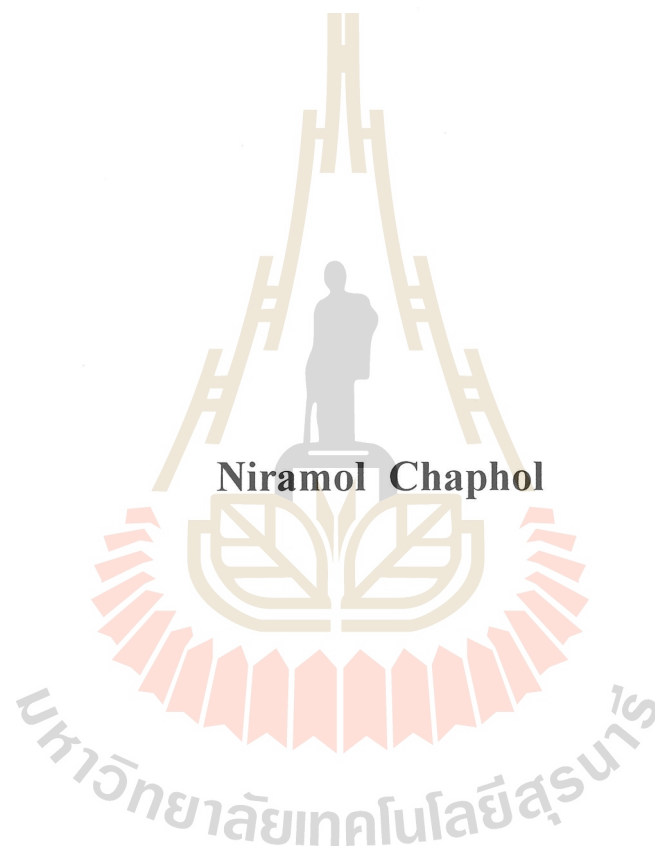


การศึกษาการกำหนดตำแหน่งการตัดส่วนที่ไม่ใช่ชิ้นงานในกระบวนการ
Laminated Object Manufacturing (LOM)



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมอุตสาหการ
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2559

**A STUDY OF IDENTIFICATION CUTTING POSITIONS
FOR NON-OBJECT AREA IN LAMINATED OBJECT
MANUFACTURING (LOM)**



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

Suranaree University of Technology

Academic Year 2016

การศึกษาการกำหนดตำแหน่งการตัดส่วนที่ไม่ใช่ชิ้นงาน ในกระบวนการ

Laminated Object Manufacturing (LOM)

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์



(ผศ. ดร.พงษ์ชัย จิตตะมัย)

ประธานกรรมการ



(ผศ. ดร.ปภากร พิทยชาวล)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)



(ผศ. ดร.ประเมศวร์ ห่อแก้ว)

กรรมการ



(รศ. ดร.พรศิริ จงกล)

กรรมการ



(อ. ดร.จงกล ศรีธร)

กรรมการ



(ศ. ดร.สันติ แม่นศิริ)

รองอธิการบดีฝ่ายวิชาการ

และพัฒนาความเป็นสากล



(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

นิรมล ชาพล : การศึกษาการกำหนดตำแหน่งการตัดส่วนที่ไม่ใช่ชิ้นงาน ในกระบวนการ
Laminated Object Manufacturing (LOM) (A STUDY OF IDENTIFICATION
CUTTING POSITIONS FOR NON-OBJECT AREA IN LAMINATED OBJECT
MANUFACTURING (LOM)) อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.ปภากร พิทยชาล,
104 หน้า

ในปัจจุบันการสร้างชิ้นงานต้นแบบเป็นกระบวนการสำคัญของการออกแบบและพัฒนาผลิตภัณฑ์เป็นอย่างยิ่ง การสร้างชิ้นงานต้นแบบรวดเร็ว Rapid Prototyping (RP) ด้วยเทคนิค Laminated Object Manufacturing (LOM) เป็นวิธีการหนึ่งในการสร้างชิ้นงานต้นแบบรวดเร็ว ที่มีข้อดีคือ วัสดุตั้งต้นที่ใช้เป็นแผ่นบางมีราคาถูก ไม่ต้องสร้างฐานรองรับชิ้นงาน เป็นวิธีรวดเร็วและประหยัด แต่จะต้องมีขั้นตอน De-cubing หลังการสร้างชิ้นงานต้นแบบ โดยทำการตัดส่วนที่ไม่ใช่ชิ้นงานออกเป็นชิ้นเล็กๆ เพื่อให้สามารถแยกส่วนที่เป็นชิ้นงานออกมาได้ง่าย ลดเวลาและลดความเสียหายบริเวณผิวขอบชิ้นงานอีกด้วย

งานวิจัยนี้จึงได้วิเคราะห์การกำหนดตำแหน่งการตัดส่วนที่ไม่ใช่ชิ้นงาน สำหรับกระบวนการ De-cubing ของการขึ้นรูปในแต่ละชั้น โดยวิเคราะห์ผ่านโปรแกรมคอมพิวเตอร์ โดยผลลัพธ์ที่ได้จะเป็นรูปภาพที่แสดงเส้นทางการทำ De-cubing จำนวนครั้งที่เครื่องมือตัดทำงาน และจำนวนพิกเซลที่เครื่องมือตัดต้องทำงานทั้งหมด

