยลำดับอนุกรมฟูเรียร์ ซึ่งจะทำให้ได้รูปทารกที่ใกล้เคียงกับรูปร่างเดิม แต่จะมีพื้นผิวราบเรียบขึ้น ซึ่งขั้นตอนการทดลองทั้งหมดได้อธิบายไว้แล้วในบทถัดไป

บทที่ 3

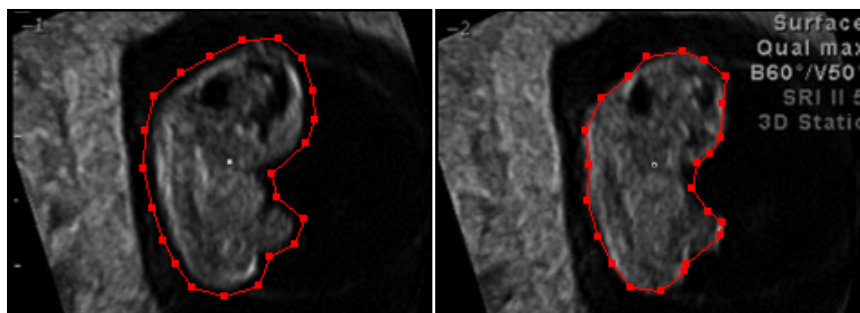
วิธีดำเนินการวิจัย

3.1 กล่าวนำ

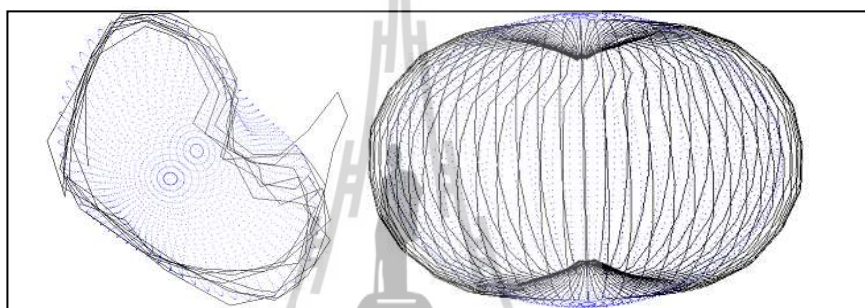
งานวิจัยนี้จะกล่าวถึงการสร้างแบบจำลองสามมิติของทารกในครรภ์จากภาพ Ultrasound ตามหัวข้อวิจัย การสร้าง และวิเคราะห์แบบจำลอง 3 มิติของทารกในครรภ์จากข้อมูลของเครื่องถ่ายภาพเหนือเสียง (THREE – DIMENSIONAL MODEL RECONSTRUCTION OF FETAL ULTRASONOGRAPHY) จะเป็นการนำเอาภาพที่มีอยู่แล้วทั่วไปจากเครื่อง Ultrasound ที่เดิมเป็นแบบสองมิติ มาสร้างเป็นแบบจำลองสามมิติ เพื่อช่วยในการวิเคราะห์ วินิจฉัยความผิดปกติของทารกในครรภ์ เพื่อให้ง่ายต่อการมอง อีกทั้งยังเป็นลดต้นทุนค่าใช้จ่ายในการสั่งซื้ออุปกรณ์ที่มีราคาแพงอีกด้วย โดยในส่วนของ การจำลองพื้นผิวขึ้นมา นั้นปกติแล้วจะใช้เวลามาก ยิ่งข้อมูลที่เข้ามามีมากก็ยิ่งช้าขึ้นไปอีก ดังนั้นในส่วนนี้ได้มีการเพิ่มความสามารถในการแก้ไข และปรับแต่งพื้นผิวเพียงบางจุดที่มีความไม่ถูกต้องได้ (Interactive Surface Editing) เนื่องจากถ้ามีการอ่านข้อมูลเข้ามาใหม่ทั้งหมด แทนที่จะเป็นเพียงส่วนที่แก้ไข จะทำให้ใช้เวลาในการประมวลผลค่อนข้างมาก และอาจทำให้คอมพิวเตอร์บางเครื่องที่มีหน่วยความจำน้อยไม่สามารถใช้งานได้ ซึ่งบทนี้จะอธิบายขั้นตอนงานวิจัยทั้งหมดไว้ โดยแบ่งขั้นตอนเป็น 3 ส่วนคือ (1) การแยกองค์ประกอบของภาพด้วยแอคทีฟคอนทัวร์ (Active Contour) ดังรูปที่ 3.1ก และ 3.1ข (2) การสร้างพื้นผิวสามมิติด้วยจุดและการขยายค่าจริงของฟังก์ชันทรงกลมแบบฮาร์โมนิก (Spherical Harmonics) และ (3) การปรับพื้นผิวเฉพาะบางจุด (Interactive Surface Editing) ในส่วนที่ 2 และ 3 อธิบายด้วยรูปที่ 3.2 เนื่องจากมีขั้นตอนการทำงานคือ แก้ไขส่วนที่ 1 แล้วทำการสร้างพื้นผิวใหม่ในขั้นตอนที่ 2



รูปที่ 3.1ก การแยกองค์ประกอบของภาพในขั้นตอนการหาคอนทัวร์



รูปที่ 3.1 ข การแยกองค์ประกอบของภาพในขั้นตอนการหาเวกเตอร์ตั้งฉาก



รูปที่ 3.2 การซ้อนทับกันของคอนทัวร์และแบบจำลอง

3.2 ระเบียบวิธีวิจัย

ขั้นตอนในการทำงานวิจัย

1) ศึกษาและรวบรวมสรุปงานวิจัยที่เกี่ยวข้อง

2) พัฒนาขั้นตอนวิธีการสร้างแบบจำลองสามมิติแบบกึ่งอัตโนมัติ

2.1) ออกแบบการใช้งานระบบว่าควรมีโมดูลอะไรบ้าง ได้ตอบกับผู้ใช้งานอย่างไร

(Use case Daigram)

2.2) ออกแบบหน้าที่ของแต่ละคลาสเพื่อเขียนแผนภาพคลาส (Class Diagram) ของ

ระบบ

2.3) ศึกษาเทคนิคและวิธีในการเขียนโปรแกรม

3) การสร้างพื้นผิวสามมิติจากข้อมูลภาพสองมิติ

3.1) ศึกษาและพัฒนาการสร้างพื้นผิวจากการกระจายตัวของจุด (Surface

reconstruction from point could)

3.2) ศึกษาและพัฒนาการสร้างพื้นผิวโดยวิธีขยายทรงกลมด้วยค่าจริงของฟังก์ชันทรงกลมแบบฮาร์โมนิก (Spherical Harmonic) ในแต่ละลำดับ (order)

4) ศึกษาและพัฒนากระบวนการปรับแก้ไขพื้นผิวเพียงบางส่วน (Interactive Surface Editing) เมื่อต้องการแก้ไขข้อมูลที่ไม่ถูกต้อง เพื่อเพิ่มความรวดเร็วในการสร้างภาพจำลองแบบสามมิติ

5) ทดสอบระบบที่พัฒนาขึ้นในการวิเคราะห์ความผิดปกติของทารกในครรภ์อย่างง่าย และสรุปผลการทดลอง

3.3 ภาพรวมของระบบที่ทำการพัฒนา

ในงานวิจัยชิ้นนี้ได้นำเสนอวิธีการสร้างภาพจำลองสามมิติโดยใช้ข้อมูลจากเครื่องถ่ายภาพเหนือเสียง (Ultrasound) มาเป็นข้อมูลนำเข้า ซึ่งโดยปกติแล้วจะเป็นภาพแบบสองมิติ โดยอาศัยกระบวนการทางคอมพิวเตอร์กราฟิกมาช่วยในการพัฒนา ซึ่งระบบจะเป็นแบบกึ่งอัตโนมัติโดยผู้ใช้งานจะเป็นผู้กำหนดข้อมูลเริ่มต้นให้แก่ระบบ จากนั้นระบบจะทำการจัดการต่อจนสิ้นสุดกระบวนการสำหรับวิธีการและเทคนิคที่ใช้จะประกอบไปด้วย 3 ส่วนคือ (1) การแยกองค์ประกอบของภาพ (Image Segmentation) เป็นการแยกส่วนของภาพที่สนใจออกจากองค์ประกอบอื่นทั้งหมดของภาพ ดังนั้นเมื่อได้ข้อมูลเหล่านี้มาแล้วจะมาอยู่ในส่วนที่ (2) การสร้างแบบจำลองสามมิติด้วยจุด (Surface Reconstruction from point cloud) และการขยายค่าจริงของฟังก์ชันทรงกลมแบบฮาร์โมนิก (Spherical Harmonics) ในส่วนนี้ระบบจะทำการสร้างแบบจำลองสามมิติขึ้นมา แต่บางครั้งอาจพบว่าแบบจำลองที่ถูกสร้างขึ้นมีการผิดพลาดของข้อมูลเพียงเล็กน้อย จึงเป็นที่มาในส่วนที่ (3) การปรับพื้นผิวเฉพาะบางจุด (Interactive Surface Editing) โดยขั้นตอนการพัฒนาทั้งหมดเขียนได้ดังนี้

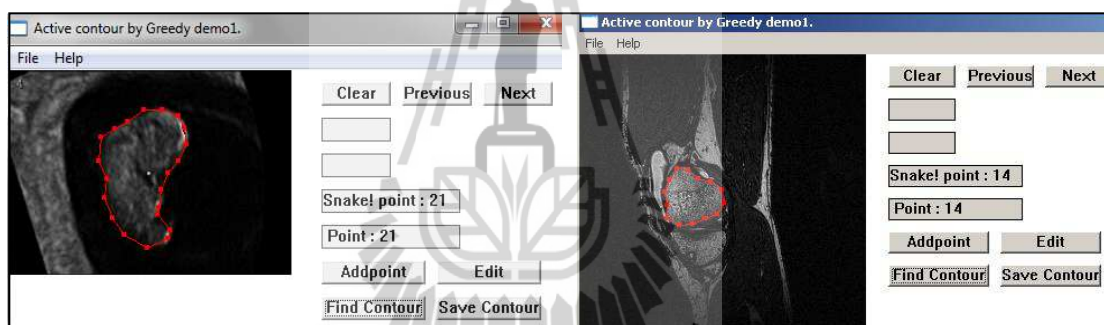
3.3.1 เครื่องมือที่ใช้ในการพัฒนา

สำหรับเครื่องมือที่ใช้ในการพัฒนาระบบนี้ ได้ใช้คอมพิวเตอร์โน้ตบุ๊กส่วนตัวในการพัฒนา เนื่องจากการทำงานเกี่ยวกับการประมวลผลภาพต้องใช้ทรัพยากรของเครื่องค่อนข้างสูง และความสะดวกสามารถพกพาไปทำงานได้ทุกที่ สำหรับโน้ตบุ๊กที่ใช้ในการพัฒนาระบบคือ SAMSUNG รุ่น: R-478 CPU: Intel Core i5 2.26GHz การ์ดจอ: nVidia GeForce GT 330M (1GB GDDR3) แรม: 4 GB DDR3 ระบบปฏิบัติการ: Windows 7 เครื่องมือที่ใช้พัฒนาระบบ: Microsoft Visual Studio 2010, WatcomC++, DevC++ และภาษาที่ใช้คือ C++

3.3.2 โครงสร้างของระบบ

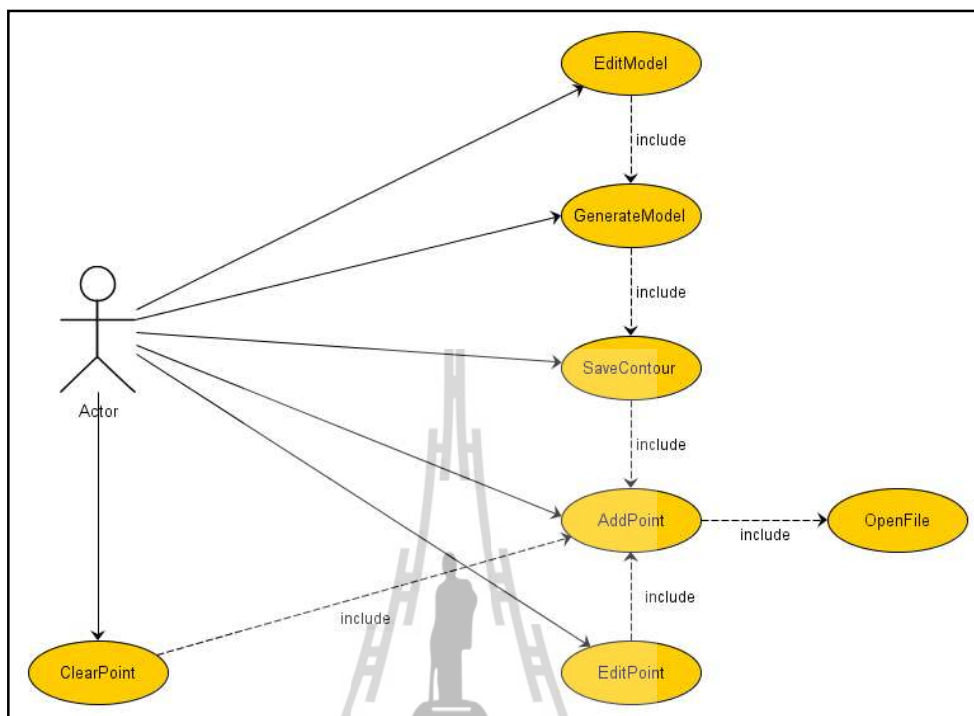
สำหรับระบบการแยกองค์ประกอบของภาพสามารถเขียนเป็น โมดูลย่อย ๆ ดังนี้

- 1) เลือกไฟล์หรือโฟลเดอร์เพื่อนำภาพที่ต้องการแยกองค์ประกอบเข้ามาสู่ระบบ
- 2) แสดงภาพที่นำเข้ามา และกำหนดตำแหน่งจุดรอบวัตถุที่สนใจ เพื่อหาขอบเขตหรือแยกองค์ประกอบต่อไป โดยแบ่งย่อยดังนี้
 - ปุ่มเพิ่มจุด เพื่อกำหนดตำแหน่ง
 - แก้ไขข้อมูล
 - เลื่อนภาพถัดไป ก่อนหน้า
- 3) แยกองค์ประกอบของภาพด้วยปุ่ม FindContour
- 4) เก็บข้อมูลของภาพ
- 5) นำข้อมูลที่ได้มาทำการสร้างแบบจำลองสามมิติ



รูปที่ 3.3 ระบบแยกองค์ประกอบภาพ

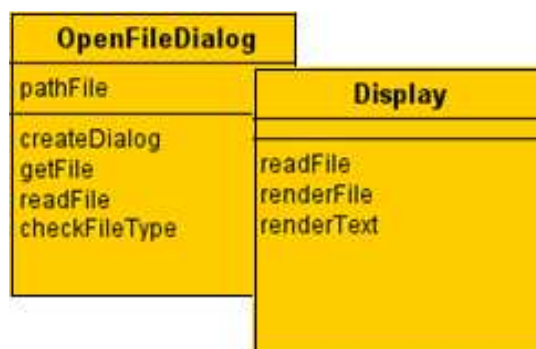
สำหรับการทำงานของระบบสามารถอธิบายด้วยแผนภาพ Use case Diagram ได้ดังนี้



รูปที่ 3.4 Use case diagram ของระบบ

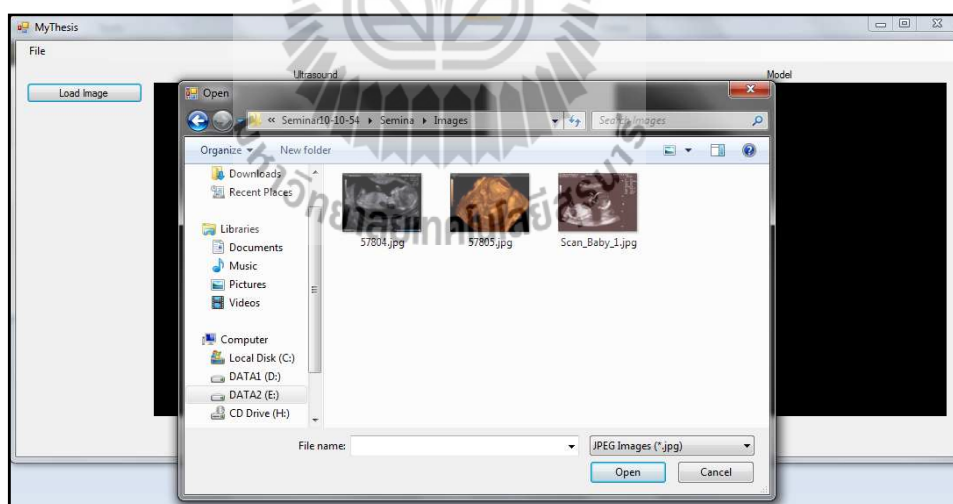
จากแผนภาพจะเห็นได้ว่าผู้ใช้สามารถปรับเปลี่ยนแบบจำลองที่ถูกสร้างขึ้นได้ สร้างแบบจำลองได้ บันทึกข้อมูลของคอนทัวร์ที่หามาได้ เพิ่มจุดเพื่อหาคอนทัวร์ สามารถปรับจุดของคอนทัวร์ที่ไม่ถูกต้องได้ รวมไปถึงการลบข้อมูลคอนทัวร์ทั้งหมดได้ สำหรับการทำงานของระบบนั้น ผู้ใช้จำเป็นต้องทำการเลือกภาพเข้ามาสู่ระบบก่อนเป็นอันดับแรกจึงจะทำงานในขั้นตอนอื่น ๆ ได้ จากนั้นผู้ใช้งานจะต้องทำการกำหนดจุดเริ่มต้นให้แก่ระบบก่อนเพื่อหาคอนทัวร์ของภาพ จึงจะสามารถลบ แก้ไขข้อมูลจุดเหล่านั้นได้ เมื่อทำขั้นตอนดังกล่าวเสร็จก็จะสามารถสร้างเป็นแบบจำลองได้ แต่จะต้องทำการบันทึกข้อมูลจุดหรือคอนทัวร์ของภาพเสียก่อน เมื่อระบบทำการสร้างแบบจำลองขึ้นมาแล้ว หากผู้ใช้พบว่าบางจุดบนคอนทัวร์มีความผิดพลาดหรือไม่ถูกต้อง ก็สามารถที่จะแก้ไขโมเดลได้ แต่ก็ต้องทำการระบวนการดังที่กล่าวมาทั้งหมดก่อน

จากรูปที่ 3.4 Use case diagram ของระบบ ทำให้สามารถแบ่งการทำงานของระบบออกเป็นคลาสตามหน้าที่แบบคร่าว ๆ เพื่อนำไปใช้ในการพัฒนาระบบ โดยอธิบายการทำงานของแต่ละคลาสได้ดังนี้



รูปที่ 3.5 คลาสที่ใช้สำหรับจัดการข้อมูลรูป

1) จากรูปที่ 3.5 คลาส OpenFileDialog เป็นคลาสสำหรับทำการเปิดปิด เลือกไฟล์หรือชุดข้อมูลภาพสองมิติที่สนใจ ซึ่งเป็นภาพที่ได้จากเครื่อง Ultrasound โดยมีตัวแปรที่สำคัญใช้ในการเก็บที่อยู่ของไฟล์ข้อมูลที่ต้องการเปิดหรือทำการหาคอนทัวร์คือ pathFile และมีฟังก์ชันที่ทำหน้าที่สร้าง Dialog เพื่อทำการเลือกไฟล์ ฟังก์ชันที่ใช้ในการเก็บที่อยู่ไฟล์ ฟังก์ชันในการอ่านรูปภาพ และ ฟังก์ชันสำหรับเช็คว่าข้อมูลที่เข้ามาเป็นไฟล์ชนิดใด ผลลัพธ์ดังรูปที่ 3.6

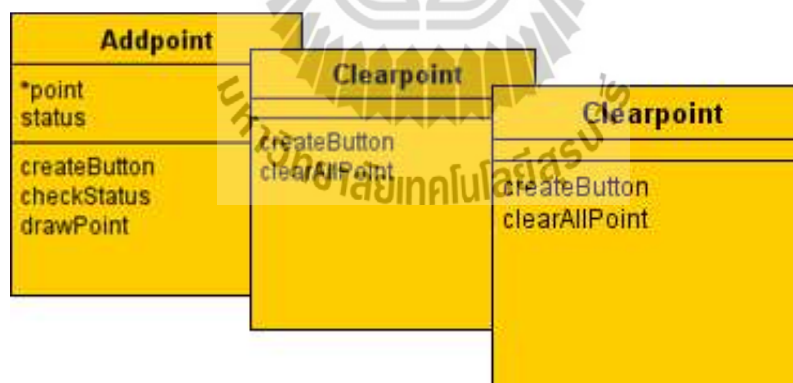


รูปที่ 3.6 OpenFileDialog

2) คลาส Display เป็นคลาสสำหรับการจัดการการแสดงผลภาพที่เลือกเข้ามาสู่ระบบ รวมถึงข้อความต่าง ๆ ที่แสดงบนโปรแกรม โดยจะทำการเก็บค่าความเข้มในแต่ละพิกัด (pixel) ของภาพ ปรับเป็นภาพเกรย์สเกล แล้วทำการวาดข้อมูลเหล่านี้ออกมาแสดงผลเป็นรูปภาพ เพื่อให้ผู้ใช้ใช้งานในขั้นตอนต่อไปดังรูปที่ 3.7



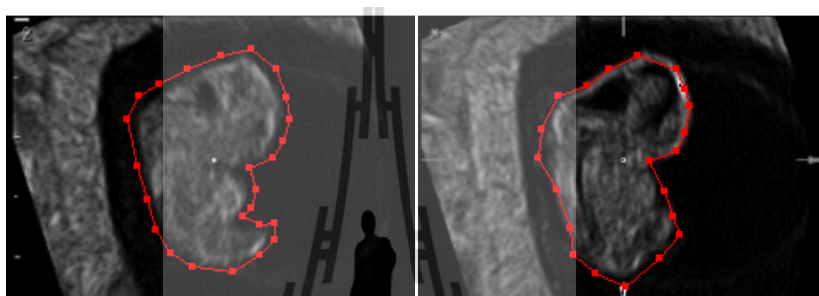
รูปที่ 3.7 เมนูการหาคู่ประกอบของภาพและสร้างแบบจำลองสามมิติ



รูปที่ 3.8 คลาสสำหรับการแยกองค์ประกอบของภาพ

เมื่อแบ่งการทำงานของระบบตามหน้าที่การใช้งานแล้ว รูปที่ 3.8 คือคลาสที่ใช้สำหรับการแยกองค์ประกอบของภาพ จะทำงานในส่วนของการหาคอนทัวร์ หรือกลุ่มของจุดรอบวัตถุที่สนใจ เพื่อนำไปใช้ในขั้นตอนต่อไป โดยสามารถอธิบายได้ดังนี้

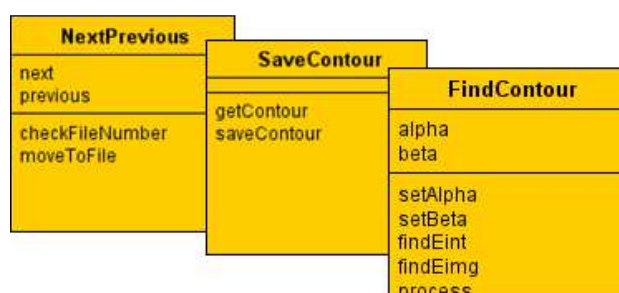
3) Addpoint เป็นคลาสสำหรับการเพิ่ม หรือกำหนดจุดจากผู้ใช้ลงไปยังภาพ โดยผู้ใช้งานจะต้องกำหนดจุดเหล่านี้ลงไปรอบวัตถุที่เราสนใจลงไป อาจจะใกล้เคียงกับขอบของวัตถุที่สนใจหรือไม่ก็ได้ เพราะระบบจะทำการหาขอบอัตโนมัติอยู่แล้ว แต่ก็ต้องระวังโดยอาจไปวงรอบวัตถุอื่นจนทำให้โปรแกรมจับขอบภาพออกมาผิดก็เป็นได้ สำหรับวิธีคือเริ่มจากกำหนดจุด ๆ บนภาพเป็นจุดเริ่มต้น จากนั้นก็ทำการวาดเป็นเส้นรอบวัตถุ อาจใกล้เคียงมากหรือน้อยก็ได้แต่ความสะดวกของผู้ใช้ แต่ต้องระวังว่าอย่าไปใกล้กับส่วนที่เป็นขอบขององค์ประกอบอื่นที่เราไม่สนใจ และค่อย ๆ ทำไปจนจุดสุดท้ายและจุดแรกเป็นจุดเดียวกันเป็นอันเสร็จขั้นตอนนี้



รูปที่ 3.9 ตัวอย่างภาพการหาคอนทัวร์ที่ได้จากการเพิ่มและปรับจุด

4) Editpoint เป็นคลาสสำหรับเมื่อเราทำการวาดรูปในขั้นตอนก่อนหน้าเสร็จเรียบร้อยแล้วเกิดมีจุดใดจุดหนึ่งที่สังเกตว่าไม่ตรงกับขอบของวัตถุจริง หรืออาจไปยึดกับขอบขององค์ประกอบของภาพอื่น ก็ทำการแก้ไขได้โดยคลาสนี้

5) Clear เป็นคลาสสำหรับเมื่อระบบแยกองค์ประกอบของภาพผิดพลาดซึ่งไม่ยึดติดกับขอบ หรือไปยึดกับขอบของวัตถุอื่นที่ไม่สนใจ เนื่องจากตัวระบบเองหรือการกำหนดจุดเริ่มต้นจากผู้ใช้งานผิดพลาด สามารถทำการเคลียร์ค่าทั้งหมดได้ และทำการเริ่มต้นใหม่



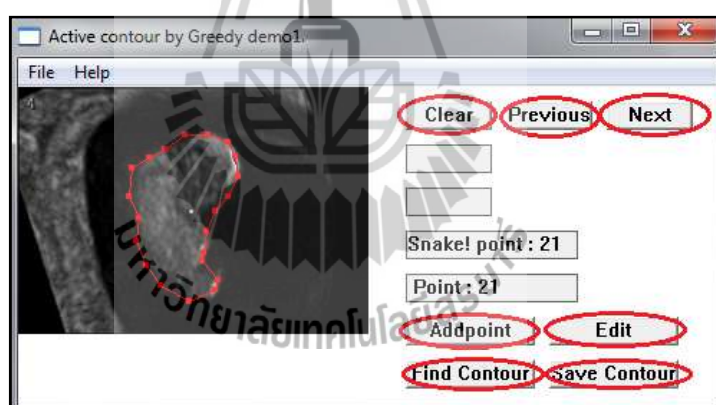
รูปที่ 3.10 คลาสสำหรับการจัดการการเก็บข้อมูลต่างๆ บนคอนทัวร์

จากรูปที่ 3.10 คือคลาสที่ใช้สำหรับการเก็บข้อมูลของคอนทัวร์ที่หามาได้ ซึ่งในการหาคอนทัวร์ของแต่ละรูปนั้น จำเป็นต้องทำการบันทึกข้อมูลของเสียก่อน และทำการเลื่อนรูปภาพเพื่อทำการหาคอนทัวร์ของภาพต่อไป อธิบายได้ดังนี้

6) NextPrevious เป็นคลาสสำหรับการเปลี่ยนรูปภาพไปสู่รูปก่อนหน้า หรือรูปถัดไป เนื่องจากข้อมูลที่น่าเข้ามาสู่ระบบ จะเป็นชุดของข้อมูลภาพ โดยปกติแล้วจะเป็นภาพสองมิติที่มีความกว้าง และความยาวเท่ากัน แต่ภาพที่ได้จะแตกต่างกันคือ จะนำเอาภาพสองมิติเหล่านี้จากแต่ละมุมของเครื่อง Ultrasound มาทำการเรียงกันจนเสมือนว่ากลายเป็นส่วนลึก จะทำให้จุดหรือคอนทัวร์ที่ได้เหล่านี้กลายเสมือนเป็นข้อมูลจุดในรูปแบบสามมิติ

7) FindContour จะเป็นคลาสที่สำคัญ โดยจะเป็นการแยกองค์ประกอบที่สนใจออกจากองค์ประกอบอื่นของภาพ

8) SaveContour เป็นคลาสสุดท้ายของกระบวนการแยกองค์ประกอบของภาพ ซึ่งจะเป็นตัวจัดการข้อมูลที่เราหาได้ทั้งหมด โดยจะเป็นการจัดเก็บข้อมูลของภาพ หรือคอนทัวร์ที่เราได้มาเพื่อใช้งานต่อไป



รูปที่ 3.11 ปุ่มที่สร้างจากคลาสต่างๆ

GenerateModel	CheckChangePoint
contour	changePoint
SHpoint	
norContour	getPoint
changeToSpherical	setPoint
findCim	checkPointinContour
findSHPoint	
generateSurface	

รูปที่ 3.12 คลาสสำหรับจัดการในส่วนของแบบจำลองสามมิติ

เมื่อทำการหาข้อมูลคอนทัวร์ของทุกรูปแล้ว ขั้นตอนต่อไปคือการนำคอนทัวร์เหล่านี้มาทำการสร้างแบบจำลอง 3 มิติ โดยคลาสที่จำเป็นสำหรับการสร้างโมเดลรูปที่ 3.12 ซึ่งสามารถอธิบายได้ดังนี้

9) GenerateModel เป็นคลาสในการนำข้อมูลที่หาได้ทั้งหมดในขั้นตอนด้วยการแยกองค์ประกอบของภาพไปใช้สำหรับการสร้างแบบจำลองสามมิติด้วยวิธีการขยายฟังก์ชันทรงกลมแบบฮาร์โมนิก โดยแบบจำลองที่ถูกสร้างขึ้นอาจมีการเปลี่ยนแปลงอีกครั้งเมื่อผู้มีการแก้ไขข้อมูลจากคลาส Editpoint ซึ่งเป็นการปรับแก้ไขข้อมูลใหม่ เพื่อให้ได้ข้อมูลที่ต้องการ มีผลทำให้แบบจำลองสามมิติมีการเปลี่ยนแปลง

10) CheckChangePoint เป็นคลาสที่ใช้ในเช็คว่า เมื่อผู้มีการเปลี่ยนตำแหน่งจุดข้อมูลบนคอนทัวร์ที่หาได้ จะทำการเช็คว่าเป็นจุดใดบนคอนทัวร์และอยู่ในสไลด์ที่เท่าใด จากนั้นระบบจะทำการสร้างพื้นผิวขึ้นมาใหม่เฉพาะในส่วนที่มีการเปลี่ยนแปลงเท่านั้น



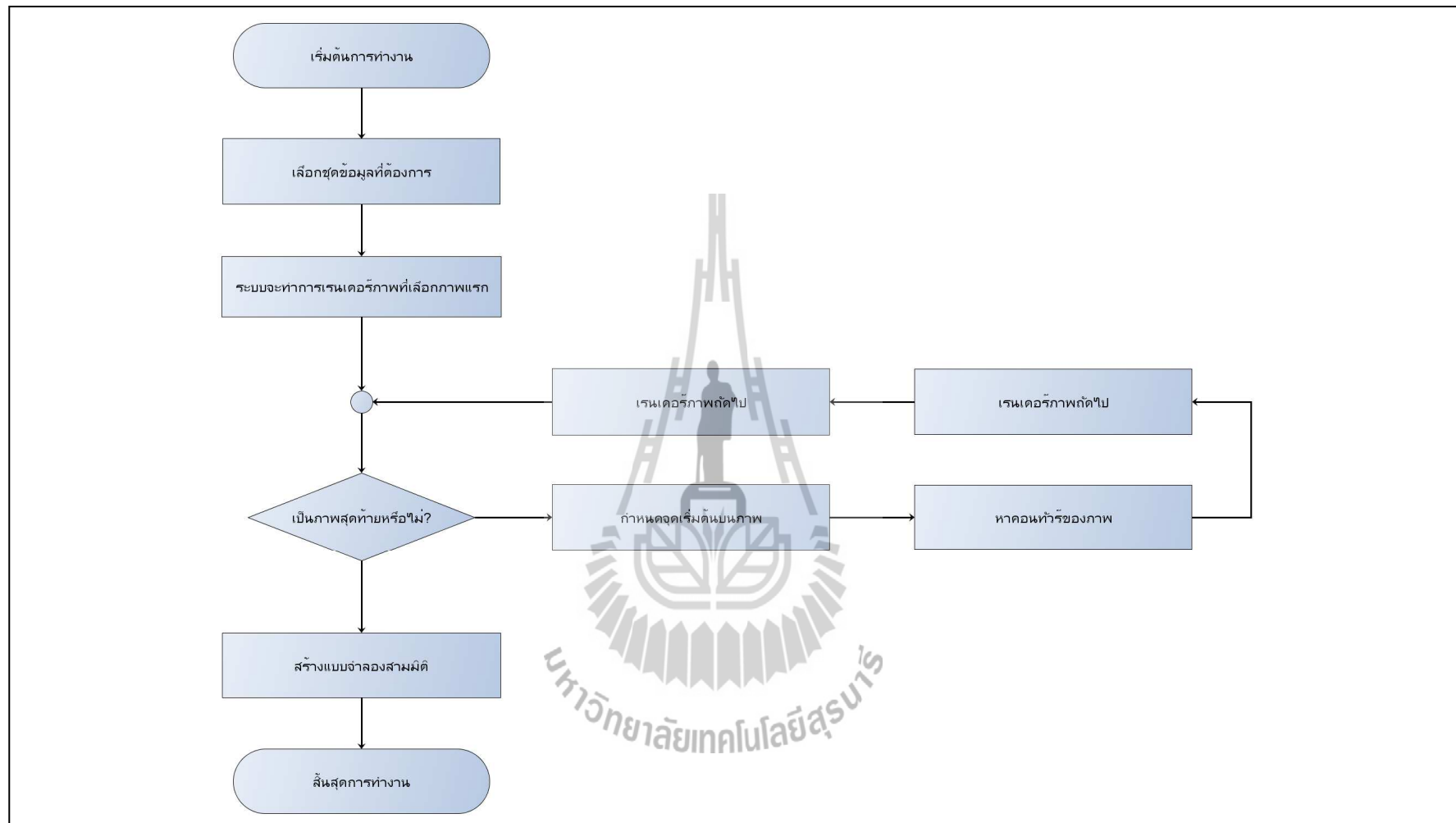
รูปที่ 3.13 หากคอนทัวร์เสร็จสิ้นระบบจะสร้างปุ่ม Create Model



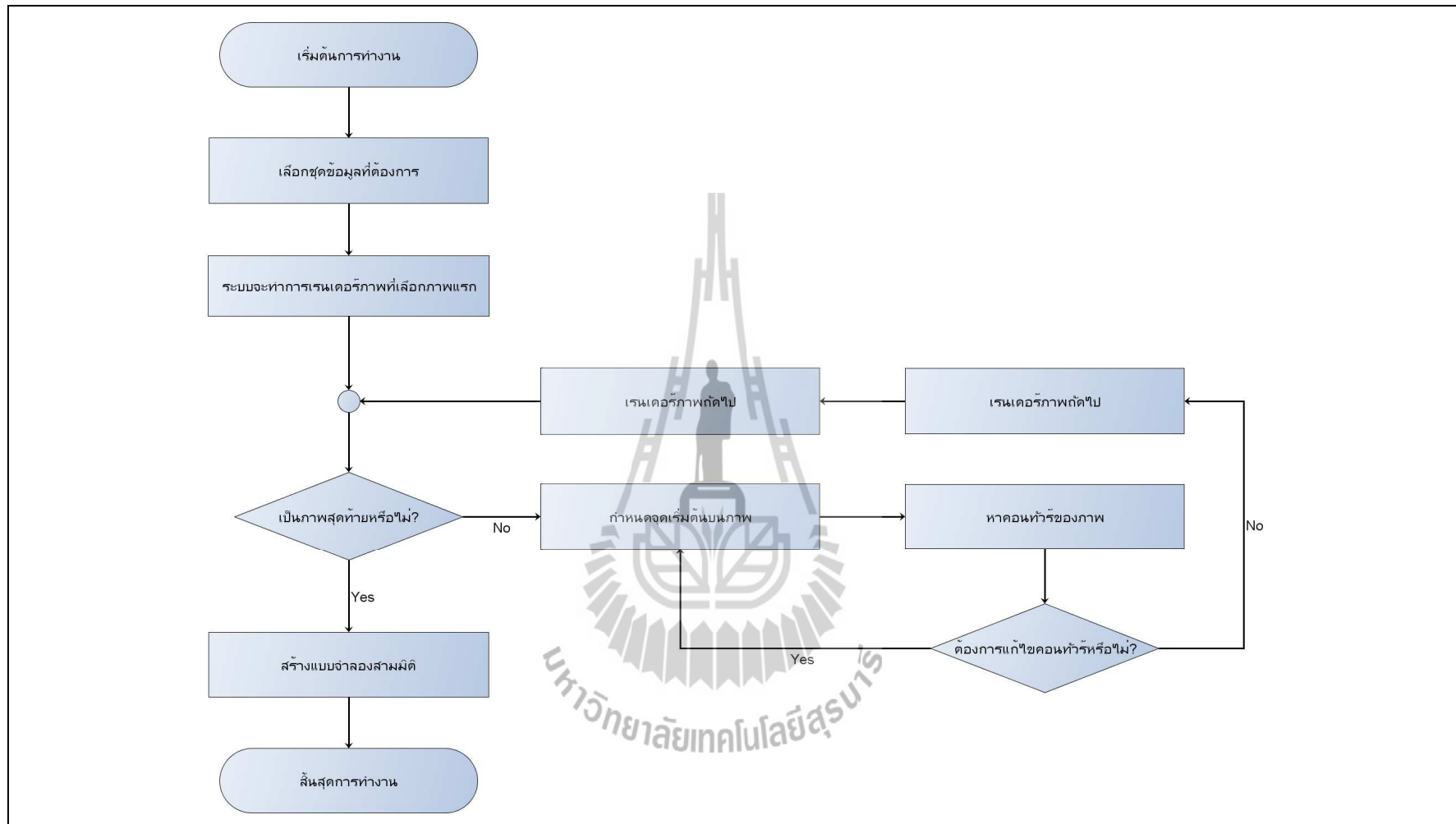
รูปที่ 3.14 สร้างแบบจำลองสามมิติ

จากรูปที่ 3.13 เมื่อทำการหาคอนทัวร์ของทุกรูปได้แล้วจะมีปุ่ม Create Model เกิดขึ้นดังนั้น เมื่อผู้ใช้กดปุ่มแบบจำลองสามมิติจะถูกสร้างดังรูปที่ 3.14