ผลการวิจัยพบว่าเวลาและความละเอียดของชิ้นงานต้นแบบแปรผกผันกับค่าอัตราส่วน พิกเซลสีดำต่อพิกเซลสีขาวที่กำหนดในโปรแกรม (τ) และค่าจำนวนพิกเซลที่เล็กที่สุดของกล่องขอบเขตย่อยที่จะพิจารณาต่อตามเงื่อนไขที่กำหนด (β) โดยหาก τ และ β มีค่าน้อย จะส่งผลให้ชิ้นงานต้นแบบที่ได้เหมือนจริง และมีความละเอียดของผิวชิ้นงานมากขึ้น แต่ต้องใช้เวลาในการสร้างชิ้นงานมากขึ้นด้วย หาก τ และ β มีค่ามาก จะส่งผลให้ชิ้นงานต้นแบบที่ได้ไม่เหมือนจริง และมีความหยาบของผิวชิ้นงานมากขึ้น

สาขาวิชา วิศวกรรมอุตสาหการ
ปีการศึกษา 2559

ลายมือชื่อนักศึกษา ณัฐชา
ลายมือชื่ออาจารย์ที่ปรึกษา ปภากร
ลายมือชื่ออาจารย์ที่ปรึกษาร่วม ปภากร และ พัทธ

NIRAMOL CHAPHOL : A STUDY OF IDENTIFICATION CUTTING
POSITIONS FOR NON-OBJECT AREA IN LAMINATED OBJECT
MANUFACTURING (LOM). THESIS ADVISOR : ASST. PROF.
PAPHA KORN PITAYACHAVAL, Ph.D., 104 PP.

RAPID PROTOTYPING TECHNOLOGY (RP)/ LAMINATE OBJECT
MANUFACTURING/ IMAGE PROCESSING/ ADAPTIVE DE-CUBING/ LAYER
BY LAYER

Rapid prototyping technology (RP) is a technology, that reduce time of a product design and development process, to construct prototype layer by layer. Since there are three types of starting material states for creating RP model, a Laminated Object Manufacturing (LOM) technique creates prototype base on solid material, such as paper sheet and plastic sheet. The advantages of LOM technique are low prices materials and unnecessary for creating support structure according to the waste materials are removed after model is formed. To separate model sheet and waste sheet, a de-cubing process is applied to assign shape of waste material into generally small square shape that can be easily remove in order to reduce time and avoid prototype damages.

To improve the de-cubing process, this research presents method to identify the cutting points for de-cubing process base on the MATLAB program. An image of each layer is import as an input of algorithm and an output cutting points that can be used to generate tool path planning for adaptive de-cubing process in order to calculate total of cutting points (time) and total cutting length (pixel).

The results were shown that the construction time and boundary roughness of prototype relates to the threshold (τ), which is a number of black pixels per number of white pixels, and minimize pixel of each sub-bounding box (β). If the τ and β is precious small, then a physical model should be has smooth boundary surface, while spent a lot of time for created physical model. On the other hand, if τ and β is large, these parameters conduct the physical model are rough surface and wrinkle prototype.



School of Industrial Engineering

Academic Year 2016

Student's Signature _____ ณัฐดนัย

Advisor's Signature _____ ณัฐดนัย

Co-Advisor's Signature _____ ณัฐดนัย

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี เนื่องจากได้รับความกรุณาอย่างสูงจาก ท่านผู้ช่วยศาสตราจารย์ ดร.ปภากร พิทยवाल อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งได้ให้โอกาสทางการศึกษา และกรุณาให้คำปรึกษาแนะนำ ซึ่งแนะแนวทางการศึกษาด้วยความเอาใจใส่อย่างยิ่ง รวมถึงยังให้ข้อคิดในการดำเนินชีวิตและการทำงาน ผู้วิจัยตระหนักถึงความตั้งใจและความทุ่มเทของท่านอาจารย์เป็นอย่างยิ่ง ผู้วิจัยขอกราบขอบพระคุณอย่างสูงไว้ ณ โอกาสนี้

ขอกราบขอบพระคุณ ท่านผู้ช่วยศาสตราจารย์ ดร.ปรเมศวร์ ห่อแก้ว อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ที่ได้ให้คำปรึกษาแนวความคิดต่างๆ และชี้แนะแนวทางในดำเนินงานวิจัย

ขอกราบขอบพระคุณ ท่านรองศาสตราจารย์ ดร.พรศิริ จงกล ท่านผู้ช่วยศาสตราจารย์ ดร.พงษ์ชัย จิตตะมัย อาจารย์ ดร.จงกล ศรีธร และผู้ช่วยศาสตราจารย์ ดร.ปวีร์ ศิริรักษ์ ที่ได้กรุณาให้การอบรมสั่งสอน ให้คำแนะนำทั้งด้านการศึกษาและการดำเนินชีวิต รวมถึงให้ข้อเสนอแนะแก้ไข และแนวคิดต่างๆที่เป็นประโยชน์

ขอขอบพระคุณ มหาวิทยาลัยเทคโนโลยีสุรนารี ที่ให้ทุนสนับสนุนการศึกษาและการทำงานวิจัย

ท้ายนี้ ผู้วิจัยขอกราบขอบพระคุณบิดา มารดาที่ได้ให้การอุปการะเลี้ยงดู อบรมสั่งสอนเป็นอย่างดี และขอบพระคุณทุกคนในครอบครัวที่ให้ความรัก สนับสนุนการศึกษา ให้คำปรึกษาเป็นกำลังใจและเป็นแรงผลักดัน จนทำให้ผู้วิจัยประสบผลสำเร็จในชีวิต