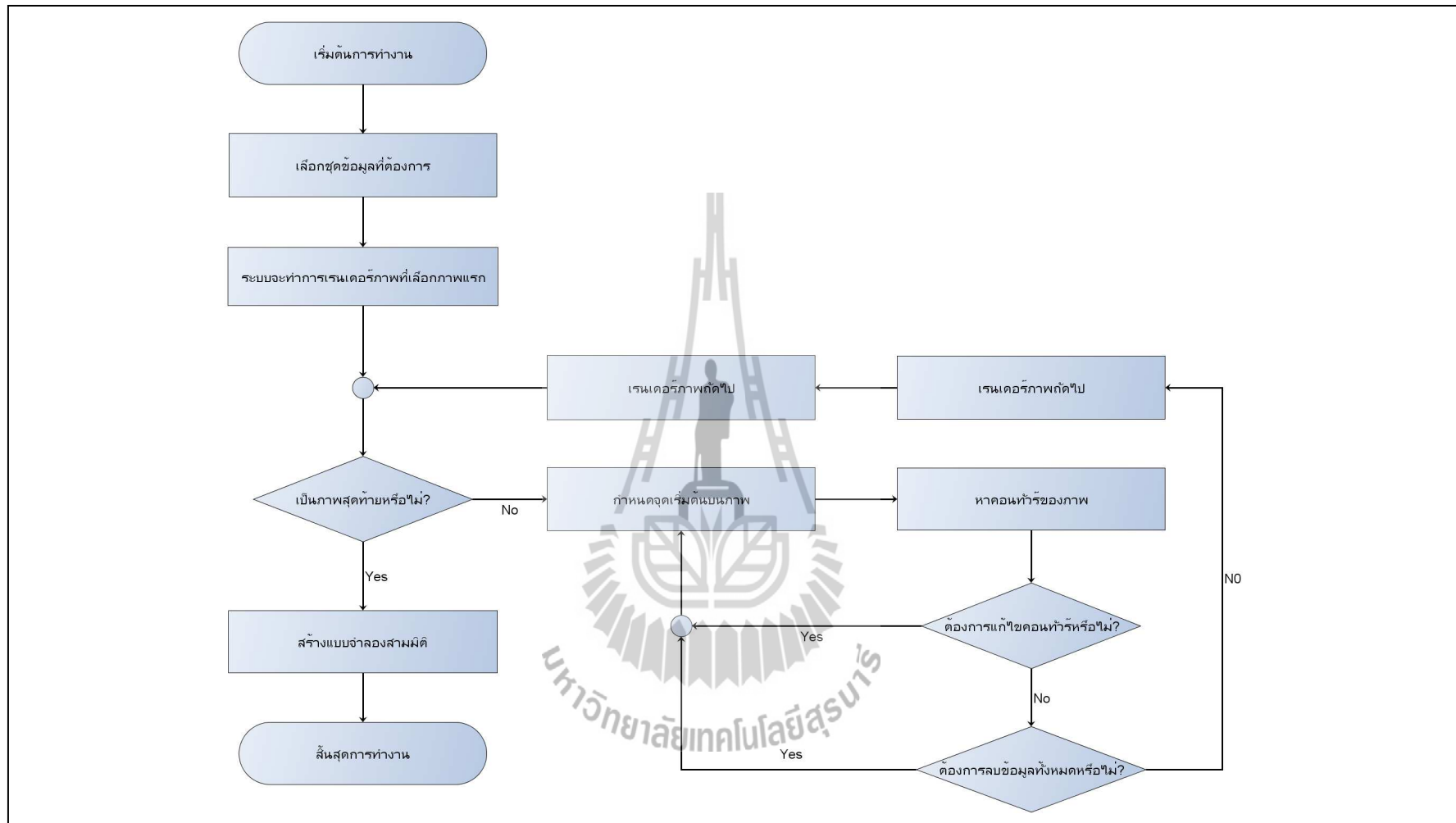
จากภาพรวมการทำงานของระบบ ทำให้สามารถเขียนเป็นแผนภาพขั้นตอนการทำงานของโปรแกรม (Flow chart) ด้วยระบบแผนภาพทางวิศวกรรมซอฟต์แวร์ (Software Engineering) โดยการทำงานของระบบจะเริ่มจากการเปิดชุดข้อมูลภาพที่เราสนใจ แสดงผลภาพที่เลือกมาเพื่อกำหนดตำแหน่งของจุดรอบวัตถุที่สนใจ จนถึงขั้นตอนการสร้างแบบจำลองสามมิติและการแก้ไขพื้นผิวสามารถแบ่งเป็น 4 กรณีคือ (1) เริ่มจากกระบวนการแรกจนถึงกระบวนการสุดท้ายโดยไม่มีการแก้ไขใด ๆ (2) เริ่มจากกระบวนการแรกและมีการแก้ไขการกำหนดจุดเริ่มต้นบนคอนทัวร์ (3) เมื่อกำหนดจุดครบทุกจุดแต่เกิดข้อผิดพลาดกับคอนทัวร์ที่หาได้จึงต้องมีการเคลียร์ค่าใหม่ทั้งหมด และ (4) มีการปรับเปลี่ยนในจุดของพื้นผิวเพียงบางตำแหน่ง ทั้งหมดเขียนเป็นแผนภาพลำดับการทำงานได้ดังนี้



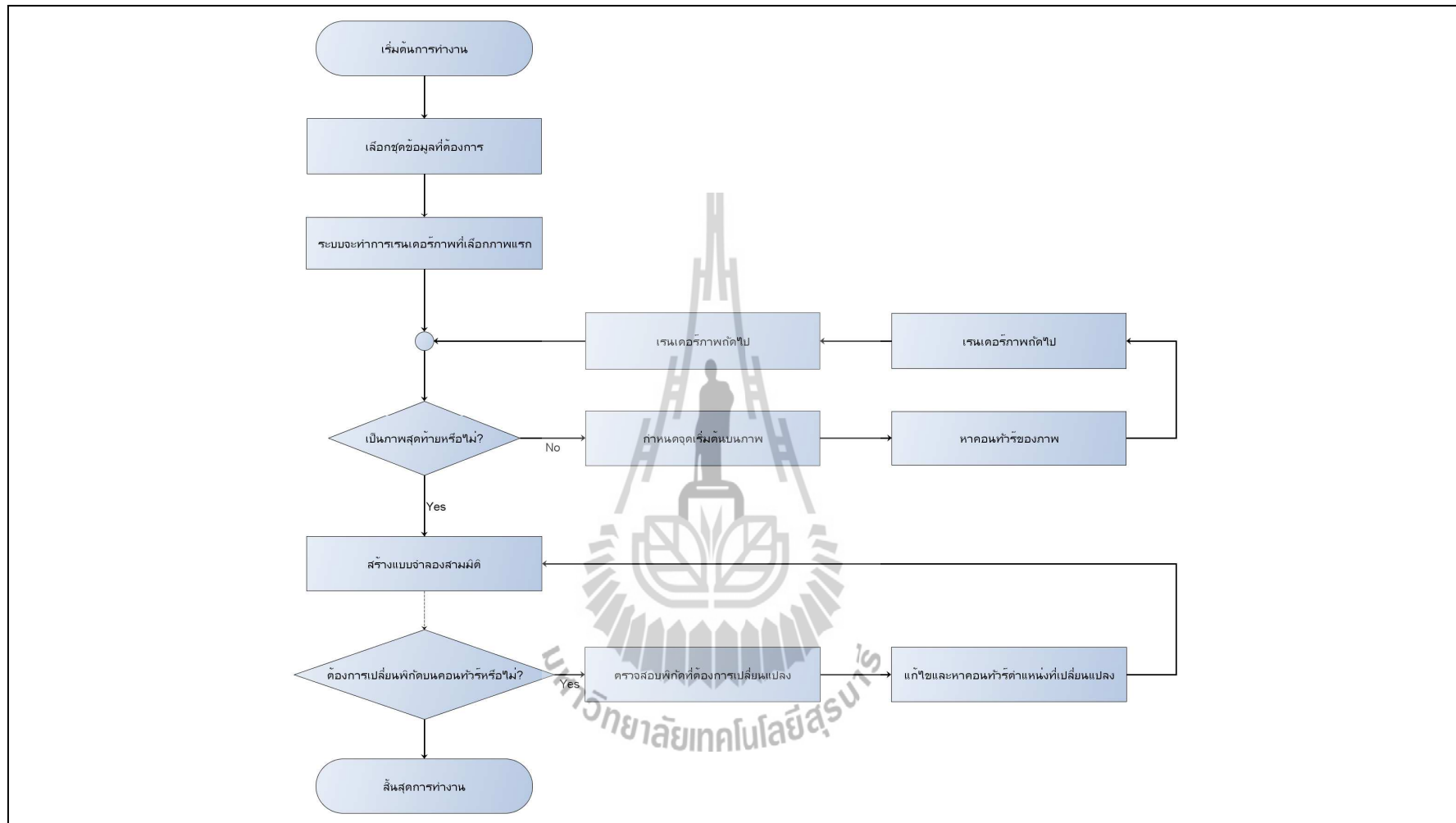
รูปที่ 3.15 ไม่มีการแก้ไขใด ๆ บนระบบ



รูปที่ 3.16 มีการแก้ไขจุดบนคอนทัวร์ที่ทำได้



รูปที่ 3.17 มีการลบจุดทั้งหมดบนคอนทัวร์ออก



รูปที่ 3.18 มีการแก้ไขตัวแบบจำลองสามมิติ

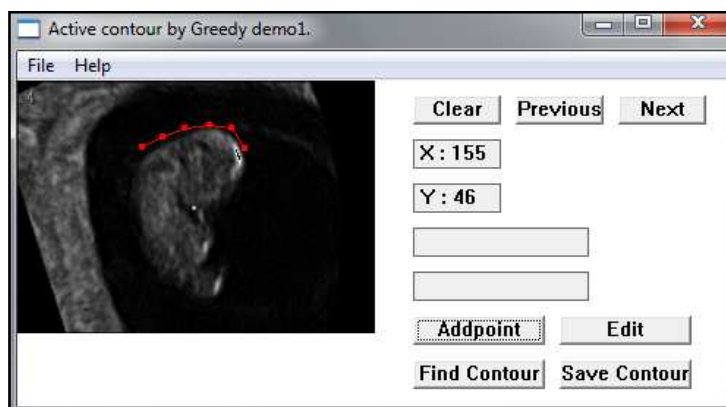
จากรูปที่ 3.15 การทำงานเริ่มจากผู้ใช้เลือกไฟล์เข้ามาจากนั้นระบบจะทำการแสดงภาพบนหน้าจอเพื่อให้ผู้ใช้กำหนดจุดบนภาพรอบวัตถุที่สนใจ เมื่อผู้ใช้กำหนดจุดครบ ก็จะทำการหาคอนทัวร์ของภาพ และหาจนกว่าจะครบทุกสไลด์ของภาพ เมื่อผู้ใช้หาครบทุกภาพแล้วจะสามารถสร้างเป็นโมเดล 3 มิติออกมาได้

จากรูปที่ 3.16 ในขั้นตอนของการหาคอนทัวร์ เมื่อหาคอนทัวร์ออกมาแล้วผู้ใช้จะเห็นว่า มีข้อผิดพลาด เช่น อาจมีบางจุดบนคอนทัวร์ที่หาไม่ได้ไม่เหมาะสม หรือไม่ชิดกับขอบของวัตถุ ซึ่งอาจเกิดจากการกำหนดค่าเริ่มต้นผู้ใช้จะกำหนดจุดไปทับกับขอบขององค์ประกอบอื่น ดังนั้นระบบจึงได้มีการสร้างส่วนแก้ไขขึ้นมา โดยผู้ใช้สามารถปรับจุดใหม่ ณ ที่จุดนั้นได้ และโปรแกรมจะทำการหาคอนทัวร์ใหม่ จากนั้นก็จะกลับเข้าสู่ระบบเดิมในการสร้างภาพสามมิติ

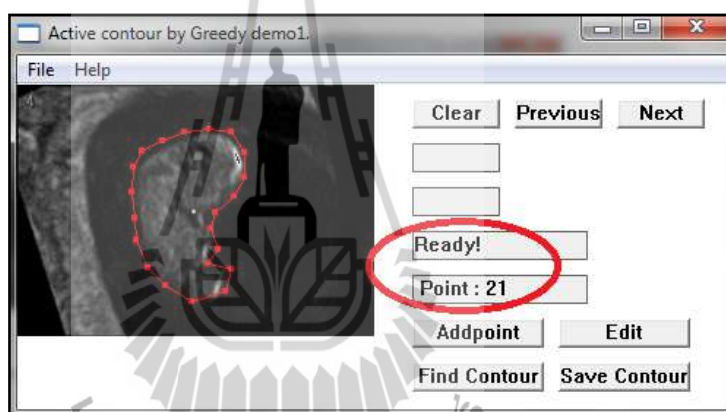
จากรูปที่ 3.17 จะเห็นว่า มีบางกรณีที่ผู้ใช้ทำการกำหนดจุดแล้ว แต่ผลลัพธ์ออกมาไม่ตรง หรือมีการแก้ไขแล้วยังคงไม่ตรง ดังนั้นระบบจึงรองรับส่วนนี้โดยการสร้างฟังก์ชันการเคลียร์จุดทั้งหมดใหม่เพื่อให้ผู้ใช้ได้เริ่มการกำหนดจุดใหม่ทั้งหมดบนคอนทัวร์ ก่อนทำการสร้างภาพจำลองสามมิติต่อไป

จากรูปที่ 3.18 จะเห็นว่าเมื่อเราสร้างแบบจำลองสามมิติออกมาแล้วมีบางพื้นที่ไม่เป็นไปตามที่ควรจะเป็น ดังนั้นระบบจึงได้สร้างฟังก์ชันของการแก้ไขพื้นผิวบางส่วนขึ้นมา โดยการให้ผู้ใช้สามารถกลับไปแก้ไขจุดบนคอนทัวร์ที่มีการเปลี่ยนแปลง จากนั้นระบบก็จะดำเนินการต่อ โดยจะสร้างแบบจำลองสามมิติขึ้นมาใหม่เพียงบางส่วนเท่านั้นเพื่อลดระยะเวลาการทำงานของระบบ และเพิ่มประสิทธิภาพให้ระบบด้วย

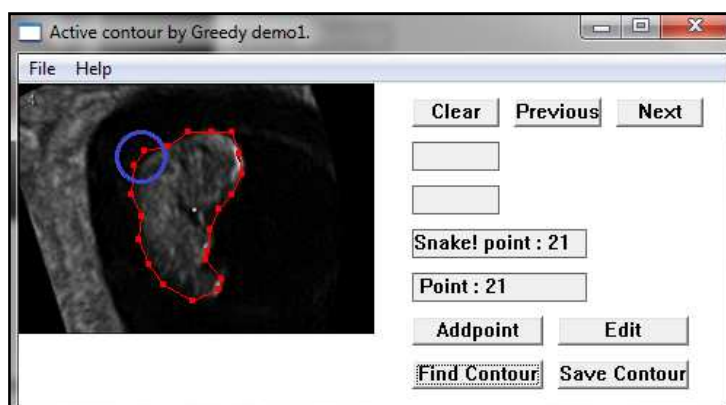
สรุปขั้นตอนการทำงานอย่างของระบบแยกองค์ประกอบของภาพ จะเริ่มจากการเปิดเลือกไฟล์ภาพจากนั้นจะเป็นการเพิ่มจุดบนภาพเพื่อกำหนดค่าเริ่มต้นจากผู้ใช้ดังรูปที่ 3.16 จากนั้นเมื่อกำหนดจุดครบจะมีข้อความบอกผู้ใช้ว่ากำหนดข้อมูลเรียบร้อยแล้ว และบอกจำนวนจุดที่กำหนดว่ามีกี่จุดรูปที่ 3.17 เมื่อกำหนดจุดครบแล้วก็สามารถหาคอนทัวร์ของภาพได้จากปุ่ม Find Contour จะแสดงผลดังรูปที่ 3.18 และในภาพจะเห็นว่า มีบางจุดที่ไม่ชิดกับขอบ จะสามารถทำการแก้ไขได้โดยกดปุ่ม Edit สามารถปรับยึดจุดรูปที่ 3.19 ได้ จากนั้นกดปุ่มหาคอนทัวร์อีกครั้งจะได้ผลลัพธ์ดังรูปที่ 3.20 แต่ถ้าหากต้องการเคลียร์ค่าทุกจุดกดปุ่ม Clear และ Next, Previous จะเป็นการเลื่อนภาพก่อนหน้าและถัดไป



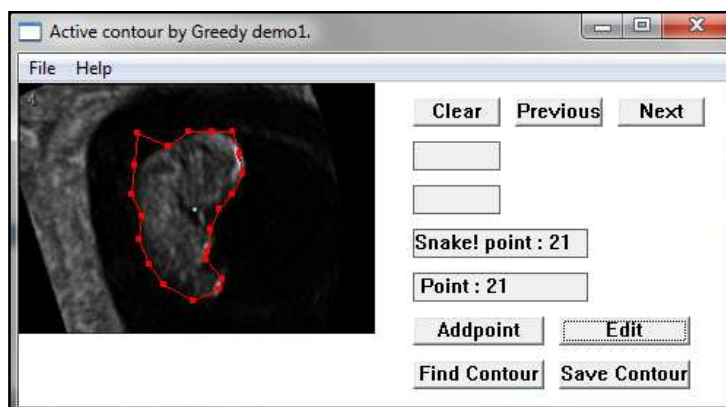
รูปที่ 3.19 AddPoint



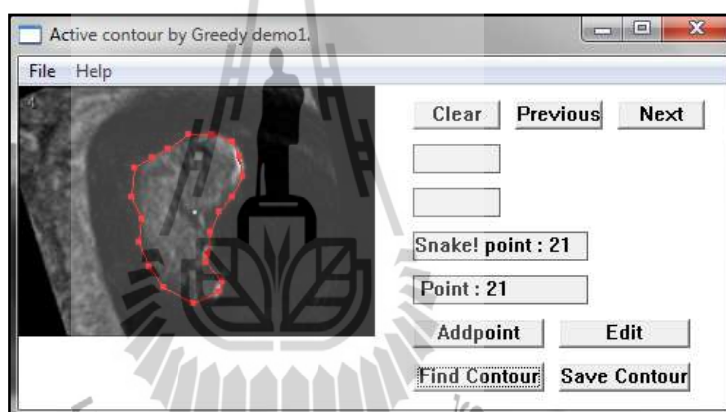
รูปที่ 3.20 ข้อความบอกผู้ใช้ว่าสามารถหาคอนทัวร์ได้แล้ว พร้อมกับจำนวนจุดข้อมูล



รูปที่ 3.21 มีข้อมูลที่ไม่ยึดติดกับขอบรูป



รูปที่ 3.22 แก้ไขข้อมูลจากปุ่ม Editpoint



รูปที่ 3.23 ผลลัพธ์การรันโปรแกรม

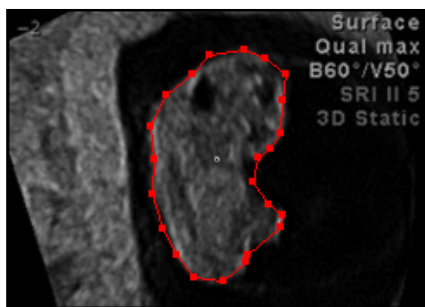
3.4 การแยกองค์ประกอบของภาพด้วยแอคทีฟคอนทัวร์ (Active Contour)

ระบบนี้เป็นขั้นตอนหนึ่งที่สำคัญมากสำหรับการสร้างภาพจำลองสามมิติ เพราะข้อมูลที่ได้มาจากเครื่อง Ultrasound นั้นเป็นภาพสองมิติที่มีองค์ประกอบที่ไม่ต้องการรวมอยู่ในภาพด้วย เช่น เนื้อเยื่อ กระดูก อวัยวะข้างเคียง หรือแม้กระทั่งสัญญาณรบกวน (Noise) ที่ขึ้นในขณะถ่ายภาพ ด้วยเหตุนี้จึงจำเป็นต้องใช้การแยกองค์ประกอบภาพมาทำหน้าที่ตัดแยกส่วนที่ต้องการออกมา ซึ่งในหัวข้อนี้จะนำเสนอระบบการแยกองค์ประกอบของภาพที่ได้รับความนิยมอย่างมากคือ แอคทีฟคอนทัวร์ (Active Contour) โดยจะอธิบาย

วิธีการที่ใช้ในขั้นตอนการแยกส่วนประกอบของคอนทัวร์นี้จะใช้เทคนิคที่นิยมคือ Snake Active Contour Model โดยปกติแล้วการแยกองค์ประกอบของภาพ จะอธิบายได้โดยสมการของพลังงานซึ่งถูกนิยามด้วยสมการ 2-1 และ 2-2 โดยปกติแล้วในภาพแต่ละภาพแต่ละจะมีพลังงานที่อยู่ภายใน เช่น ความสว่างของแสงแต่ละพิกเซล (pixel) ความต่อเนื่องของลายเส้นขอบรูปภาพ ขอบของวัตถุภายในภาพ หรือภาพที่มีความโค้งเว้าลึกตื้นเองก็ล้วนแต่ถูกนิยามไว้ด้วยสมการของพลังงานทั้งสิ้น จากสมการที่ 3-2 เราสามารถแบ่งพลังงานของรูปออกเป็น 3 เทอมด้วยกัน ซึ่ง $E_{int}(v(s))$ คือ พลังงานภายในที่เกิดจากการหาขอบของรูป เมื่อผู้ใช้งานกำหนดจุดรอบวัตถุที่เราสนใจเสร็จสิ้น เมื่อระบบเริ่มทำการประมวลผล จะเริ่มมีการดึงจุดเหล่านี้เข้าไปหาขอบของวัตถุโดยอัตโนมัติ ซึ่งสิ่งที่ใช้ในการดึงก็คือพลังงานนี้เอง โดยจะใช้มากน้อยต่างกันไปในแต่ละจุด หรือแต่ละบริเวณ ดังนั้น $E_{int}(v(s))$ จึงถูกนิยามด้วยสมการ 2-3 ในส่วนนี้จะมีสัมประสิทธิ์ที่สำคัญอยู่ 2 ตัวคือ α และ β โดย α จะเป็นตัวกำหนดแรงที่ใช้ยึดขอบของวัตถุ และ β จะเป็นตัวกำหนดความโค้งเว้า ทั้งสองตัวนี้เป็นตัวควบคุมสมการของพลังงานภายในภาพ โดยมีความต่อเนื่อง แต่ถ้าหากจุดใดที่มีความโค้งเว้ามากจนอาจเป็นมุมหักค่า β จะเป็น 0 เพราะฉะนั้นภาพแต่ละภาพทั้ง α และ β ผู้ใช้จะต้องปรับให้ดี ซึ่งจะมีผลต่อการเปลี่ยนแปลงของคอนทัวร์ หรือขอบของวัตถุที่เราได้ เช่น ถ้าผู้ใช้กำหนดจุดให้ภาพที่มีความเว้ามากก็เซตค่าให้ β ต่ำ ๆ ไว้จนอาจเป็น 0 ค่าของรูปปกติก็จะอยู่ในช่วง 0 – 1 ซึ่งในงานวิจัยชิ้นนี้สังเกตได้จากรูปที่ 3.24 เป็นรูปทรงกลมธรรมดา ไม่มีส่วนโค้งเว้าแสดงว่าไม่มีจุดที่มุมหักงอ และรูปทรงมีความต่อเนื่อง จึงปรับค่าของ α และ β ให้เป็นหนึ่ง



รูปที่ 3.24 รูปทรงปกติ



รูปที่ 3.25 ลักษณะของภาพที่มีโค้งเว้าลึก

ส่วนรูปที่ 3.25 จะสังเกตเห็นความไม่ต่อเนื่องของรูป ซึ่งมีทั้งส่วนที่เป็นโค้งเว้า ภาพในบางส่วนก็หายไป ดังนั้นจึงปรับ α และ β เข้าใกล้ 0 เพื่อให้ได้คอนทราสต์ที่ต้องการ ดังนั้นผู้ใช้จึงต้องทำการปรับค่า α และ β ใหม่ตามเห็นสมควรเพื่อให้ได้วัตถุที่สนใจจริงๆ สำหรับพลังงานที่อยู่ใดภายในภาพเองก็มีเช่นกัน เช่น ขอบของภาพ บริเวณที่ความเข้มแสงต่างกันมาก ๆ จะทำให้เราทราบว่าเป็นขอบนั้นเป็นขอบเนื่องจากมีความไม่ต่อเนื่อง และต่างกันชัดเจน เป็นต้น

นิยามด้วยสมการ 2-4 โดยในระบบเองจะใช้เพียงในส่วนของ $W_{edge}E_{edge}$ ซึ่งมากพอแล้วสำหรับการหาพลังงานในส่วนนี้ เกิดจากการหาความเข้มของแสง ณ พิกัดต่าง ๆ แล้วครอบด้วยเทมเพลตที่กำหนดไว้โดยซึ่งเทมเพลตที่ใช้ในการคำนวณก็จะแตกต่างกันไปในแต่ละวิธีโดยผลการทดสอบแต่ละวิธีได้รูปที่ 3.26, 3.27, 3.28 และ 3.29 (ที่มา : <http://ajgo.blogspot.com/2013/02/edge-detection.html> [13 July 2010])



รูปที่ 3.26 ภาพต้นแบบ



รูปที่ 3.27 หาขอบด้วยโซเบล (Sobel)



รูปที่ 3.28 หาขอบด้วย (Laplacian)



รูปที่ 3.29 ภาพต้นแบบ (Canny)

ส่วนเทอมสุดท้ายคือ $E_{con}(v(s))$ คือ แรงที่มาจากภายนอก เช่น เมื่อผู้ใช้มีเปลี่ยนแปลงจุดบนคอนทัวร์ ก็จะทำให้เกิดพลังงานในการดึงเส้นเชื่อมของจุดแต่ละจุดบนคอนทัวร์ เป็นต้น

ดังนั้นการแยกองค์ประกอบของภาพด้วยวิธีแอคทีฟคอนทัวร์คือ (Active Contour) การหาขอบของวัตถุ หรือขอบเขตของวัตถุที่เราสนใจนั่นเอง ซึ่งใช้หลักการลดลงของพลังงานภายใน หรือภายนอกที่มากระทำกับวัตถุที่เราสนใจ โดยผู้ใช้จำเป็นต้องกำหนดตำแหน่ง หรือจุดบนคอนทัวร์ของภาพที่ต้องการหาขอบเขต จากนั้นระบบจะจัดการหาขอบเขตเองโดยอัตโนมัติ (semi-automatic) ซึ่งสามารถเขียนจำลองการทำงานได้ดังนี้

```
numOfPoint // จำนวนจุดบนทั้งหมดบนคอนทัวร์

for(round = 0; round <= n; round++) { //กำหนดรอบของการค้นหาว่าจะทำงานกี่รอบ
    for( i = 0 ; i < numOfPoint; i++)
    {
        for(j = i - 1 ; j <= i + 1; j++)
            for(k = i - 1; k <= i + 1; k++)
                
$$E_{snake}^* = \int_0^1 E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s)) ds$$

                
$$\text{if } E_{snake_{j,k}}^* < E_{snake_i}^* )$$

                Moveto j,k
    }
}
```

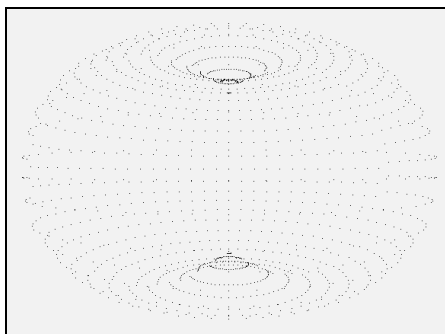

เริ่มจากเมื่อผู้ใช้งานกำหนดจุดครบทุกจุดรอบคอนทัวร์แล้ว ระบบจะทำการวนจากจุดแรกไปยังจุดสุดท้ายบนคอนทัวร์ โดยจะทำการเช็คพิกัด (Pixel) ที่อยู่รอบตัวอีก 8 พิกัด จากนั้นจะทำการเช็คหาตำแหน่งที่มีพลังงานภายในน้อยที่สุด แล้วทำการเลื่อนตำแหน่งไปยังตำแหน่งนั้นแทน โดยจะทำการเท่ากับตามจำนวนรอบที่ระบบกำหนด ดังนั้นถ้าต้องการความถูกต้องเพิ่มขึ้น ก็ต้องเพิ่มรอบตามไปด้วย ทำให้เวลาในการคำนวณก็มากตามไปด้วยเช่นกัน ดังนั้นเพื่อเป็นการลดเวลาตัวผู้ใช้งานกำหนดจุดเริ่มต้นใกล้เกี่ยวกับขอบของวัตถุ เพื่อความถูกต้อง และรวดเร็ว อีกทั้งยังเป็นการเพิ่มประสิทธิภาพให้กับระบบอีกด้วย

3.5 การสร้างพื้นผิวสามมิติด้วยจุด (Surface Reconstruction from point cloud) และการขยายพื้นผิวด้วยค่าจริงของฟังก์ชันทรงกลมแบบฮาร์โมนิก (Spherical Harmonics)

จากขั้นตอนของการแยกองค์ประกอบของภาพ จะได้คอนทัวร์ของภาพ หรือเซตของจุดที่อยู่รอบวัตถุที่เราสนใจ แต่เนื่องจากข้อมูลภาพสองมิติที่ได้มามีข้อมูลที่ถ่ายในแนวลึกด้วย ดังนั้นเซตของจุดที่ได้จึงอยู่ในลักษณะของจุดในสามมิติตามแนวแกน (x, y, z) หรือก็คือจุดที่อยู่ในลักษณะที่เป็นรูปทรงของภาพจำลองสามมิติที่เราต้องการสร้างนั่นเอง ขั้นตอนต่อไปก็นำข้อมูลจุดเหล่านี้มาทำการสร้างพื้นผิว โดยเทคนิคที่ศึกษาในงานวิจัยนี้มีด้วยกันอยู่ 2 วิธีดังนี้

3.5.1 การสร้างพื้นผิวสามมิติด้วยจุด

สำหรับเทคนิคแรกที่ใช้ในการสร้างพื้นผิวคือ การสร้างแบบจำลองสามมิติจากข้อมูลกลุ่ม (Surface Reconstruction from Point Cloud) วิธีการคือ เมื่อเรามีเซตของจุดที่กระจายตัวกันอยู่ในพิกัดเชิงฉาก (Cartesian) เสมือนว่ามีวัตถุที่สนใจอยู่แต่ยังไม่มีพื้นผิวในรูปที่ 3.30 มีเฉพาะจุดรูปทรงเหมือนกับวัตถุที่สนใจนั่นเอง ดังนั้นมีวิธีการอย่างไรที่จะสามารถสร้างพื้นผิวเชื่อมจุดเหล่านี้ อาจคิดว่าสามารถสร้างโครงสร้างตาข่ายเชื่อมระหว่างจุดเหล่านั้นแล้วสร้างเป็นพื้นผิวได้เลย แต่ถ้าหากว่าข้อมูลมีปริมาณน้อยหรือจุดกระจายตัวอยู่ห่างกัน ลักษณะของพื้นผิวที่ได้จะเป็นพื้นผิวหยาบ ๆ เพราะฉะนั้นวิธีการจะคล้าย ๆ กับแนวคิดนี้ โดยเริ่มจากการนำจุดที่ได้เหล่านี้มาทำการหาบริเวณเล็กที่มีจุดอยู่ใกล้เคียงจำนวนหนึ่ง (Neighborhood) ในรูปที่ 3.31 ให้เสมือนกับว่าบริเวณนี้คือหนึ่งพื้นผิว เพราะฉะนั้นถ้านำพื้นผิวเล็กๆ เหล่านี้มาต่อกันจำนวนมากจะได้แบบจำลองสามมิตินั่นเอง



รูปที่ 3.30 คอนทัวร์ของภาพ

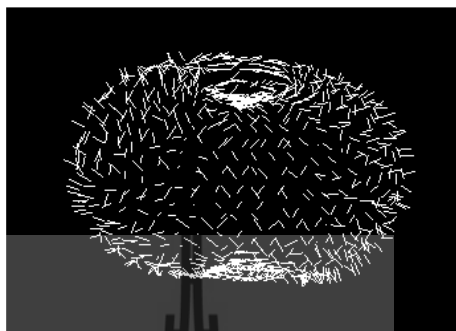


รูปที่ 3.31 Neighborhood

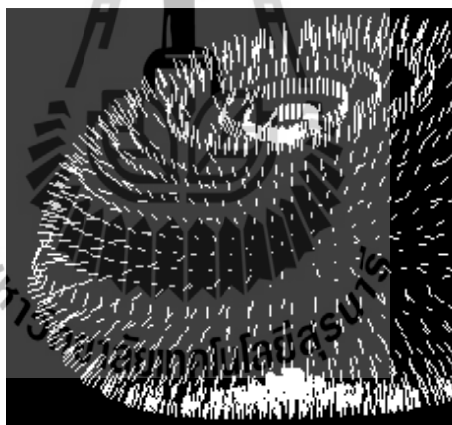
จากนั้นเมื่อหาสมาชิกแต่ละบริเวณเรียบร้อยแล้วจะทำการหาเวกเตอร์ตั้งฉาก ซึ่งจะเป็นการบอกทิศทางให้เราทราบว่าพื้นผิวของระนาบอยู่ในทิศทางใด การหาระนาบในงานวิจัยจะเทคนิคอยู่ 2 วิธี วิธีแรกคือ PCA จะเป็นการลดมิติของข้อมูลลง พุดง่าย ๆ คือ การตัดข้อมูลส่วนที่ไม่จำเป็นออกไป หรือก็คือต้องหารให้ผ่านจุดมากที่สุด โดยหาจากวิธีการทางสถิติคือ ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation or SD) และสูตรหาความแปรปรวน (Covariance) เป็นสิ่งที่บ่งบอกว่ายิ่งข้อมูลกระจายตัวออกจากค่าเฉลี่ยมากเท่าใดส่วนเบี่ยงเบนจะยิ่งเยอะ ดังนั้นเรามีจุดที่อยู่ในพิกัด (x, y, z) จึงเขียนเป็นเมตริกซ์เพื่อหาความแปรปรวนสมการ 2-12

ส่วนอีกวิธีคือ วิธีกำลังสองน้อยที่สุดของระนาบสามมิติ (Planar Fitting 3D: Least square plan) หลักการจะคล้าย ๆ กันคือ จะเป็นหารลดจำนวนมิติของข้อมูล จะหาเส้นหรือระนาบที่ผ่านจุดให้มากที่สุด นิยามได้ด้วยเมตริกซ์ดังสมการ 2-18

เมื่อผ่านขั้นตอนดังกล่าวจะได้ผลลัพธ์ดังรูปที่ 3.32 และ 3.33 ทำให้ทราบพื้นผิวที่จะสร้างมีลักษณะหน้าตาเป็นเช่นไร เพราะโดยปกติแล้วพื้นผิวที่จะถูกสร้างขึ้นจะตั้งฉากกับเวกเตอร์ตั้งฉากที่หาได้นั่นเอง



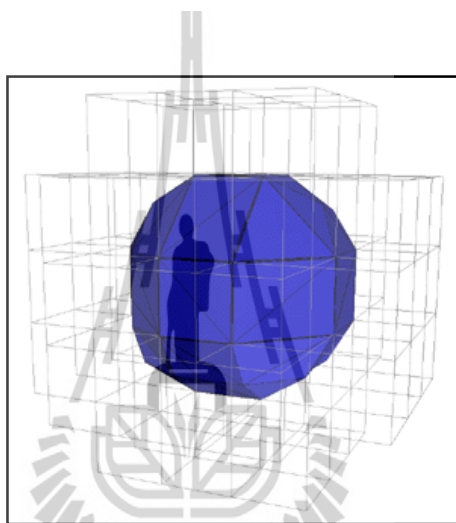
รูปที่ 3.32 เวกเตอร์ตั้งฉาก PCA



รูปที่ 3.33 เวกเตอร์ตั้งฉาก Least Square

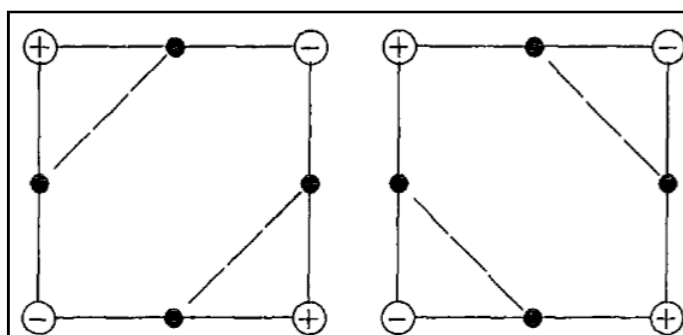
จะสังเกตว่าจากรูปที่ 3.30 และ 3.31 เวกเตอร์จะชี้ไปในทิศทางคนละทิศคนละทาง ดังนั้นจึงจำเป็นต้องกำหนดทิศทางให้เวกเตอร์พุ่งเข้าหา หรือจะให้พุ่งออกจากวัตถุทั้งหมดเป็นไปในทิศทางเดียวกัน เพื่อการสร้างแบบจำลองที่ถูกต้อง ดังนั้นการที่จะกลับเวกเตอร์ทั้งหมดให้เป็นไปในทิศทางเดียวกัน จำเป็นจะต้องหาเส้นทางโดยที่ใช้พลังงานในการกลับเวกเตอร์ให้น้อยที่สุด พูดย่าง ๆ ก็คือเส้นทางที่ทำการกลับเวกเตอร์น้อยตัวที่สุดนั่นเอง งานวิจัยนี้เลือกใช้อัลกอริทึมของพริม (Prim's algorithm) ในการหาเส้นทาง ด้วยการเปลี่ยนค่าน้ำหนักของเส้นเชื่อมแต่ละจุดที่เชื่อมกันจากระยะทางเป็นพลังงานในการกลับเวกเตอร์ เมื่อได้เส้นทางแล้ว ก็ทำการกลับเวกเตอร์ทั้งหมดตาม

เส้นทางนั้น ขึ้นต่อไปจะเป็นการนำเวกเตอร์เหล่านี้มาสร้างพื้นผิวด้วยวิธีการต่อกันของลูกบาศก์ (Marching Cubes Algorithm) ซึ่งมีลักษณะดังรูปที่ 3.34 ลูกบาศก์หนึ่งลูกนั้นหาได้จากจากสไลด์ที่อยู่ติดกัน 2 สไลด์ของภาพต้นแบบดังรูปที่ 2.23 ขั้นตอนนี้เริ่มจากการนำเวกเตอร์ที่มีอยู่มาหาระยะทางจากจุดกึ่งกลางของระนาบ (Signed Distance Function) ไปยังทุกมุมของลูกบาศก์ค่าที่ได้ อาจเป็นได้ทั้งลบและบวก ซึ่งจะเป็นตัวบ่งมีรูปทรงอยู่ที่ลูกบาศก์นี้หรือไม่ ถ้าระหว่างมุมมีมุมใดมุมหนึ่งเป็นบวกและมุมใกล้เคียงกันเป็นลบแสดงว่ามีพื้นผิวอยู่บริเวณนั้นดังรูปที่ 3.35 ซึ่งจุดตัดตรงนี้เองจะมีค่าเป็น 0 คือจุดที่พื้นผิวตัดผ่านลูกบาศก์ โดยเมื่อได้ข้อมูลทั้งหมดแล้วจะทำการเข้ารหัสแล้วเช็คคูตารางว่าลักษณะพื้นผิวที่ได้ตรงกับรูปแบบใดดังรูปที่ 2.14



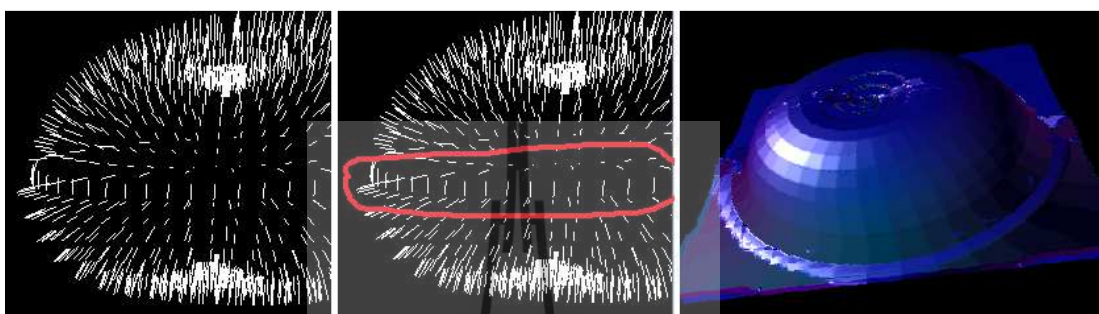
รูปที่ 3.34 การสร้างพื้นผิวด้วยลูกบาศก์

ที่มา : <http://paulbourke.net/geometry/polygonise/> [7 May 2012]



รูปที่ 3.35 หาจุดตัดของพื้นผิวบนลูกบาศก์

เมื่อเสร็จสิ้นกระบวนการดังกล่าวจะได้แบบจำลองสามมิติออกมาดังรูปที่ 3.36 แต่ในส่วนนี้ ได้เกิดปัญหาขึ้นเนื่องจากเวกเตอร์ตั้งฉากที่ได้มามีความผิดพลาด ซึ่งอาจเกิดจากลักษณะจำเพาะของ รูปต้นแบบ หรืออาจเกิดความผิดพลาดขณะเขียน โปรแกรม ดังนั้นจึงได้ศึกษาและพัฒนาเทคนิคการ สร้างพื้นผิวด้วยการขยายทรงกลมด้วยค่าจริงของฟังก์ชันทรงกลมแบบฮาร์โมนิกซึ่งเป็นอีกหนึ่งวิธีที่ มีการใช้กันอยู่