นิรมล ชาพล

สารบัญ

หน้า

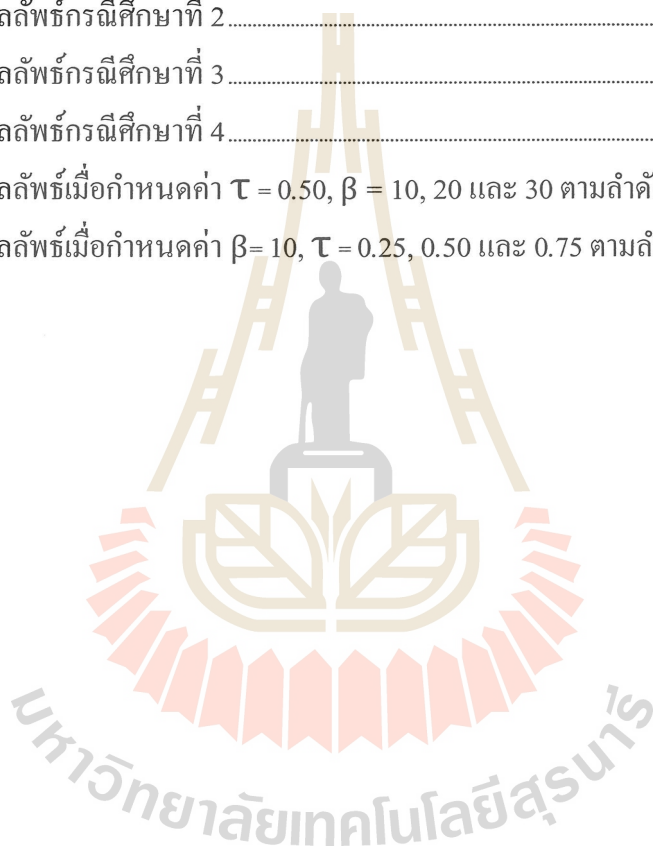
บทคัดย่อ (ภาษาไทย).....	ก
บทคัดย่อ (ภาษาอังกฤษ).....	ข
กิตติกรรมประกาศ.....	ง
สารบัญ.....	จ
สารบัญตาราง.....	ช
สารบัญรูป.....	ซ
คำอธิบายสัญลักษณ์และคำย่อ	ฎ
บทที่	
1 บทนำ.....	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	10
1.3 สมมติฐานการวิจัย.....	10
1.4 ขอบเขตการวิจัย.....	11
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	10
1.6 ส่วนประกอบของวิทยานิพนธ์.....	12
2 วรรณกรรมที่เกี่ยวข้อง.....	13
2.1 กรรมวิธีการสร้างต้นแบบ (Prototype).....	13
2.2 การสร้างชิ้นงานต้นแบบรวดเร็ว (Rapid Prototyping: RP).....	14
2.3 ทบทวนวรรณกรรม.....	21
2.4 ทฤษฎี Image Processing.....	23
2.5 โปรแกรม MATLAB.....	25
3 การดำเนินงานวิจัย.....	27
3.1 การดำเนินงานวิจัย.....	27
3.2 โครงสร้างโปรแกรมการวิเคราะห์ Adaptive de-cubing.....	29

สารบัญ (ต่อ)

	หน้า
4. ผลการทดสอบ.....	41
4.1 กรณีศึกษาที่ 1.....	41
4.2 กรณีศึกษาที่ 2.....	44
4.3 กรณีศึกษาที่ 3.....	47
4.4 กรณีศึกษาที่ 4.....	50
4.5 ผลลัพธ์เมื่อเปลี่ยนแปลงค่า β และ τ	52
5 สรุปและข้อเสนอแนะ.....	54
5.1 สรุปผลการวิจัย.....	54
5.1.1 การวิเคราะห์ผลการทดสอบระหว่าง Uniform de-cubing กับ Adaptive de-cubing	54
5.1.2 การวิเคราะห์ผลการทดสอบ Adaptive de-cubing เมื่อกำหนดค่า τ เท่ากับ 0.25, 0.50 และ 0.75.....	55
5.2 ข้อเสนอแนะในการทำงานวิจัยต่อไป.....	55
เอกสารอ้างอิง.....	56
ภาคผนวก	
ภาคผนวก ก. ตัวแปรในการทำงานของโปรแกรม.....	58
ภาคผนวก ข. โปรแกรม MATLAB.....	61
ภาคผนวก ค. การใช้งานโปรแกรม.....	82
ภาคผนวก ง. บทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างศึกษา.....	85
ประวัติผู้เขียน.....	104

สารบัญตาราง

ตารางที่	หน้า
4.1	แสดงผลลัพธ์กรณีศึกษาที่ 1 43
4.2	แสดงผลลัพธ์กรณีศึกษาที่ 2 46
4.3	แสดงผลลัพธ์กรณีศึกษาที่ 3 49
4.4	แสดงผลลัพธ์กรณีศึกษาที่ 4 51
4.5	แสดงผลลัพธ์เมื่อกำหนดค่า $\tau = 0.50$, $\beta = 10, 20$ และ 30 ตามลำดับ 53
4.6	แสดงผลลัพธ์เมื่อกำหนดค่า $\beta = 10$, $\tau = 0.25, 0.50$ และ 0.75 ตามลำดับ 53



สารบัญรูป

รูปที่	หน้า
1.1	แสดงกระบวนการออกแบบและพัฒนาผลิตภัณฑ์..... 2
1.2	แสดงการภาพตัวอย่างชิ้นงานที่เป็นไฟล์CAD และ STL ไฟล์ตามลำดับ..... 3
1.3	แสดงกระบวนการสร้างชิ้นงานต้นแบบด้วยวิธีการ Rapid Prototyping..... 4
1.4	แสดงการแบ่งประเภทของ RP ตามวัสดุตั้งต้นที่ใช้ในกระบวนการ..... 4
1.5	แสดงแผนผังกระบวนการของ LOM..... 6
1.6	แสดงกระบวนการ Laminate Object Manufacturing..... 7
1.7	แสดงชิ้นงานรูปทรงต่างๆ..... 7
1.8	แสดงชิ้นงานที่มีลักษณะซับซ้อน..... 7
1.9	แสดงกระบวนการ de-cubing..... 8
1.10	แสดงชิ้นงานก่อนการทำ De-cubing (ซ้าย)และ ชิ้นงานหลังการทำ De-cubing (ขวา)..... 8
1.11	แสดง (ก) ชิ้นงานสามมิติ (ข) ภาพหน้าตัดชิ้นงาน (ค) Uniform de-cubing (ง) Adaptive de-cubing..... 9
2.1	แสดงการแบ่งประเภทของ RP ตามวัสดุตั้งต้น ที่ใช้ในกระบวนการ..... 15
2.2	แสดงกระบวนการขึ้นรูปชิ้นงานด้วยเทคนิค SLA..... 16
2.3	แสดงกระบวนการขึ้นรูปชิ้นงานด้วยเทคนิค FDM..... 17
2.4	แสดงกระบวนการขึ้นรูปชิ้นงานด้วยเทคนิค SLS..... 18
2.5	แสดงกระบวนการขึ้นรูปชิ้นงานด้วยกระบวนการ SGC..... 19
2.6	แสดงกระบวนการขึ้นรูปชิ้นงานด้วยเทคนิค LOM..... 19
2.7	แสดงแผนผังกระบวนการของ LOM..... 20
2.8	แสดงตัวอย่างชิ้นงานต้นแบบที่สร้างด้วยวิธีการ Rapid Prototyping..... 21
2.9	แสดงวิธีการตัดชิ้นงานแบบใหม่..... 22
2.10	แสดงกระบวนการประมวลผลภาพ..... 24

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.1	แสดงกรอบแนวคิดงานวิจัย.....28
3.2	แสดงวิธีการดำเนินงานทั้งหมด.....28
3.3	แสดงโครงสร้างโปรแกรมการวิเคราะห์ Adaptive de-cubing.....29
3.4	แสดงโครงสร้างโปรแกรมการวิเคราะห์ Adaptive de-cubing (ต่อ).....30
3.5	แสดงตัวอย่างชิ้นงาน 3 มิติ.....30
3.6	แสดงรูปภาพนำเข้า.....31
3.7	แสดงรูปภาพนำเข้าหลังจากถูกแปลงเป็นภาพขาว-ดำ.....31
3.8	แสดงการสร้างกล่องขอบเขต.....32
3.9	แสดงการแบ่งกล่องขอบเขตออกเป็น 4 กล่องย่อย.....33
3.10	แสดงผลการตรวจสอบค่า β ในแกน X และแกน Y.....34
3.11	แสดงตัวอย่างการตรวจสอบค่า β35
3.12	แสดงค่าต่างๆ ที่ต้องใช้การตรวจสอบค่า τ ในแกน X และแกน Y.....36
3.13	แสดงตัวอย่างลำดับการตรวจสอบค่า τ37
3.14	แสดงผลการตรวจสอบค่า τ ในแกน X และแกน Y ของกล่องขอบเขตย่อย.....38
3.15	แสดงลำดับการวิเคราะห์การแบ่งกล่องย่อยเมื่อกำหนด $\beta = 10, \tau = 0.50$39
4.1	แสดงชิ้นงานสามมิติกรณีศึกษาที่ 1.....41
4.2	แสดงภาพหน้าตัดชิ้นงานสามมิติกรณีศึกษาที่ 1.....41
4.3	แสดงข้อมูลของรูปภาพนำเข้ากรณีศึกษาที่ 1.....42
4.4	แสดงผลลัพธ์เปรียบเทียบในกรณีศึกษาที่ 1.....43
4.5	แสดงชิ้นงานสามมิติกรณีศึกษาที่ 2.....44
4.6	แสดงภาพหน้าตัดชิ้นงานสามมิติกรณีศึกษาที่ 2.....44
4.7	แสดงข้อมูลของรูปภาพนำเข้ากรณีศึกษาที่ 2.....45
4.8	แสดงผลลัพธ์เปรียบเทียบในกรณีศึกษาที่ 2.....46
4.9	แสดงชิ้นงานสามมิติกรณีศึกษาที่ 3.....47
4.10	แสดงภาพหน้าตัดชิ้นงานสามมิติกรณีศึกษาที่ 3.....47

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.11	แสดงข้อมูลของรูปภาพนำเข้ากรณีศึกษาที่ 3.....48
4.12	แสดงผลลัพธ์เปรียบเทียบในกรณีศึกษาที่ 3.....49
4.13	แสดงภาพหน้าตัดชิ้นงานสามมิติกรณีศึกษาที่ 4.....50
4.14	แสดงข้อมูลของรูปภาพนำเข้ากรณีศึกษาที่ 4.....50
4.15	แสดงผลลัพธ์เปรียบเทียบในกรณีศึกษาที่ 3.....52
ค.1.1	แสดงการนำเข้าไฟล์รูปภาพ.....83
ค.1.2	แสดงการกำหนดค่า β ในโปรแกรม.....83
ค.1.3	หน้าการแสดงผลการรันโปรแกรมวิเคราะห์ Uniform de-cubing.....83
ค.1.4	แสดงการนำเข้าไฟล์รูปภาพ.....84
ค.1.5	แสดงการกำหนดค่า τ ในโปรแกรม.....84
ค.1.6	หน้าการแสดงผลการรันโปรแกรมวิเคราะห์ Adaptive de-cubing.....84