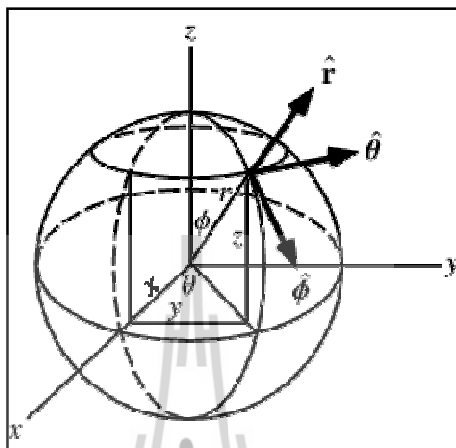


รูปที่ 3.36 การสร้างพื้นผิวด้วยลูกบาศก์

3.5.2 การขยายพื้นผิวด้วยค่าจริงของฟังก์ชันทรงกลมแบบฮาร์โมนิก

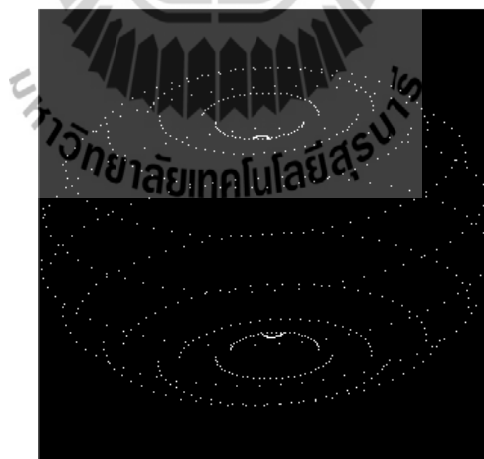
ในการทำงานจริงนั้นบางครั้งอาจจะต้องพบกับปัญหาต่าง ๆ ซึ่งอาจจะเกิดจากความ ไม่ได้ตั้งใจ ความผิดพลาดจากผู้ใช้งาน ความผิดพลาดของระบบ หรือบางทีอาจจะเกิดจากลักษณะ จำเพาะของตัวข้อมูลเองที่ไม่สามารถใช้ได้กับอัลกอริทึมที่เลือกมา สำหรับวิธีที่เลือกใช้วิธีแรกคือ Surface reconstruction from point cloud นั้นข้อดีคือ รูปที่ได้มามีลักษณะสมจริง ภาพมีความ ละเอียด แต่ข้อเสียคือ ใช้เวลาในการประมวลผลนานมาก กระบวนการที่ใช้ในการสร้างแบบจำลอง สามมิตินั้นมีค่อนข้างมาก อาจเกิดข้อผิดพลาดได้ในขั้นตอนใดขั้นตอนหนึ่ง ซึ่งในงานวิจัยชิ้นนี้ เองเกิดในส่วนของการหาเวกเตอร์ตั้งฉาก ดังนั้นจึงได้หันมาศึกษาเทคนิคการขยายทรงกลมด้วยค่า จริงของฟังก์ชันทรงกลมแบบฮาร์โมนิก (Spherical harmonic) แตน (Martin Styner, Ipek Oguz, Shun Xu, Christian Brechbuhler, Dimitrios Pantazis, Guido Gerig, 2006) เพื่อช่วย ในการ สร้าง พื้นผิวของแบบจำลองสามมิติ ซึ่งสามารถสร้างแบบจำลองและใช้เวลาประมวลผลที่เร็วกว่าวิธีแรก อีกทั้งยังสามารถสร้างพื้นผิวให้กับคอนทัวร์ไม่ใช่ภาพปิด หรือภาพที่ไม่สมบูรณ์นั่นเอง รวมไปถึง ขั้นตอนการสร้างแบบจำลองก็มีน้อยกว่าช่วยลดข้อผิดพลาดที่อาจจะเกิดขึ้นได้ ส่วนข้อเสียคือในการ สร้างแบบจำลองต้องมีการหาค่าสัมประสิทธิ์ซึ่งต้องคำนวณหาจากทุกจุดของคอนทัวร์ และอาจใช้ เวลาหากเลือกใช้ลำดับสูง รวมไปถึงมีหลายวิธีที่ใช้ในการคำนวณหาค่าด้วย

โดยปกติข้อมูลตั้งต้นที่ได้มาจะเป็นคอนทัวร์ในแต่ละสไลด์ของภาพสองมิติ เมื่อนำมารวมกันกับทุก ๆ ภาพในแนวลึกทำให้คอนทัวร์ที่ได้มาคือจุดแบบสามมิติดังรูปที่ 3.38 อยู่ในรูปของ (x, y, z) จากนั้นนำจุดที่ได้เหล่านี้มาทำการแปลงให้อยู่ในพิกัดของทรงกลม



รูปที่ 3.37 การแปลงพิกัดทรงกลมและพิกัดฉาก

ที่มา: <http://mathworld.wolfram.com/SphericalCoordinates.html> [24 October 2012]



รูปที่ 3.38 คอนทัวร์ 3 มิติ

จากสมการที่ 2-21 เราจำเป็นต้องแปลงพิกัดฉากให้อยู่ในรูปของพิกัดทรงกลม หาได้จากสมการที่ 3-1 ถึง 3-3 เพื่อนำไปใช้ในการหาค่าสัมประสิทธิ์ c_l^m จากสมการที่ 2-22 นั่นเอง

$$r = \sqrt{x^2 + y^2 + z^2} \quad (3-1)$$

$$\theta = \tan^{-1}\left(\frac{y}{x}\right) \quad (3-2)$$

$$\phi = \cos^{-1}\left(\frac{z}{r}\right) \quad (3-3)$$

และสามารถแปลงกลับได้ด้วยสมการ

$$x = r \sin(\phi) \cos \theta \quad (3-4)$$

$$y = r \sin(\phi) \sin \theta \quad (3-5)$$

$$z = r \cos(\phi) \quad (3-6)$$

สำหรับสมการที่ 2-23 ปรับ $(\cos \theta) e^{im\phi}$ ได้เป็นฟังก์ชันค่าจริงทรงกลมแบบฮาร์โมนิก (Real-Valued Spherical Harmonic Function)

$$Y_l^m(\theta, \phi) = \sqrt{\frac{2l+1}{4\pi} \frac{(l-|m|)!}{(l+|m|)!}} P_l^{|m|}(\theta) \Phi(\phi) \quad (3-7)$$

โดยมีเงื่อนไขดังนี้ (Functions to evaluate a spherical harmonic basis function at a given point, <http://www.paulsprojects.net>)

$$\Phi(\phi) = \begin{cases} \sqrt{2} \cos m\phi, & m < 0 \\ 1 & m = 0 \\ \sqrt{2} \sin |m|\phi, & m > 0 \end{cases} \quad (3-8)$$

เนื่องจากฟังก์ชันฟรี $\Phi(\phi)$ เป็น basis ของ spherical harmonic ตามนิยามของ basis เขียนอยู่ในรูปดังกล่าว 3-8 และรากที่ 2 ของ 2 ที่นำมาคูณนั้นเป็นตัว scale เพื่อให้ basis vector ที่ใช้มีขนาดเท่ากับ 1 หรือเรียกว่า orthonormal basis และฟังก์ชันโพลิโนเมียล P จำแนกได้ดังนี้

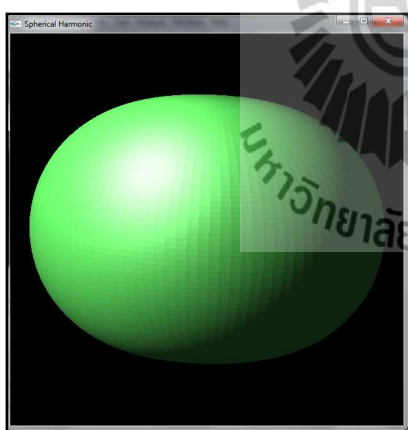
$$P_0^0(\cos \theta) = 1 \quad (3-9)$$

$$P_m^m(\cos\theta) = (2m-1)\sin\theta P_{m-1}^{m-1}(\cos\theta) \quad (3-10)$$

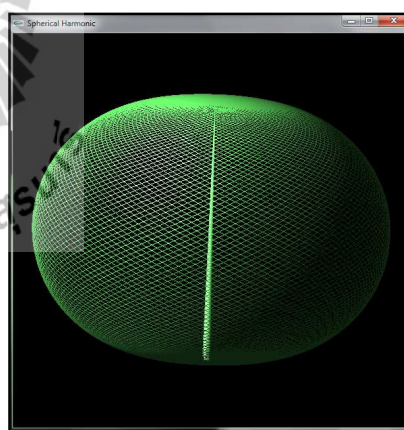
$$P_{m+1}^m(\cos\theta) = (2m+1)\cos\theta P_m^m(\cos\theta) \quad (3-11)$$

$$P_l^m(\cos\theta) = \frac{(2l-1)\cos\theta P_{l-1}^m(\cos\theta) - (l+m-1)P_{l-2}^m(\cos\theta)}{l-m} \quad (3.12)$$

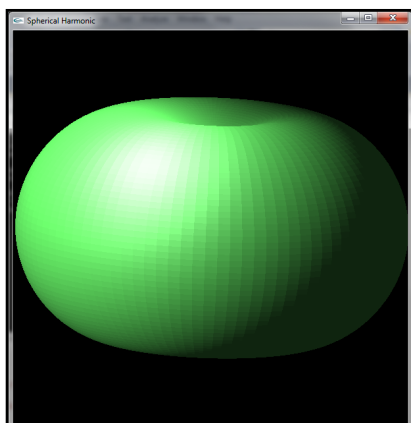
จากสมการที่ 2-22 และ 3-7 ถึง 3-12 การสร้างแบบจำลองสามมิติด้วยฟังก์ชันทรงกลมแบบฮาร์โมนิกนั้นจะสังเกตว่ามีการใช้ลำดับเข้ามาเกี่ยวข้องซึ่งก็คือค่า l, m โดยจะมีการคำนวณหาสัมประสิทธิ์ c_l^m เก็บไว้ตามลำดับตามสมการที่ 2-25 ซึ่งการคำนวณหาค่านั้นจะใช้ข้อมูลต้นแบบหรือคอนทัวร์มาทำการคำนวณ ดังนั้นเมื่อทำการขยายพื้นผิวในแต่ละระดับของอนุกรมฟูเรียร์ แบบจำลองสามมิติที่ถูกสร้างขึ้นจะใกล้เคียงกับวัตถุจริงมากขึ้นตามลำดับ สรุปคือยิ่งลำดับเพิ่มขึ้นแบบจำลองสามมิติที่ถูกสร้างขึ้นจะใกล้เคียงกับวัตถุต้นแบบเพิ่มขึ้นเช่นเดียวกัน และหากต้องการความละเอียดของแบบจำลองเพิ่มขึ้น ให้ทำการเปลี่ยนมุม θ และ ϕ ปรับขึ้นทีละน้อยโดยที่ $\theta \in [0, 2\pi]$ และ $\phi \in [0, \pi]$ ตามลำดับ ซึ่งได้ผลดังรูปที่ 3.39 – 3.41



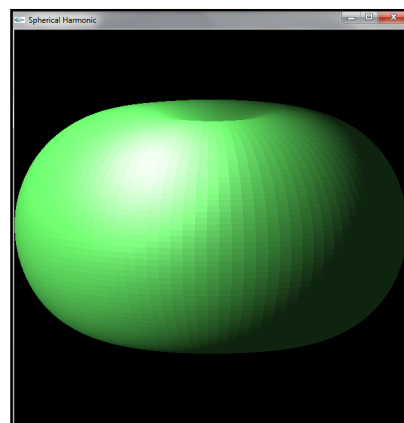
รูปที่ 3.39ก พื้นผิวลำดับที่ 2



รูปที่ 3.39ข โครงตาข่ายลำดับที่ 2



รูปที่ 3.40 พื้นผิวลำดับที่ 5

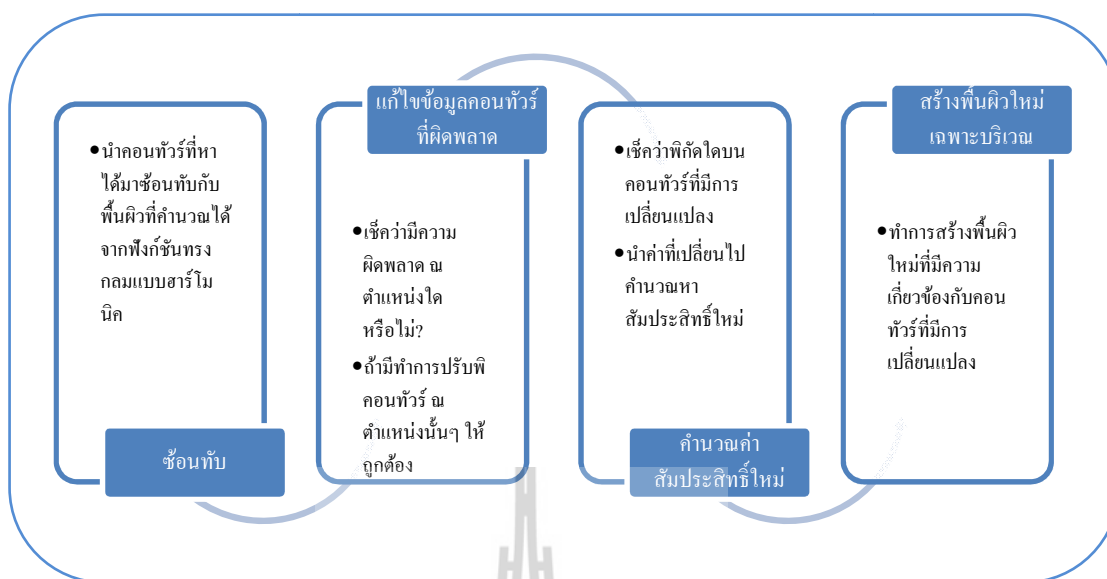


รูปที่ 3.41 พื้นผิวลำดับที่ 10

จากรูปที่ 3.39ก, 3.39ข, 3.40, 3.41 จะสังเกตว่าพื้นผิวเริ่มไม่มีการเปลี่ยนแปลงแล้ว ดังนั้น การเลือกลำดับจึงขึ้นอยู่กับลักษณะของตัวต้นแบบด้วย ซึ่งนี้เป็นเพียงตัวอย่างง่าย ๆ หนึ่ง ตัวอย่าง เพื่อให้ผู้สนใจสามารถพัฒนาต่อไปในอนาคต

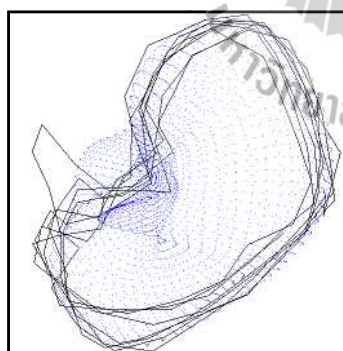
3.6 การปรับพื้นผิวเฉพาะบางจุด (Interactive Surface Editing)

จากขั้นตอนการแยกองค์ประกอบของภาพและการสร้างพื้นผิวด้วยฟังก์ชันทรงกลมแบบฮาร์โมนิก ระบบจะได้ภาพจำลองสามมิติที่มีความใกล้เคียงกับวัตถุจริง แต่บางครั้งหากทำการวิเคราะห์แล้วพบว่าเกิดข้อผิดพลาดขึ้นเพียงเล็กน้อยกับข้อมูล เช่น มีบางจุดหรือบางสไลด์ที่หาคอนทัวร์ออกมาไม่ตรงกับภาพที่ควรจะเป็น ดังนั้นถ้าหากผู้ใช้ทำการสร้างภาพจำลองขึ้นมาใหม่โดยทำการเรนเดอร์ทั้งหมด จะต้องใช้เวลาค่อนข้างมาก ยิ่งสำหรับเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพไม่สูงมากนัก หรือตัวข้อมูลเองมีขนาดใหญ่ จะทำให้การประมวลผลเป็นไปได้อย่างลำบาก ดังนั้นจึงได้นำเสนอแนวทางนี้ขึ้นมาเพื่อช่วยลดปัญหาในจุดนี้ โดยระบบจะนำคอนทัวร์ที่ได้จากการแยกองค์ประกอบของภาพ มาซ้อนลงบนพื้นผิวที่ถูกสร้างขึ้นด้วยฟังก์ชันทรงกลมแบบฮาร์โมนิก จากนั้นให้ผู้ใช้เลือกจุดบนคอนทัวร์ที่เห็นว่ามีข้อผิดพลาด แล้วให้กำหนดจุดนั้นใหม่ โดยให้ไปอยู่ในตำแหน่งที่ถูกต้องเพื่อทำการหาคอนทัวร์บนจุดนั้นใหม่อีกครั้ง หลังจากนั้นระบบจะนำจุดของคอนทัวร์ที่แก้ไขนี้ไปทำการคำนวณหาความสัมพันธ์เพื่อใช้ในการสร้างพื้นผิวแบบฮาร์โมนิก และจะทำการเรนเดอร์ภาพขึ้นมาใหม่ แต่ไม่ได้ทำทั้งหมด จะเลือกทำเพียงบริเวณรอบ ๆ จุดที่มีการเปลี่ยนแปลงเท่านั้น ทำให้สามารถเพิ่มความเร็วได้มากในการประมวลผลต่อไป ซึ่งสามารถเขียนเป็นแผนภาพขั้นตอนการทำงานได้ดังนี้

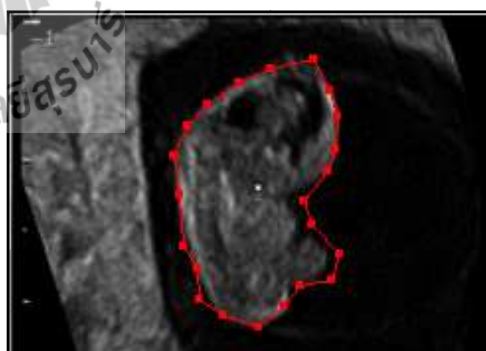


รูปที่ 3.42 แผนภาพการทำงานของขั้นตอนการปรับพื้นผิวบางจุดหรือบางบริเวณ

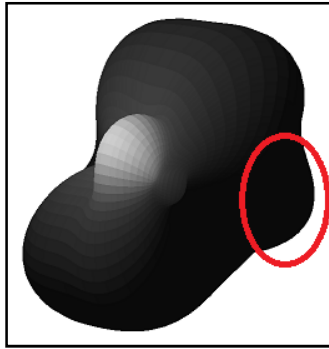
เมื่อนำคอนทัวร์ที่ได้มาซ้อนทับกับพื้นผิวที่ถูกสร้างขึ้นด้วยฟังก์ชันทรงกลมแบบฮาร์โมนิก ดังรูปที่ 3.43 หากพบว่ามีบางจุดที่มีความผิดพลาดดังรูปที่ 3.45 จะทำการปรับแก้ไขที่พิกัดนั้นบนคอนทัวร์ และทำการสร้างแบบจำลองใหม่ที่จุดนั้นตามรูปที่ 3.46