คำอธิบายสัญลักษณ์และคำย่อ

SB	=	ขนาดของกล่องขอบเขตของรูปภาพเส้นโครงร่างที่นำเข้า
SB_X	=	ขนาดของกล่องขอบเขตด้านแกน X
Bx_0	=	พิกเซลที่เป็นสีดำพิกเซลแรกของรูปภาพด้านแกน X
Ex_0	=	พิกเซลที่เป็นสีดำพิกเซลสุดท้ายของรูปภาพด้านแกน X
SB_Y	=	ขนาดของกล่องขอบเขตด้านแกน Y
By_0	=	พิกเซลที่เป็นสีดำพิกเซลแรกของรูปภาพด้านแกน Y
Ey_0	=	พิกเซลที่เป็นสีดำพิกเซลสุดท้ายของรูปภาพด้านแกน Y
C_X	=	ตำแหน่งพิกเซลที่แบ่งครึ่งกล่องขอบเขตด้านแกน X
C_Y	=	ตำแหน่งพิกเซลที่แบ่งครึ่งกล่องขอบเขตด้านแกน Y
BP_X	=	จำนวนพิกเซลสีดำในแกน X
τ	=	ค่าอัตราส่วนของจำนวนพิกเซลสีดำต่อพิกเซลสีขาว
P_X	=	จำนวนพิกเซลทั้งหมดในแกน X
BP_Y	=	จำนวนพิกเซลสีดำในแกน Y
P_Y	=	จำนวนพิกเซลทั้งหมดในแกน Y
β	=	จำนวนพิกเซลที่เล็กที่สุดของกล่องขอบเขตย่อยที่จะพิจารณาต่อตามเงื่อนไขที่กำหนด
R	=	จำนวนรอบในการตรวจสอบ
R_a	=	ความหนาแน่นของชิ้นงาน
S	=	ระยะทางการเดินทางของเครื่องมือตัด
T	=	ระยะเวลาในการสร้างชิ้นงานต้นแบบ
P_C	=	จำนวนพิกเซลที่เครื่องมือตัดต้องเดินทางทั้งหมดสำหรับขั้นตอน De-cubing
P_{CX}	=	จำนวนพิกเซลที่เครื่องมือตัดต้องเดินทางทั้งหมดสำหรับขั้นตอน De-cubing ในแกน X
P_{CY}	=	จำนวนพิกเซลที่เครื่องมือตัดต้องเดินทางทั้งหมดสำหรับขั้นตอน De-cubing ในแกน Y
N_{CX}	=	จำนวนครั้งที่เครื่องมือต้องตัดชิ้นงานสำหรับขั้นตอน De-cubing ในแกน X
N_{CY}	=	จำนวนครั้งที่เครื่องมือต้องตัดชิ้นงานสำหรับขั้นตอน De-cubing ในแกน Y

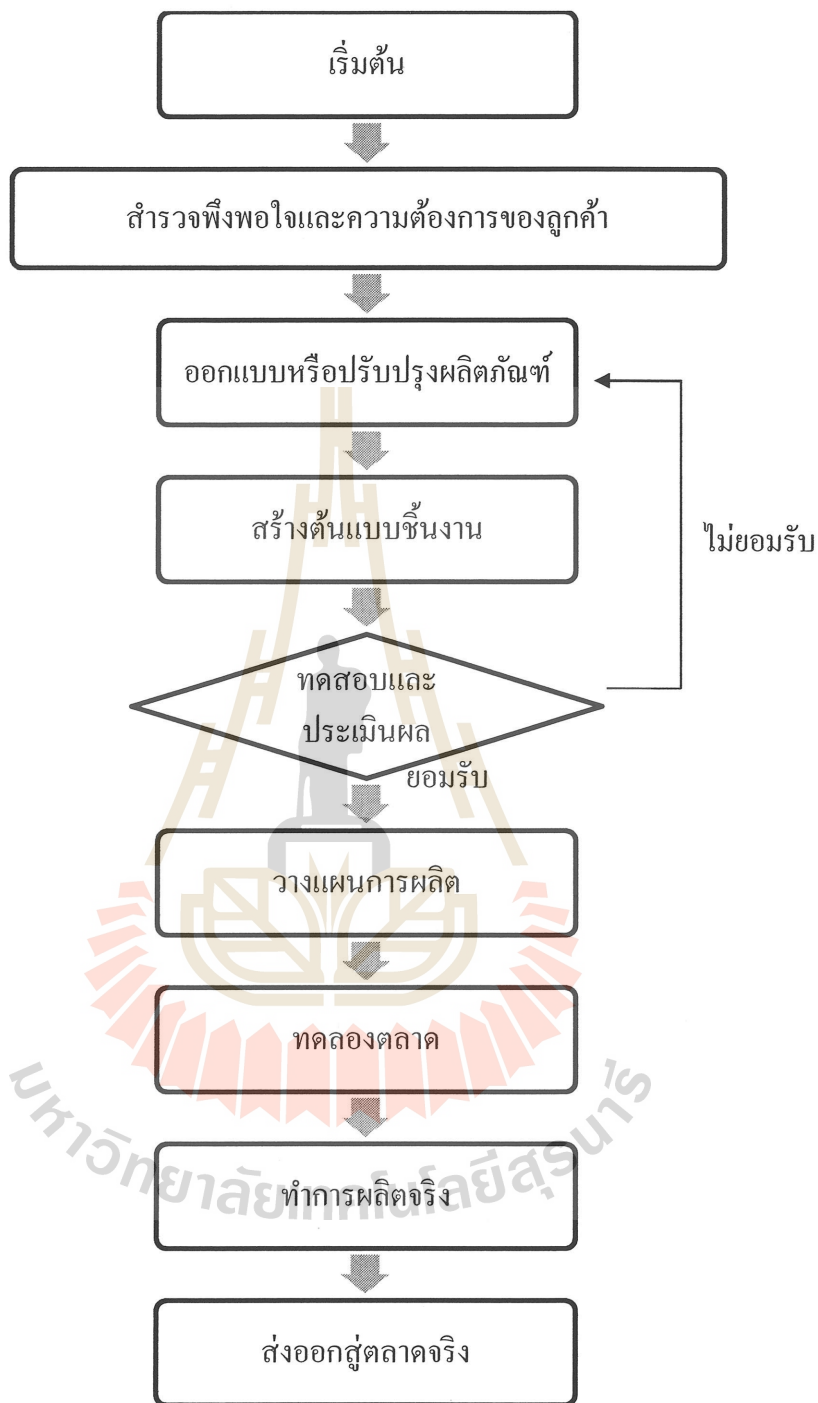
บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันการสร้างชิ้นงานต้นแบบ ถือเป็นกระบวนการสำคัญของการออกแบบและพัฒนาผลิตภัณฑ์ทั้งในทางการแพทย์ และภาคอุตสาหกรรมผลิตต่างๆ เป็นอย่างยิ่ง เพราะก่อนทำการผลิตชิ้นงานจริงจะต้องมีการออกแบบและสร้างชิ้นงานต้นแบบขึ้นมาก่อน เพื่อนำมาศึกษาและวิเคราะห์ถึงคุณสมบัติในด้านต่างๆ ว่าชิ้นงานจะสามารถตอบสนองความต้องการของลูกค้าได้หรือไม่ หากต้องมีการแก้ไขปรับปรุงสามารถแก้ไขที่ต้นแบบจนได้ต้นแบบที่ครบถ้วนสมบูรณ์แล้ว จึงค่อยทำการผลิตสินค้าจริงต่อไป เพื่อเป็นการลดต้นทุนที่จะใช้ในการผลิตสินค้าได้ การออกแบบผลิตภัณฑ์ หมายถึง การถ่ายทอดรูปแบบจากความคิดออกมาเป็นผลงานที่ผู้อื่นสามารถมองเห็นรับรู้ หรือสัมผัสได้ เพื่อให้มีความเข้าใจในผลิตภัณฑ์ร่วมกัน และการพัฒนาผลิตภัณฑ์หมายถึง การพัฒนา ปรับปรุง หรือเปลี่ยนแปลงผลิตภัณฑ์เดิมให้ตรงตามความต้องการของผู้บริโภคให้ได้มากที่สุด การออกแบบและพัฒนาผลิตภัณฑ์เป็นหัวใจสำคัญในการแข่งขันเชิงการตลาด มีธุรกิจมากมายที่ประสบความสำเร็จ เพราะผู้ประกอบการเห็นความสำคัญ และลงทุนกับกระบวนการออกแบบและพัฒนาผลิตภัณฑ์ ซึ่งเป็นการลงทุนสูงและผู้ประกอบการต้องทำความเข้าใจและร่วมปฏิบัติตามขั้นตอนจนสามารถดำเนินการด้วยตนเองได้ในที่สุด จึงจะถือว่าประสบความสำเร็จ และได้ผลคุ้มค่ากับการลงทุน การออกแบบและพัฒนาผลิตภัณฑ์มีหลักการ หรือขั้นตอนหลักคือ เริ่มต้นจากการการสำรวจข้อมูลความต้องการของลูกค้าประกอบการออกแบบ แล้วจึงทำการวิเคราะห์ข้อมูลเพื่อทำการออกแบบผลิตภัณฑ์หรือปรับปรุงผลิตภัณฑ์ ทำการการสร้างต้นแบบชิ้นงาน ทำการทดสอบและประเมินผลต้นแบบชิ้นงาน เมื่อชิ้นงานต้นแบบมีลักษณะตามที่ต้องการแล้วจึงทำการผลิตสินค้าเพื่อนำออกสู่ตลาดต่อไป ดังรูปที่ 1.1 ซึ่งแสดงกระบวนการออกแบบและการพัฒนาผลิตภัณฑ์

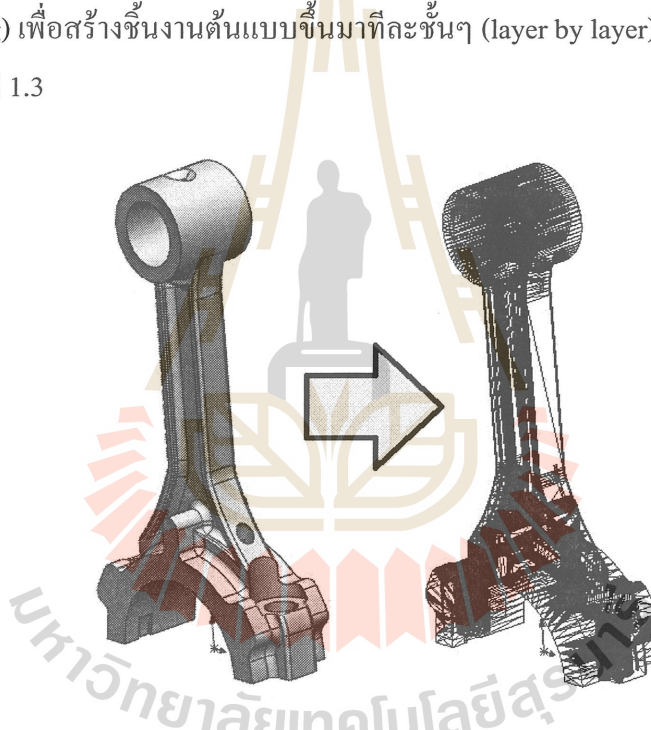
การออกแบบและพัฒนาผลิตภัณฑ์ต้องคำนึงถึงข้อกำหนดและปัจจัยหลายๆ อย่าง ได้แก่ รูปร่าง รูปทรง การเคลื่อนที่ การรับแรง วัสดุ ความปลอดภัย หลักการยศาสตร์ การผลิต การควบคุมคุณภาพ การประกอบ การขนส่ง การบำรุงรักษา ต้นทุน และเวลาที่ใช้อีกด้วย (Kevin N. Otto & Kristin L. Wood, 2001)