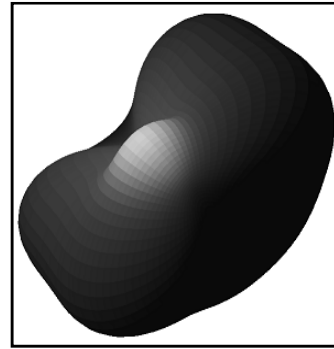
รูปที่ 3.43 โครงตาข่ายรูปทรง



รูปที่ 3.44 แก้ไขคอนทัวร์



รูปที่ 3.45 แบบจำลองสามมิติเดิม



รูปที่ 3.46 แบบจำลองสามมิติที่ถูกสร้างใหม่



บทที่ 4

การทดสอบและผลการทดสอบ

4.1 กล่าวนำ

ในบทนี้จะกล่าวถึงขั้นตอนและวิธีการทดสอบระบบที่ได้ทำการพัฒนาขึ้น โดยใช้เวลาความถูกต้องของข้อมูลต้นแบบเทียบกับข้อมูลของแบบจำลองเป็นตัวชี้วัด แบ่งการทดลองออกเป็น 4 ส่วน ส่วนที่ 1 จะวัดผลด้านเวลาในการประมวลผลการสร้างพื้นผิว ส่วนที่ 2 จะวัดค่าความถูกต้องของข้อมูลต้นแบบเทียบกับข้อมูลของแบบจำลอง ส่วนที่ 3 และ 4 เป็นการวัดความถูกต้องของข้อมูลโดยการทำซ้ำข้อมูลเดิมหลาย ๆ ครั้งแล้วตรวจสอบว่ามีค่าความผิดพลาดเฉลี่ย มากหรือน้อยเพียงใด ซึ่งอธิบายได้ดังนี้

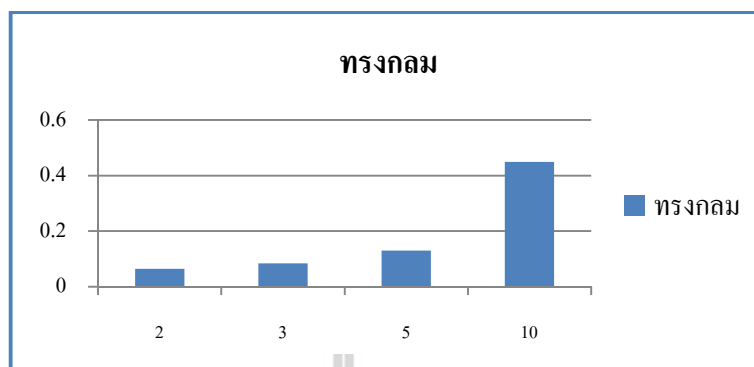
4.2 การวัดเวลาการสร้างแบบจำลองตามลำดับ

การทดลองนี้ใช้ทดสอบเวลาในการประมวลผลการสร้างพื้นผิวของแบบจำลองสามมิติด้วยวิธีการขยายค่าจริงจากฟังก์ชันทรงกลมแบบฮาร์โมนิกเป็นตัวชี้วัด ซึ่งจะทำให้ทราบเวลาในแต่ละลำดับ (Order) ของแต่ละชุดข้อมูลว่า ใช้เวลาในการประมวลผลสร้างพื้นผิวมากน้อยเพียงใด โดยใช้ข้อมูลทดสอบเพียง 1 ชุดของแต่ละต้นแบบ ได้ผลการทดลองดังตารางที่ 4.1

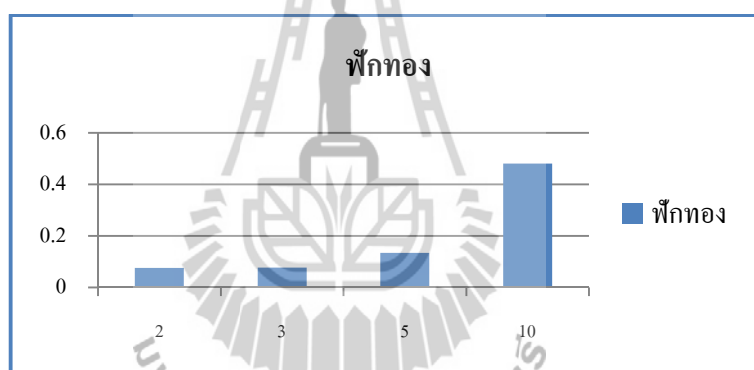
ตารางที่ 4.1 ตารางผลการทดสอบระบบด้วยเวลาประมวลผลการสร้างพื้นผิวแบบจำลอง

รูปทรงวัตถุ	จำนวนข้อมูล	ลำดับ	เวลาประมวลผลสร้างพื้นผิว
ทรงกลม	450 จุด	2	0.065
		3	0.084
		5	0.13
		10	0.449
พีคทอง	720 จุด	2	0.076
		3	0.077
		5	0.134
		10	0.481
ทารก	172 จุด	2	0.072
		3	0.087
		5	0.129
		10	-

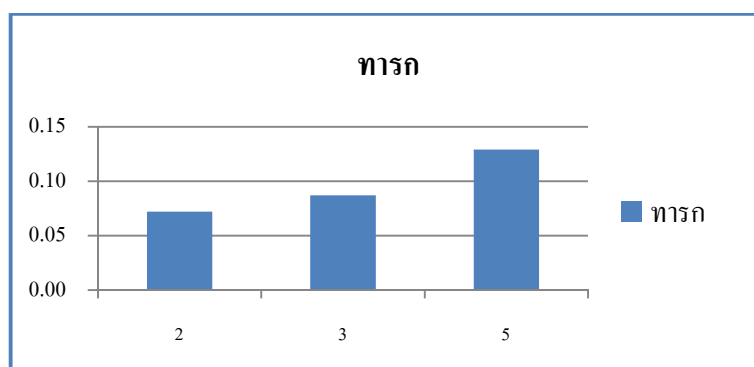
จากข้อมูลตารางที่ 4.1 สามารถนำมาสร้างเป็นกราฟแผนภูมิได้ดังรูปที่ 4.1 ถึง 4.3



รูปที่ 4.1 กราฟเวลาประมวลผลการสร้างพื้นผิวรูปทรงกลม



รูปที่ 4.2 กราฟเวลาประมวลผลการสร้างพื้นผิวรูปพีคทอง



รูปที่ 4.3 กราฟเวลาประมวลผลการสร้างพื้นผิวรูปทารก

จากผลการทดลองสังเกตว่า หากลำดับ (Order) สูงหรือมีค่ามาก เวลาที่ใช้ในการประมวลผลสร้างพื้นผิวแบบจำลองสามมิติก็จะมาก แต่ถ้าหากลำดับต่ำหรือมีค่าน้อยเวลาที่ใช้ในการประมวลผลก็จะน้อยไปด้วย ส่วนผลของการสร้างพื้นผิวจะมีความใกล้เคียงกับวัตถุต้นแบบมากหรือน้อยเพียงใดจะทดลองต่อไปในหัวข้อที่ 4.3 – 4.5 ในส่วนของรูปเด็กทารกจะสังเกตว่าลำดับที่ใช้มีเพียง 2, 3 และ 5 เท่านั้น นั่นเป็นเพราะวัตถุต้นแบบที่ได้มานั้นมีข้อมูลไม่ครบถ้วนดังรูปที่ 3.39 จะเห็นว่าที่หัวและท้ายของคอนทัวร์ไม่มีส่วนที่มาปิด แต่ด้วยวิธีของฟังก์ชันทรงกลมแบบฮาร์โมนิก จึงมีการสร้างพื้นผิวขึ้นมาปิดหัวและท้ายให้โดยอัตโนมัติ พบว่าหากใช้ลำดับมากเกินไปจะทำให้พื้นผิวที่ถูกสร้างขึ้นนั้นมีความผิดพลาด ดังนั้นเพื่อความถูกต้อง และมีประสิทธิภาพ ดังนั้นควรเลือกลำดับให้เหมาะสมกับวัตถุต้นแบบ

4.3 วัดค่าความผิดพลาด (Error) ของแบบจำลองสามมิติ

ในการทดลองนี้จะทำให้ทราบว่าวัตถุต้นแบบ กับแบบจำลองที่ถูกสร้างขึ้นด้วยวิธีการขยายพื้นผิวของฟังก์ชันทรงกลมแบบฮาร์โมนิก (Spherical Harmonic) มีความใกล้เคียงกันมากหรือน้อยเพียงใด ซึ่งจะวัดจากค่าความผิดพลาดที่เกิดขึ้น หากวัตถุต้นแบบมีความใกล้เคียงกับแบบจำลองที่ถูกสร้างขึ้นนั้นแสดงว่าค่าความผิดพลาดต้องมีค่าน้อย ในทางตรงกันข้าม หากแบบจำลองที่ถูกสร้างขึ้นไม่มีความใกล้เคียงกับวัตถุต้นแบบ ค่าความผิดพลาดวัดได้ต้องมีค่ามากด้วยเช่นกัน โดยใช้ผู้ทดสอบเพียง 1 คนทำการทดสอบ เริ่มจากผู้ทดสอบจะทำการหาคอนทัวร์ของภาพ จากนั้นระบบจะสร้างพื้นผิวของแบบจำลองภาพสามมิติขึ้นมา และตรวจสอบว่า ทุก ๆ ตำแหน่งบนคอนทัวร์ว่า เมื่อนำไปหาค่าด้วยวิธีการขยายพื้นผิวของฟังก์ชันทรงกลมแบบฮาร์โมนิก แล้วค่าที่ได้มานั้นมีความใกล้เคียงกับพิกัดต้นแบบหรือคอนทัวร์มากหรือน้อยเพียงใด ที่มุม θ และ ϕ ใด ๆ พิกัด x, y และ z ตรงกันหรือไม่ มีค่าความผิดพลาดมากเพียงใด หาได้จาก

$$error_{xi} = x_{original} - x_{xsh} \quad (4-1)$$

$$error_{yi} = y_{original} - y_{ysh} \quad (4-2)$$

$$error_{zi} = z_{original} - z_{zsh} \quad (4-3)$$

$(x_{original}, y_{original}, z_{original})$ คือพิกัด (X, Y, Z) ที่ได้จากการหาคอนทัวร์ของภาพ เมื่อนำพิกัดจุดเหล่านี้ไปคำนวณด้วยฟังก์ชันทรงกลมแบบฮาร์โมนิกจะได้ $(x_{xsh}, y_{ysh}, z_{zsh})$ เพื่อนำไปใช้ในการสร้างพื้นผิวของแบบจำลอง 3 มิติ และเพื่อหาค่าความผิดพลาดที่เกิดขึ้นจึงนำพิกัดเดิมที่ได้จากการหาคอนทัวร์ และพิกัดที่ได้จากการหาด้วยฟังก์ชันทรงกลมแบบฮาร์โมนิกมาหาค่าความแตกต่างด้วยการลบกัน จากนั้นนำค่าความผิดพลาดของทั้ง 3 แกนยกกำลังสองมารวมกัน แล้วถอดรากเพื่อหาค่าความผิดพลาดรวมที่ตำแหน่ง θ และ ϕ ใด ๆ แล้วหารด้วยจำนวนจุดทั้งหมดของแต่ละรูปทรง จะได้ค่า $error_{order}$ เฉลี่ยในแต่ละรูปทรง เขียนได้ดังสมการ

$$error_{order} = \frac{\sum_{i=0}^{n-1} \sqrt{error_{xi}^2 + error_{yi}^2 + error_{zi}^2}}{n} \quad (4-4)$$

ซึ่งในการที่จะสร้างแบบจำลอง 3 มิตินั้น จำเป็นต้องทำการปรับค่าให้เป็นหนึ่งหน่วยเสียก่อนเนื่องจากฟังก์ชันที่ใช้ในการแสดงผลมีค่าอยู่ระหว่าง -1 ถึง 1 หาได้จากสมการที่ 4-5 ถึง 4-7

$$x' = \left(\frac{x - \min}{\max - \min} \right) * 2 - 1 \quad (4-5)$$

$$y' = \left(\frac{y - \min}{\max - \min} \right) * 2 - 1 \quad (4-6)$$

$$z' = \left(\frac{z - \min}{\max - \min} \right) * 2 - 1 \quad (4-7)$$

ดังนั้นคอนทัวร์ที่ได้จะมีค่าอยู่ระหว่าง -1 ถึง 1 และเมื่อนำไปใช้ในการสร้างพื้นผิวด้วยฟังก์ชันทรงกลมแบบฮาร์โมนิก ค่าที่ได้จากขั้นตอนการสร้างพื้นผิวนี้จะอยู่ในช่วง -1 ถึง 1 เช่นกัน จึงต้องมีการแปลงกลับเพื่อให้ได้หน่วยเป็นพิกเซลเช่นเดิมด้วยสมการที่ 4-8 ถึง 4-10

$$x = \left(\frac{(x' + 1)}{2} * (\max - \min) \right) + \min \quad (4-8)$$

$$y = \left(\frac{(y' + 1)}{2} * (\max - \min) \right) + \min \quad (4-9)$$

$$z = \left(\frac{(z' + 1)}{2} * (\max - \min) \right) + \min \quad (4-10)$$

จากนั้นทำการปรับหน่วยค่าผิดพลาดโดยวัดเป็นเปอร์เซ็นต์ หาได้จากสมการที่ 4-11

$$\%error = \frac{error_{order}}{Distance_{avg}} \times 100 \quad (4-11)$$

ซึ่งระยะทางเฉลี่ยของแต่ละรูปทรงหาได้จากการหาระยะทางเฉลี่ยของคอนทัวร์โดยวัดจากจุดศูนย์กลางของคอนทัวร์ถึงแต่ละจุดบนคอนทัวร์ดังสมการที่ 4-12

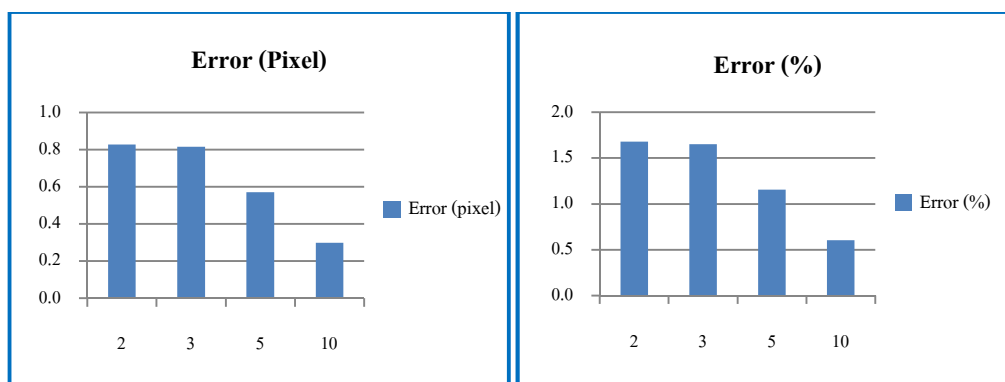
$$Distance_{avg} = \frac{\sum_{i=0}^{n-1} \sqrt{x^2+y^2+z^2}}{n} \quad (4-12)$$

ด้วยวิธีต่าง ๆ ค่าความผิดพลาดแต่ละลำดับเฉลี่ยแล้วได้ผลทดลองดังข้อมูลในตารางที่ 4.2 ดังนี้

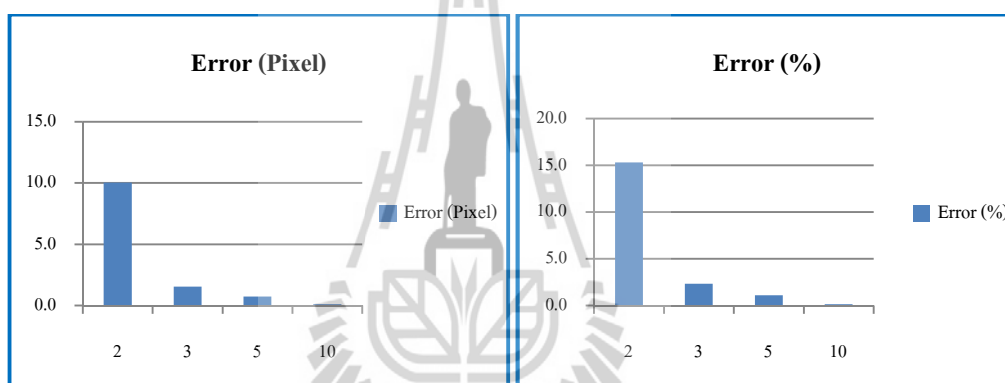
ตารางที่ 4.2 ตารางวัดค่าความผิดพลาดของการสร้างแบบจำลอง 3 มิติ

รูปทรงวัตถุ	จำนวนข้อมูล	ระยะทางเฉลี่ย แต่ละรูปทรง	ลำดับ	ค่าความผิดพลาด (Pixel)	ค่าความผิดพลาด (%)
ทรงกลม	420 จุด	49.35289	2	0.828011	1.677736
			3	0.815095	1.651565
			5	0.570121	1.155192
			10	0.298232	0.604285
พีคทอง	660 จุด	65.59584	2	10.04605	15.31508
			3	1.532411	2.33614
			5	0.718813	1.09582
			10	0.100121	0.152633
ทารก	175 จุด	54.101545	2	9.548295383	17.6488405
			3	6.197688546	11.45565907
			5	4.924463086	9.102259624
			10	-	-

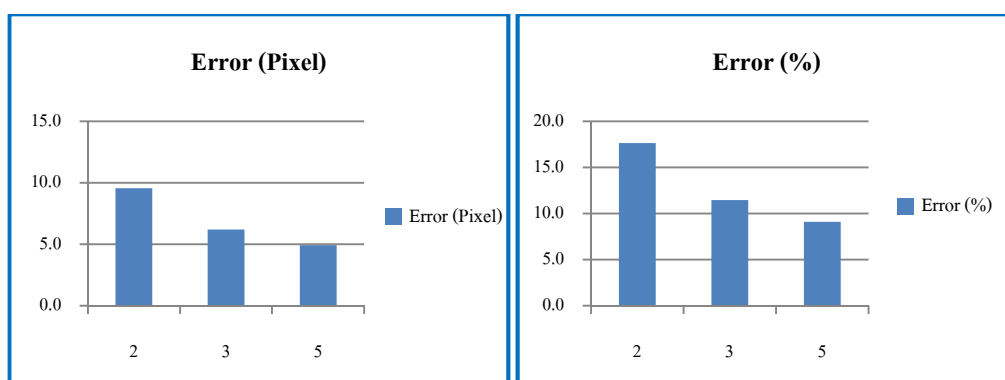
จากข้อมูลตารางที่ 4.2 สามารถนำไปสร้างเป็นกราฟแผนภูมิได้ดังรูปที่ 4.4 ถึง 4.6



รูปที่ 4.4 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติทรงกลม



รูปที่ 4.5 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติรูปฟักทอง



รูปที่ 4.6 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติรูปทหาร

จากผลการทดลองเห็นได้ว่ายิ่งลำดับ (Order) สูงเท่าไร ค่าความผิดพลาดของจุดต้นแบบกับจุดของแบบจำลอง 3 มิติที่ถูกสร้างขึ้นจะมีค่าน้อยลงเช่นเดียวกัน แสดงให้เห็นว่ายิ่งลำดับสูงเท่าใดแบบจำลองที่ถูกสร้างขึ้นก็จะความใกล้เคียงกับวัตถุต้นมากขึ้นเท่านั้น ดังนั้นเพื่อความถูกต้อง และมีประสิทธิภาพ ควรเลือกลำดับให้เหมาะสมกับวัตถุต้นแบบ

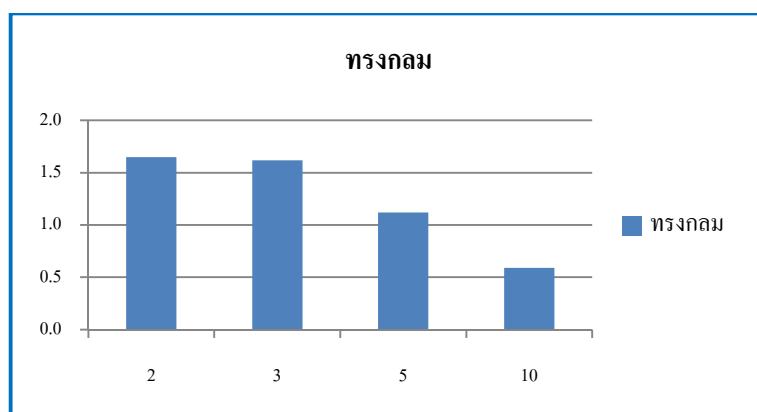
4.4 ทดสอบค่าความผิดพลาดจากผู้ทดสอบ 1 คนทำการทดสอบหลายครั้ง (Intra Observer Variability)