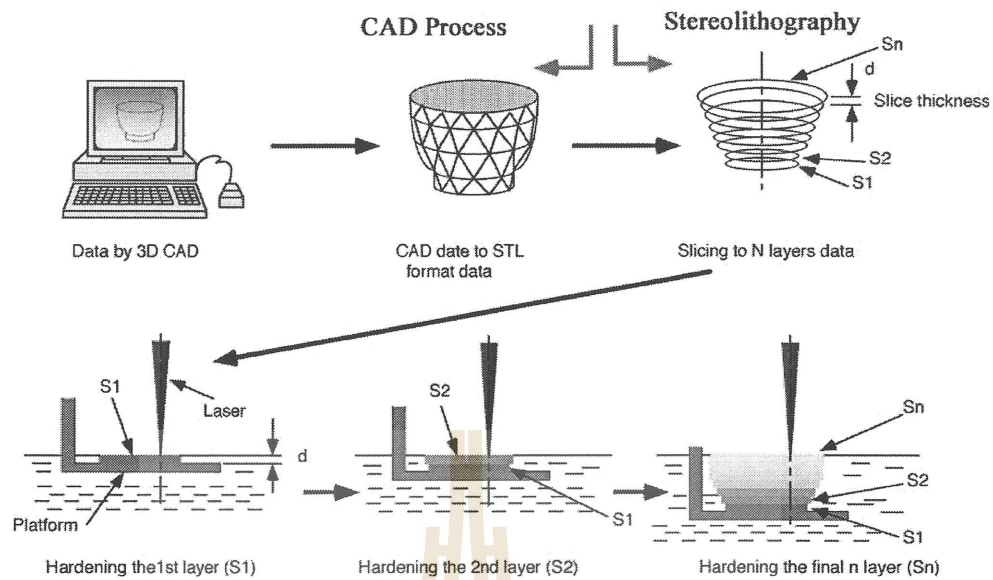
รูปที่ 1.1 กระบวนการออกแบบและพัฒนาผลิตภัณฑ์

ในการสร้างต้นแบบชิ้นงานนั้น แต่เดิมอาจใช้การสร้างจากการแกะสลักไม้ การสร้างแม่พิมพ์ หรือการตัดโฟม เป็นต้น ซึ่งการสร้างต้นแบบชิ้นงานแบบนี้จะใช้เวลาค่อนข้างมากในการสร้างแม่พิมพ์ รวมถึงหากต้นแบบมีการปรับปรุงแก้ไขจะต้องสร้างแม่พิมพ์ขึ้นมาใหม่ ทำให้เสียทั้ง

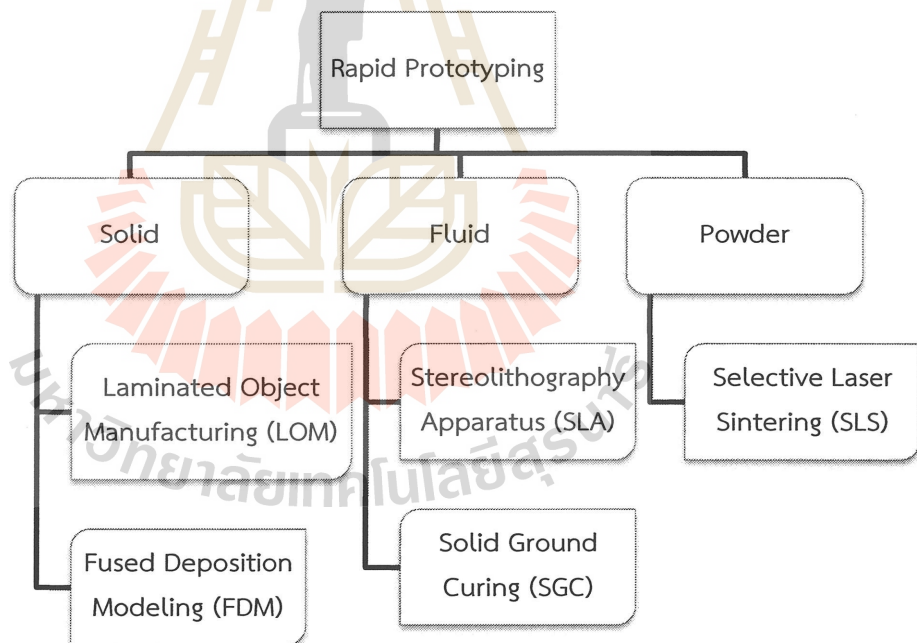
เวลาและต้นทุนในการสร้างต้นแบบมากขึ้น รวมถึงยังไม่มีการนำโปรแกรมทางคอมพิวเตอร์ (Computer Aided Design: CAD) เข้ามาช่วยในการออกแบบชิ้นงาน เนื่องจากปัญหาในการสร้างต้นแบบชิ้นงานแบบเก่าจึงได้มีการนำเทคโนโลยีใหม่ที่เข้ามาช่วยให้การสร้างต้นแบบชิ้นงานรวดเร็ว รวมถึงใช้ต้นทุนในการผลิตต่ำอีกด้วย เรียกวิธีการนี้ว่า การสร้างชิ้นงานต้นแบบรวดเร็ว (Rapid Prototype : RP) เป็นการสร้างต้นแบบชิ้นงานโดยอาศัยการขึ้นรูปจากไฟล์ CAD หรือ STL (standard tessellation language) ไฟล์ดังแสดงในรูปที่ 1.2 ซึ่งจะทำให้การสร้างต้นแบบชิ้นงานมีความรวดเร็ว และใช้ต้นทุนต่ำ รวมถึงสามารถทำการแก้ไขต้นแบบได้ง่ายมากยิ่งขึ้นอีกด้วย กระบวนการสร้างต้นแบบชิ้นงานด้วยเทคโนโลยี RP เริ่มจากการออกแบบชิ้นงานสามมิติ แล้วนำข้อมูลที่ได้ส่งไปยังเครื่อง RP เพื่อทำการแปลงข้อมูลจากไฟล์ CAD ให้เป็น STL ไฟล์ แล้วจึงทำการแบ่งชั้น (Slicing) เพื่อสร้างชิ้นงานต้นแบบขึ้นมาทีละชั้นๆ (layer by layer) จนได้ชิ้นงานต้นแบบที่สมบูรณ์ ดังรูปที่ 1.3