จะเป็นการตรวจสอบค่าความผิดพลาดของแบบจำลองสามมิติที่เกิดจาก ผู้ทดสอบ 1 คนทำการทดสอบการสร้างแบบจำลองสามมิติขึ้นมา 4 ครั้ง โดยการเริ่มจากการหาคอนทัวร์ของภาพทำคล้ายขั้นตอนที่ 4.3 แต่จะคำนวณทั้งหมด 4 ครั้งโดยหาค่าเฉลี่ยรวมในแต่ละลำดับ เพื่อเปรียบเทียบค่าความผิดพลาดว่าออกมาในลักษณะใด โดยสรุปเป็นตารางที่ 4.3 ได้ดังนี้

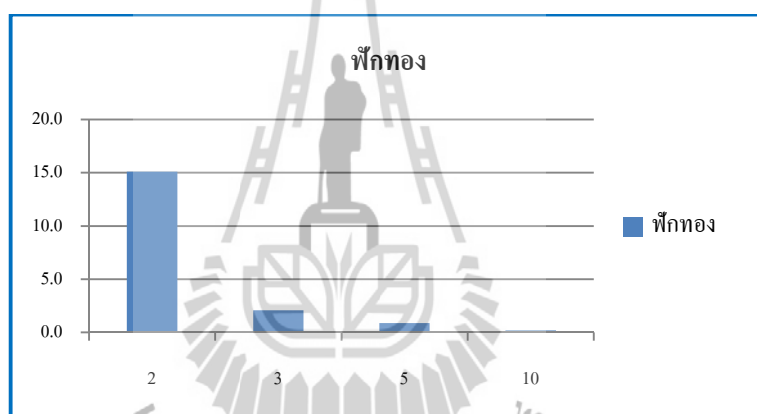
ตารางที่ 4.3 ตารางผลการทดสอบผู้ใช้ 1 คนทำการทดลอง 4 ครั้ง

รูปทรงวัตถุ	ลำดับ	Error1 (%)	Error2 (%)	Error3 (%)	Error4 (%)	Error เฉลี่ย (%)
ทรงกลม	2	1.689104	1.677736	1.502886	1.72346	1.6482965
	3	1.656685	1.651565	1.46936	1.695004	1.6181535
	5	1.161212	1.155192	0.978929	1.187432	1.1206913
	10	0.616332	0.604285	0.506686	0.63916	0.5916158
ฟักทอง	2	15.02199	14.89046	15.31508	15.1427	15.092558
	3	1.991845	1.835053	2.33614	2.18372	2.0866895
	5	0.706598	0.577153	1.09582	1.08134	0.8652278
	10	0.18262	0.21096	0.152633	0.137221	0.1708585
ทารก	2	16.52334395	17.82908368	17.44493165	17.6488405	17.36155
	3	10.43020598	11.06574759	11.08798377	11.45565907	11.009899
	5	7.853006647	8.190991189	8.147735543	9.102259624	8.3234983
	10	-	-	-	-	-

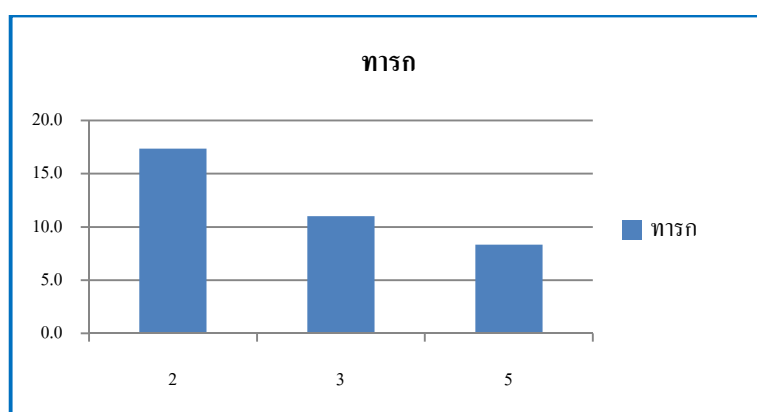
จากตารางที่ 4.3 จะเห็นได้ว่าการสร้างแบบจำลองสามมิติของรูปทารกนั้น ในลำดับที่ 10 ไม่มีการคำนวณหาความผิดพลาด เนื่องจากภาพต้นแบบมีลักษณะคือส่วนหัวและท้ายจะโป่ง ไม่มีคอนทัวร์บริเวณนั้นเลย ดังนั้นหากลำดับสูงขึ้น ด้วยการขยายตัวของฟังก์ชันแบบฮาร์โมนิกจะสร้างพื้นผิวขึ้นมาเพื่อแบบจำลองสามมิติสมบูรณ์และเกิดข้อผิดพลาดขึ้นนั่นเอง จากตารางที่ 4.3 สามารถเขียนเป็นกราฟได้ดังนี้



รูปที่ 4.7 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติทรงกลม



รูปที่ 4.8 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติรูปพีักทอง



รูปที่ 4.9 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติรูปทารก

จากการทดสอบพบว่า ผู้ทดสอบ 1 คนที่ทำการทดสอบในแต่ละครั้งอาจมีค่าความผิดพลาดเกิดขึ้นได้ ซึ่งในแต่ละครั้งอาจไม่เท่ากัน จากตารางและแผนภูมิจะเห็นได้ว่าผู้ทดสอบอาจทำการหาคอนทัวร์ออกมาได้ต่างกันในแต่ละครั้ง หรืออาจมีความผิดพลาดที่เกิดขึ้นในคอนทัวร์ ส่งผลให้ได้ค่าความผิดพลาดไม่เท่ากันในแต่ละรอบ แต่ก็ไม่ได้ส่งผลต่อการสร้างพื้นผิวในลำดับต่าง ๆ เพราะค่าความผิดพลาดที่วัดได้จากการทดสอบก็ยังคงมีค่าน้อยลงเรื่อย ๆ ตามลำดับที่สูงขึ้น สรุปได้ว่าหากใช้ลำดับสูงขึ้นภาพจำลองสามมิติที่ถูกสร้างขึ้นจะยิ่งใกล้เคียงกับวัตถุต้นแบบ

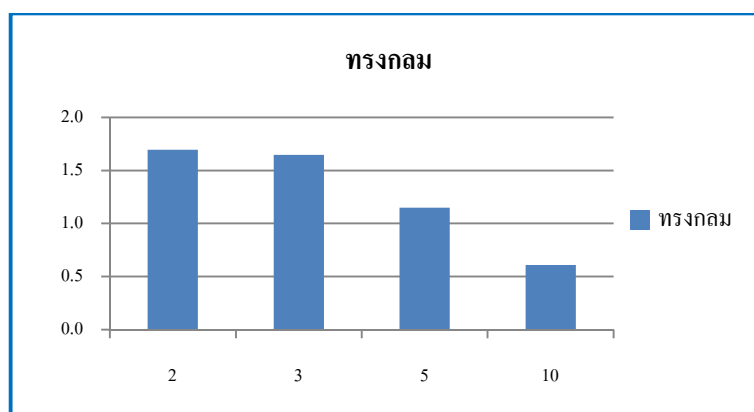
4.5 ทดสอบค่าความผิดพลาดจากผู้ทดสอบหลายคน ทำการทดสอบคนละ 1 ครั้ง (Inter Observer Variability)

การทดสอบนี้จะคล้ายกับการทดสอบที่ 4.4 แตกต่างตรงที่จะใช้ผู้ทดสอบ 4 คน ทำการทดสอบการสร้างแบบจำลองสามมิติขึ้นมามีคนละ 1 ครั้ง ซึ่งผลการทดสอบได้ตามตารางที่ 4.4

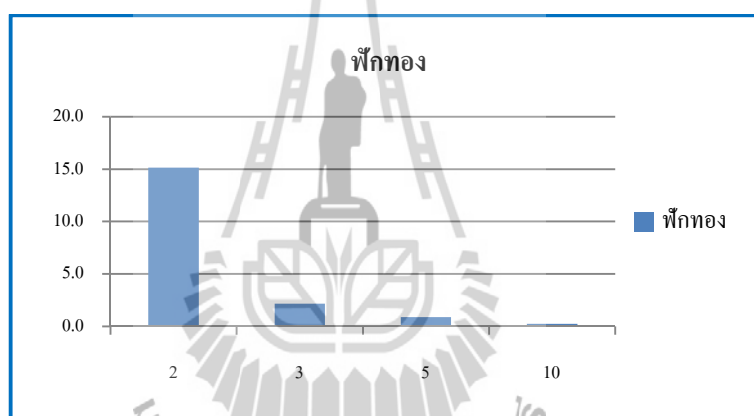
ตารางที่ 4.4 ตารางผลการทดสอบจากผู้ใช้ 4 คน ทำการทดลองคนละ 1 ครั้ง

รูปทรงวัตถุ	ลำดับ	Error1 (%)	Error2 (%)	Error3 (%)	Error4 (%)	Error เฉลี่ย (%)
ทรงกลม	2	1.712536	1.685527	1.666537	1.715763	1.6950908
	3	1.678431	1.619889	1.631225	1.653968	1.6458783
	5	1.143761	1.131756	1.164211	1.156749	1.1491193
	10	0.600836	0.617583	0.590616	0.633562	0.6106493
พีคทอง	2	15.02199	14.89046	15.37866	15.31508	15.151548
	3	1.991845	1.835053	2.523807	2.33614	2.1717113
	5	0.706598	0.577153	1.106225	1.09582	0.871449
	10	0.18262	0.21096	0.39669	0.152633	0.2357258
ทารก	2	16.71039253	17.11463577	17.68332675	17.69965868	17.302003
	3	10.74449434	10.25064081	11.77549937	11.35397009	11.031151
	5	7.69873707	7.735628026	9.069388555	8.362473573	8.2165568
	10	-	-	-	-	-

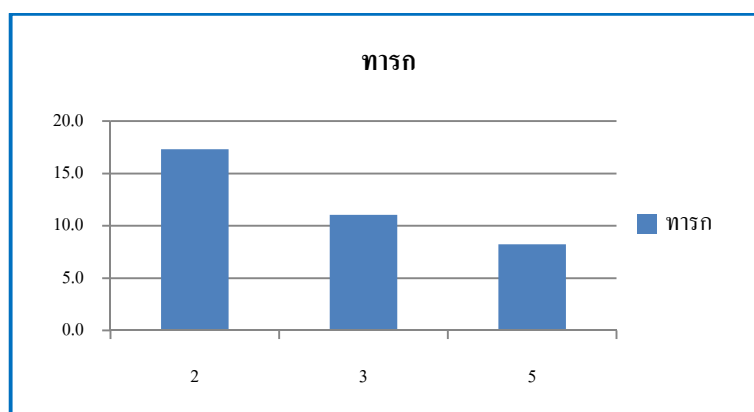
จากการทดลองดังกล่าวในตารางที่ 4.4 สามารถสร้างเป็นกราฟได้ดังนี้



รูปที่ 4.10 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติทรงกลม



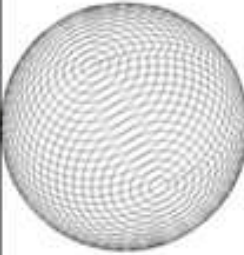
รูปที่ 4.11 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติรูปพีคทอง



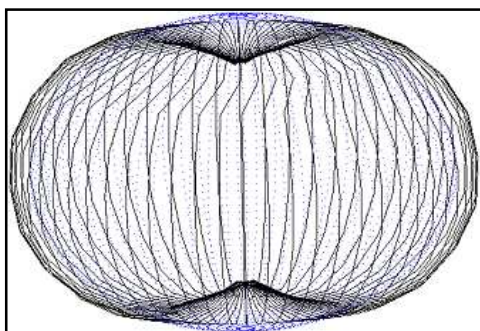
รูปที่ 4.12 กราฟวัดความผิดพลาดของการสร้างแบบจำลองสามมิติรูปทารก

จากกราฟรูปที่ 4.10, 4.11 และ 4.12 พบว่า เมื่อเปลี่ยนให้ใช้ผู้ทดสอบเป็น 4 คนทำการทดสอบ แนวโน้มของการทดลองก็ยังคงเดิมคือ ค่าความผิดพลาดจะลดลงเรื่อย ๆ ตามลำดับที่สูงขึ้น แสดงให้เห็นว่าถึงแม้จะมีค่าความผิดพลาดที่ต่างกันของแต่ละผู้ทดสอบ แต่เมื่อหาค่าเฉลี่ยแต่ละลำดับออกมาผลก็ยังคงมีแนวโน้มไม่เปลี่ยนแปลงไป ซึ่งจากทุกการทดลองให้ผลทดสอบดังรูปที่ 4.13 โดยจะเป็นรูปผลของการสร้างพื้นผิวแบบจำลองสามมิติด้วยฟังก์ชันทรงกลมแบบฮาร์โมนิกในลำดับต่าง ๆ ส่วนรูปที่ 4.14 ถึง 4.17 เป็นการนำวัตถุต้นแบบและพื้นผิวที่ถูกสร้างขึ้นมาซ้อนทับกัน ทำให้เห็นชัดว่ายิ่งลำดับสูงเท่าใด ภาพจำลองที่ถูกสร้างขึ้นก็มีความกับวัตถุต้นแบบมากขึ้นเท่านั้น และด้วยวิธีของฟังก์ชันทรงกลมแบบฮาร์โมนิกถึงแม้ว่าคอนทัวร์ที่หาจะไม่สมบูรณ์ อาจมีบางส่วนขาดหายไป โดยสังเกตได้จากรูปที่ 4.16 และ 4.17 ส่วนหัวและท้ายของคอนทัวร์มีการขาดหายของข้อมูล ทำให้เป็นคอนทัวร์ที่มีลักษณะไม่สมบูรณ์ คือไม่ใช่รูปทรงปิด แต่ด้วยวิธีของฟังก์ชันทรงกลมแบบฮาร์โมนิกก็ยังคงสร้างพื้นผิวบริเวณนั้นขึ้นมาได้ ดังนั้นเพื่อความถูกต้อง และมีประสิทธิภาพ ควรเลือกลำดับให้เหมาะสมกับวัตถุต้นแบบ

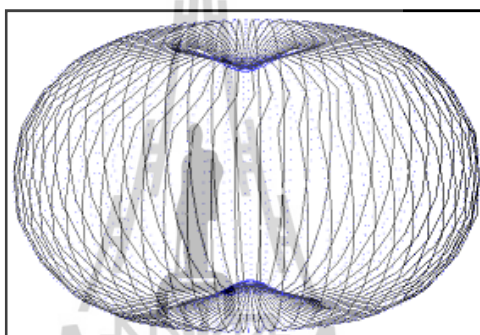


รูปทรงต้นแบบ				การแสดงผลของรูปทรงตัวต่างกัน		
ทรงกลม				2	3	5
						
พีกทอง				2	3	5
						
หารก				2	3	5
						

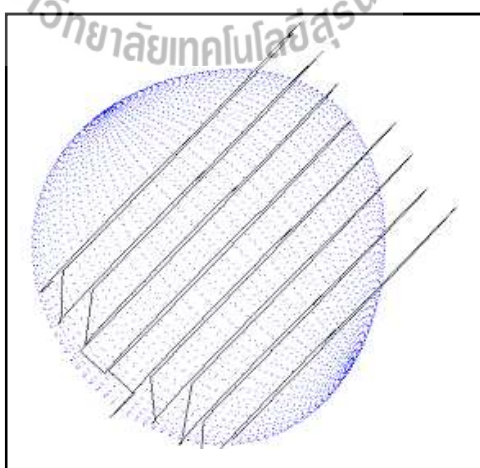
รูปที่ 4.13 ผลการสร้างแบบจำลองสามมิติของรูปทรงต่าง ๆ



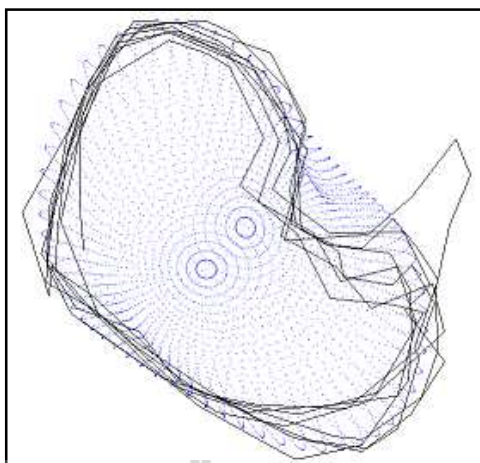
รูปที่ 4.14 การซ้อนทับของแบบจำลองลำดับที่ 2



รูปที่ 4.15 การซ้อนทับของแบบจำลองลำดับที่ 5



รูปที่ 4.16 การซ้อนทับของแบบจำลองลำดับที่ 2



รูปที่ 4.17 การซ้อนทับของแบบจำลองลำดับที่ 3



บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

ปัจจุบันคอมพิวเตอร์เป็นเครื่องมือสำคัญที่หลากหลายสาขาอาชีพนิยมนำไปใช้เพื่อเพิ่มความสะดวกรวดเร็วให้กับงานที่เกี่ยวข้อง การเก็บข้อมูลก็สามารถเก็บข้อมูลได้จำนวนมากและมีระบบความปลอดภัยในการเข้าถึงข้อมูล ป้องกันการเข้าถึงจากบุคคลจากภายนอกที่ไม่เกี่ยวข้อง รวมไปถึงการสืบค้นข้อมูลจำนวนมากได้อย่างรวดเร็วและมีประสิทธิภาพ ในทางการแพทย์เอง ก็เช่นกัน มีการนำระบบคอมพิวเตอร์มาประยุกต์ใช้อย่างมากมายเช่น ใช้ในการเก็บข้อมูลของผู้ป่วย ที่มีจำนวนมากขึ้นทุกวัน การค้นหาประวัติผู้ป่วยก็ค้นหาได้อย่างรวดเร็ว การทำระบบจ่ายยาทำให้เพิ่มความถูกต้องแม่นยำในการจ่ายยา เป็นต้น ดังนั้นเมื่อมีการนำคอมพิวเตอร์มาประยุกต์ใช้ในทางการแพทย์อย่างมากมาย ประกอบกับเครื่องมือทางการแพทย์ที่ทันสมัยก็มีราคาแพงมาก ส่วนสิ่งที่มีคือ ข้อมูลภาพสองมิติที่มีคุณภาพต่ำไม่ว่าจะเป็นภาพ X-Ray หรือภาพ Ultrasound ดังนั้นจึงเกิดความคิดที่จะสามารถนำสิ่งเหล่านี้มารวมกันได้อย่างไร เพื่อลดต้นทุนการสั่งซื้ออุปกรณ์ราคาแพงเหล่านี้ โดยนำภาพสองมิติที่มีอยู่แล้วมาใช้งาน ดังนั้นจุดประสงค์ของงานวิจัย คือการนำเอาภาพ Ultrasound สองมิติมาสร้างเป็นภาพจำลองแบบสามมิติที่คล้ายกับเครื่องมือราคาแพงเหล่านั้น ทำให้แพทย์สามารถวิเคราะห์ความผิดปกติของทารกในครรภ์ได้จากภาพสามมิติที่สร้างขึ้นมาโดยไม่ต้องสั่งซื้อเครื่องมือ และเป็นการเพิ่มความถูกต้อง แม่นยำในการวิเคราะห์ได้อีกด้วย

5.1 สรุปขั้นตอนการทำงานของระบบสร้างแบบจำลองสามมิติ

สำหรับระบบจะแบ่งการทำงานออกเป็น 3 ส่วน โดยส่วนแรกคือการแยกองค์ประกอบของภาพ จะเริ่มจากการที่ผู้ใช้งานจะต้องกำหนดตำแหน่งหรือจุดบนภาพ รอบขอบเขตของวัตถุที่สนใจ ภายในภาพ หรือตัวทารก โดยจุดเหล่านั้นจะต้องเรียงกันเป็นลำดับ ล้อมรอบวัตถุที่สนใจ ซึ่งจะเสร็จสิ้นก็ต่อเมื่อผู้ใช้งานกำหนดจุดแรกและจุดสุดท้ายเป็นจุดเดียวกัน จากนั้นระบบจะดำเนินการต่อ ลักษณะดังกล่าวเรียกว่าระบบกึ่งอัตโนมัติ (Semi-Automatic) ซึ่งขั้นตอนนี้จะมีความสำคัญมาก เพราะการจะได้แบบจำลองที่ถูกต้องนั้น จำเป็นต้องได้ข้อมูลนำเข้าที่ถูกต้องด้วยเช่นกัน สรุปคือจะต้องพยายามให้จุดที่เรากำหนดเข้ายึดติดกับขอบของวัตถุที่สนใจให้ได้มากที่สุด หรือเรียกว่า คอนทัวร์ของภาพเพื่อนำไปใช้งานในการสร้างแบบจำลองภาพสามมิติต่อไป สำหรับส่วนที่สองจะเป็นการสร้างพื้นผิวจากข้อมูลคอนทัวร์ที่ได้จากส่วนแรก โดยที่ในงานวิจัยใช้เทคนิควิธีสองรูปแบบ คือ เป็นการสร้างจากจุดตั้งต้นที่เป็นแบบสามมิติคือพิกัด (x, y, z) เริ่มจากการแบ่งจุดเหล่านี้เป็นบริเวณเล็ก ๆ ก่อน

(Neighborhoods) จากนั้นให้เปรียบเทียบกันว่าบริเวณเล็ก ๆ เหล่านี้คือหนึ่งระนาบเล็ก ๆ บนพื้นผิวดังนั้นเมื่อมีระนาบหลาย ๆ ระนาบมาต่อกันก็จะได้พื้นผิวทั้งหมด เพราะฉะนั้นเราจึงจำเป็นต้องทำการหาเวกเตอร์ตั้งฉาก (Normal Vector) ก่อน เพื่อเป็นการบอกทิศทางในการสร้างพื้นผิวว่า จะต้องสร้างพื้นผิวในทิศทางที่ตั้งฉากกับเวกเตอร์ตัวดังกล่าว เพราะฉะนั้นในภาพหนึ่งภาพจะได้เวกเตอร์ตั้งฉากจำนวนมาก ดังนั้นเพื่อให้ได้พื้นผิวที่ถูกต้องจึงต้องให้เวกเตอร์เหล่านี้พุ่งไปในทิศทางเดียวกัน คือพุ่งเข้าภายในพื้นผิว หรือพุ่งออกนอกพื้นผิวทั้งหมด วิธีการคือต้องหาเส้นทางที่สั้นที่สุดไปยังทุกจุด แต่น้ำหนักของเส้นทางจะหาได้จากพลังงานในการกลับเวกเตอร์ เพราะฉะนั้นจากจุดนี้จะได้เส้นทางที่สั้นที่สุดและใช้พลังงานน้อยที่สุดในการปรับเวกเตอร์ให้ไปในทิศทางเดียวกัน ขั้นสุดท้ายคือการเชื่อมลเหล่านี้มาสร้างพื้นผิวแบบลูกบาศก์ โดยการนำลูกบาศก์ใหญ่มาครอบข้อมูลแล้วแบ่งเป็นลูกบาศก์เล็ก ๆ หรือกริดนั่นเอง จากนั้นก็หาระยะทางจากมุมของลูกบาศก์แต่ละมุมจนถึงระนาบ โดยจะเป็นตัวบอกว่าที่ลูกบาศก์นี้จะมีพื้นผิวของภาพอยู่หรือไม่ ถ้ามีก็จะทำการสร้างโดยจะทำการสร้างตามรูปแบบที่กำหนดไว้ สุดท้ายแล้วจะได้พื้นผิวของวัตถุที่เราต้องการ สำหรับวิธีนี้ภายในงานวิจัยได้พบปัญหาในขั้นตอนการหาเวกเตอร์ตั้งฉาก พบเวกเตอร์บางบริเวณพุ่งไปในทิศทางที่ผิด ทำให้ภาพจำลองที่สร้างขึ้นมานี้มีความผิดพลาด จึงได้มีการหันมาใช้วิธีถัดมา คือการสร้างพื้นผิวจากการขยายตัวของฟังก์ชันทรงกลมแบบฮาร์โมนิก (Spherical Harmonic) เป็นการเปลี่ยนพิกัดให้มาอยู่ในรูปของพิกัดทรงกลมจะให้ผลใกล้เคียงกับภาพต้นแบบ โดยจะขึ้นอยู่กับลำดับ (order) ยิ่งถ้าลำดับเยอะจะสามารถสร้างรูปแบบจำลองได้ใกล้เคียงกับรูปจริง ส่วนสุดท้ายคือการปรับพื้นผิวเพียงบางส่วน จะเป็นการที่ผู้ใช้พบว่ามีบางพิกัดที่ไม่ถูกต้อง และต้องการแก้ไข ผู้ใช้จะสามารถแก้ไขตรงจุดนั้นได้ จากนั้นระบบจะทำการคำนวณหาว่าอยู่บนระนาบใดบนคอนทัวร์แล้วทำการปรับพื้นผิวที่จุดนั้นและบริเวณใกล้เคียงเท่านั้นเพื่อเป็นการลดเวลาในการสร้างแบบจำลองใหม่ทั้งหมด

5.2 ข้อจำกัดของงานวิจัย

งานวิจัยนี้เป็นการสร้างแบบจำลองสามมิติของทารกในครรภ์จากเครื่องถ่ายภาพเหนือเสียง (Ultrasonography) ดังนั้นจึงมีข้อจำกัดดังนี้

1) ภาพที่ได้คุณภาพค่อนข้างต่ำเนื่องจากเป็นภาพ Ultrasound แบบสองมิติ ทำให้มีองค์ประกอบอื่นปะปนมาก่อนข้างมาก อาจเป็นถุงน้ำค้ำ อวัยวะภายใน หรือสัญญาณรบกวนขณะถ่ายภาพ ดังนั้นจึงไม่มีความคมชัดของขอบสักเท่าไร ทำให้ผู้ใช้ที่ไม่มีความรู้เรื่องนี้ แยกองค์ประกอบของภาพได้ค่อนข้างลำบากเพราะไม่ทราบว่า ส่วนใดคือองค์ประกอบของวัตถุที่สนใจ จะสร้างแบบจำลองสามมิติ

- 2) ขั้นตอนการกำหนดจุด ผู้ใช้อาจกำหนดไปทับกับขอบขององค์ประกอบอื่น ๆ อาจทำให้ระบบคำนวณออกมาผิดพลาด และต้องพยายามทำให้ใกล้เคียงกับขอบให้มากที่สุด
- 3) ผู้ใช้จำเป็นต้องหาองค์ประกอบของภาพหลายภาพ อาจทำให้รู้สึกเมื่อยล้าได้
- 4) หากมีจุดคอนทราสต์ค่อนข้างมาก อาจทำให้ระบบคำนวณได้ช้า แต่หากมีข้อมูลน้อยไปก็อาจเกิดการสร้างภาพที่ผิดพลาดได้
- 5) หากต้องการความละเอียดสูง จึงอาจจะส่งผลให้คอมพิวเตอร์ที่มีทรัพยากรน้อยเกิดอาการเครื่องค้าง

5.3 ข้อเสนอแนะในงานวิจัยต่อไป

สำหรับขั้นตอนของการสร้างพื้นผิวด้วยฟังก์ชันการขยายทรงกลมแบบฮาร์โมนิกนั้นมีสัมประสิทธิ์ที่หาได้จากข้อมูลต้นแบบ ดังนั้นอาจมีอีกหลายวิธีที่สามารถพัฒนาเพื่อให้ได้ข้อมูลที่ใกล้เคียงกับข้อมูลต้นแบบมากที่สุด โดยแนวทางวิจัยที่จะพัฒนาต่อไปได้คือ

- 1) ขั้นตอนของการหาสัมประสิทธิ์เพื่อให้ได้ใกล้เคียงกับพื้นผิวต้นแบบมากที่สุด
- 2) ขั้นตอนของการแยกองค์ประกอบของภาพหรือการหาคอนทราสต์ อาจมีการเพิ่มพลังงานบางเทอมเข้าไปเพื่อให้วัตถุสามารถยึดกับขอบภาพให้ได้มากที่สุด
- 3) สามารถนำระบบนี้ไปใช้ได้กับทุกข้อมูลต้นแบบ และใช้ได้กับการสร้างแบบจำลองของอวัยวะทุกส่วน

รายการอ้างอิง

- Bill Toll, Fuhua Cheng, (1999). **Surface Reconstruction from Point Clouds.**
- Chaganti, Venkata Ravikiran, (2005). **Edge Detection of Noisy Images using 2-d Discrete Wavelet Transform.**
- Cormen, Thomas H, Leiserson, Charie E., Rivest Ronal L., Stein clifford, (2001). **Introduction to Algorithms**, 2nd McGraw-Hill.
- David Eberly, 1998-2008, **Geometric Tools**, LLC, <http://www.geometricktools.com/>.
- Emanuele Trucco, Alessandro Verri, 1998, **Introductory Techniques for 3-D Computer Vision**, Prentice Hall, 1998.
- Herve Abdi , Lynne J. Williams,(2010), **Principal Component Analysis.**
- Hill, C. R., Bamber, J. C., and ter Haar, G. R., Jeongju Choi, Jong Shik Kim, **Physical Principles of Medical Ultrasonics**, 2nd John Wiley&Sons, 2002.
- Julio Carballido-Gamio, Serge J. Belongie, and Sharmila Majumdar, **Normalized cuts in 3-D for spinal MRI segmentation**. IEEE Trans. Med. Imaging 23(1): 36-44 (2004).
- K.M.Lam and H Yan, 1993. **Fast greedy algorithm for active contours.**
- Marián Bakoš, 2007, **Active Contours and their Utilization at Image Segmentation.**
- Martin Styner, Ipek Oguz, Shun Xu, Christian Brechbuhler, Dimitrios Pantazis, Guido Gerig, 2006, **Framework for the Statistical Shape Analysis of Brain Structures using SPHARM-PDM.**
- Micheal Kass, Andrew Witkin and Demetri Tozopoulos, 1988. **Snakes: Active contour models.**
- Rindsay I Smith, 2002, **A tutorial on Principal Components Analysis.**
- Sandra V. B. Jardim, Mário A. T. Figueiredo, 2003. **Automatic Contour Estimation In Fetal Ultrasound Images.**
- Sittichai P., Paramate H., Saowapak S., **Multi-resolution ICP using real-valued spherical harmonic expansion and its application in computer-aided dental** , The 3rd International Symposium on Biomedical Engineering (ISBME 2008).
- Susanna S. Epp, 1995. **Discrete Mathematics with Applications**, 2nd ed. Brooks/Cole Publishing.

Walpole, R.E., Myers, R.H., 1990. **Probability and statistics for engineers and scientists.** 4th ed.
New York: Macmillan.

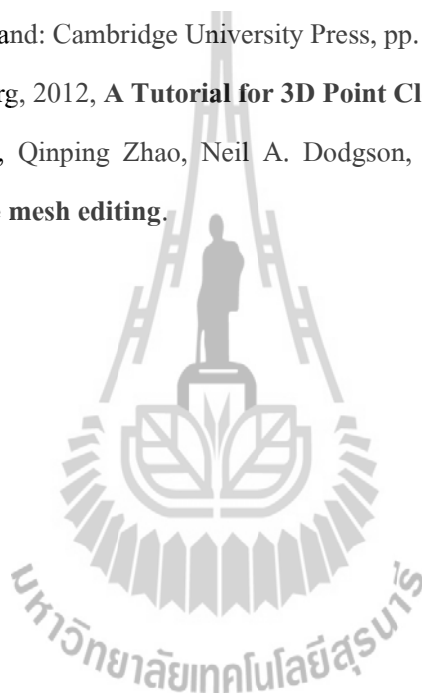
Paul Baker. 2002-2007. **Functions to evaluate a spherical harmonic basis function at a given point.**

William E. Lorensen and Harvey E. Cline, 1987, **Marching cubes: A high resolution 3D surface construction algorithm.**

W.H. Press, B.P. Flannery, S. A. Teukolsky, and W. Vetterling, 1988, "**Spherical Harmonics.**"
§6.8 in Numerical Recipes in FORTRAN: The Art of Scientific Computing, 2nd ed.
Cambridge, England: Cambridge University Press, pp. 246-248, 1992

Yue Li, Matthew Hielsberg, 2012, **A Tutorial for 3D Point Cloud Editor.**

Yujian Gao, Aimin Hao, Qiping Zhao, Neil A. Dodgson, 2009, **Skin-detached surface for interactive large mesh editing.**





ภาคผนวก ก

รหัสต้นฉบับโปรแกรม

โค้ดจำลองการสร้างแบบจำลอง 3 มิติด้วยการขยายค่าจริงของฟังก์ชัน

ทรงกลมแบบฮาร์โมนิก

```
// pseudo code to Generate surface from Spherical Harmonics
**contour //Set of point from Active Contour(x, y, z)
maxOrder = getOrderFromUser(); //get order Spherical
harmonics from user
sizeOfclm = SizeOfOrder(maxOrder) //get size Of clm
clmx = (double *) malloc(sizeOfclm * sizeof(double));
clmy = (double *) malloc(sizeOfclm * sizeof(double));
clmz = (double *) malloc(sizeOfclm * sizeof(double));

for l = 0 to maxOrder {
    ZZ = ZVx = ZVy = ZVz = 0.0;
    for m = -l to l {
        for i = 0 to n { //n number of **contour
            //V is original point **contour(x, y, z)
            //Zl,m = Ylm(θ, φ)
            Vx[i] = contour[i][0];
            Vy[i] = contour[i][1];
            Vz[i] = contour[i][2];
        }
        ///////////////////////////////////////////////////


$$Y_l^m(\theta, \phi) = \sqrt{\frac{2l+1}{4\pi} \frac{(l-|m|)!}{(l+|m|)!}} P_l^{|m|}(\theta) \Phi(\phi)$$


$$\Phi(\phi) = \begin{cases} \sqrt{2} \cos m\phi, & m < 0 \\ 1, & m = 0 \\ \sqrt{2} \sin |m|\phi, & m > 0 \end{cases}$$


$$P_0^0(\cos\theta) = 1$$


$$P_m^m(\cos\theta) = (2m-1) \sin\theta P_{m-1}^{m-1}(\cos\theta)$$


$$P_{m+1}^m(\cos\theta) = (2m+1) \cos\theta P_m^m(\cos\theta)$$


$$P_l^m(\cos\theta) = \frac{(2l-1) \cos\theta P_{l-1}^m(\cos\theta) - (l+m-1) P_{l-2}^m(\cos\theta)}{l-m}$$


$$r = \sqrt{x^2 + y^2 + z^2}$$


$$\theta = \tan^{-1} \left( \frac{y}{x} \right)$$


$$\phi = \cos^{-1} \left( \frac{z}{r} \right)$$

        ///////////////////////////////////////////////////

        Ylm[i] =  $\sqrt{\frac{2l+1}{4\pi} \frac{(l-|m|)!}{(l+|m|)!}} P_l^{|m|}(\theta) \Phi(\phi)$ 
        ZZ += (Ylm[i] * Ylm[i]);
        ZVx += (Yml[i] * Vx[i]);
        ZVy += (Yml[i] * Vy[i]);
        ZVz += (Yml[i] * Vz[i]);
    }
}
```



```

        clmx[l*(l+1)+m] = ZZ * ZVx;
        clmy[l*(l+1)+m] = ZZ * ZVy;
        clmz[l*(l+1)+m] = ZZ * ZVz;
    }
}

//Spherical harmonics point to create surface
//  $x(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l c_l^m Y_l^m(\theta, \phi)$ 
for i = 0 to gridsize { //gridsize control smooth of surface

    theta = (double)i / (double)gridsize * M_PI;

    for j = 0 to gridsize {

        phi = (double)j / (double)gridsize * M_PI * 2.0;
        ctsX [i][j] = ctsY [i][j] = ctsZ [i][j] = 0.0;

        for l = 0 to maxOrder {

            for m = -l to l {

                index = l * (l + 1) + m;
                Ylm = SH (l, m, theta, phi);
                ctsX[i][j] += (m_cX[index] * Ylm);
                ctsY[i][j] += (m_cY[index] * Ylm);
                ctsZ[i][j] += (m_cZ[index] * Ylm);

            }

        }

    }

}

```

โค้ดจำลองการสร้างพื้นผิวใหม่บางบริเวณของแบบจำลอง 3 มิติ

```
//pseudo code for generate new surface at error area
if error == 1 then    //get an error index contour and (x1,
y1, z1) from user
    idxError = getPixelError();
endif

//Find new clmx, clmy, clmz to generate surface
if error == 1 then
    // idxErrorold use old pixel to compute (x, y, z)
    // idxErrornew use new pixel to compute (x1, y1, z1)
    ZZ = ZZ - (Ylm[idxErrorold] * Ylm[idxErrorold]) +
        (Ylm[idxErrornew] * Ylm[idxErrornew]);
    ZVx = ZVx - (Yml[idxErrorold] * Vx[idxErrorold]) +
        (Yml[idxErrornew] * Vx[idxErrornew]);
    ZVy = ZVy - (Yml[idxErrorold] * Vy[idxErrorold]) +
        (Yml[idxErrornew] * Vy[idxErrornew]);
    ZVz = ZVz - (Yml[idxErrorold] * Vz[idxErrorold]) +
        (Yml[idxErrornew] * Vz[idxErrornew]);

endif

//Create new surface at error area
if error == 1 then
    //get theta1 and phi1 from error contour to create new
    //surface at error area
    theta1 = ConvertCartesian2Spherical(idxError);
    phi1 = ConvertCartesian2Spherical(idxError);
    for l = 0 to maxOrder {
        for m = -l to l {
            index = l * (l + 1) + m;
            Ylm = SH (l, m, theta1, phi1);
            ctsX += (m_cX[index] * Ylm);
            ctsY += (m_cY[index] * Ylm);
            ctsZ += (m_cZ[index] * Ylm);
        }
    }
    updateSurface(ctsX, ctsY, ctsZ);
endif
```

ประวัติผู้เขียน

นายกฤษฎ์ภาณุ โสภานุตร เกิดเมื่อวันศุกร์ที่ 25 ตุลาคม พ.ศ. 2528 ที่อำเภอเมือง จังหวัด นครพนม จบการศึกษาระดับประถมศึกษาที่โรงเรียนอนุบาลนครพนม สำเร็จการศึกษาระดับชั้น มัธยมศึกษาตอนต้นและตอนปลายจากโรงเรียนปิยะมหาราชาลัย ตำบลในเมือง อำเภอเมือง จังหวัด นครพนม จากนั้นปี พ.ศ. 2547 เริ่มศึกษาในระดับปริญญาตรี สำนักวิชาวิศวกรรมศาสตร์ สาขาวิชา วิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ขณะที่ทำการศึกษาในระดับปริญญาตรีใน ภาควิชาการศึกษาศุทธิยได้ร่วมสหกิจศึกษากับบริษัท เซเว่น สตาร์ มีเดีย เซอร์วิส (บริษัท จำกัด) ฝ่าย การเขียนโปรแกรมพัฒนาเว็บไซต์ จังหวัดกรุงเทพมหานคร เป็นระยะเวลาทั้งสิ้น 4 เดือน และได้ไป ศึกษาด้าน Image Processing เพิ่มเติมที่ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) พร้อมทั้งช่วยพัฒนาระบบเป็นระยะเวลาประมาณ 2 เดือน ต่อมาสำเร็จการศึกษาเมื่อ ปี พ.ศ. 2551 และเข้าศึกษาต่อในระดับปริญญาโทที่ สาขาวิศวกรรมคอมพิวเตอร์ ณ สถาบัน เดิม โดยได้รับทุนการศึกษาระดับบัณฑิตศึกษาจากแหล่งทุนภายนอก จากโครงการทุนสถาบัน บัณฑิตวิทยาศาสตร์และเทคโนโลยีไทย (TGIST) สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยี แห่งชาติ (สวทช.) ในหัวข้อวิจัย การสร้างและวิเคราะห์แบบจำลอง 3 มิติของทารกในครรภ์จาก ข้อมูลเครื่องถ่ายภาพเหนือเสียง ระหว่างศึกษาได้รับความอนุเคราะห์ดียิ่งจากคณาจารย์ในสาขา โดยได้รับวางใจให้เป็นผู้ช่วยสอนในวิชาการโปรแกรมโดยยึดเหตุการณ์ และวิชาฐานข้อมูล