รูปที่ 1.2 แสดงการภาพตัวอย่างชิ้นงานที่เป็นไฟล์ CAD และ STL ไฟล์ ตามลำดับ
(ที่มา: <http://www.scielo.org>)



รูปที่ 1.3 กระบวนการสร้างชิ้นงานต้นแบบด้วยวิธีการ Rapid Prototyping
(ที่มา: <http://www.thagiwara.jp>)



รูปที่ 1.4 แสดงการแบ่งประเภทของ RP ตามวัสดุตั้งต้นที่ใช้ในกระบวนการ

Rapid Prototyping (RP) แต่ละประเภทจะมีลักษณะการทำงาน หรือกระบวนการ และวัสดุตั้งต้นที่ใช้สร้างชิ้นงานแตกต่างกันออกไปดังแสดงในรูปที่ 1.4 ได้แก่ กระบวนการ Stereolithography (SLA) ซึ่งถือว่าเป็นเทคโนโลยี rapid prototype แรกที่คิดค้นขึ้นมา เป็นการขึ้นรูป

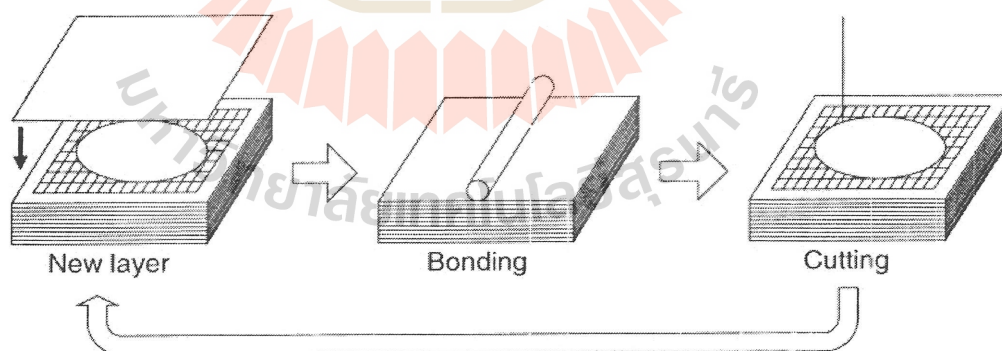
ขึ้นงานโดยการยิงแสงเลเซอร์ไปบนของเหลว เพื่อให้ของเหลวในแต่ละชั้นแข็งตัว โดยวัสดุของเหลวเป็นแบบ liquid photopolymer (resin) ขึ้นงานที่ได้จะมีความละเอียดและเที่ยงตรงสูง แต่ราคาเครื่องจักรและค่าใช้จ่ายในการดูแลรักษาก็สูงเช่นกัน กระบวนการ Fused Deposition Modeling (FDM) เป็นวิธีที่ใช้แพร่หลายที่สุด ใช้วัสดุหลักเป็นเส้นพลาสติกหรือโลหะที่พันเป็นม้วนส่งผ่านไปยังหัวอัดรีดซึ่งจะหลอมให้พลาสติกละลายก่อนนำมาพ่นลงบนแท่นวาง ซึ่งเมื่อพ่นออกมาแล้วจะแข็งตัวอย่างรวดเร็ว เครื่องจะพ่นทีละชั้นกระทั่งกลายเป็นโมเดล 3 มิติตามรูปทรงที่ต้องการ วัสดุที่ใช้ในการขึ้นรูปด้วยวิธีนี้ได้แก่ เทอร์โมพลาสติก เช่น ABS, โพลีคาร์บอเนต และโพลิฟีนิลซัลโฟนาพวกอีลาสโตเมอร์ กระบวนการ Selective Laser Sintering (SLS) คือ การหลอมผงวัสดุที่ใช้ด้วยแสงเลเซอร์ ใช้หลักการเดียวกันกับ SLA แต่ทำให้สามารถใช้วัสดุที่มีความแข็งแรงมากกว่า เช่น Thermoplastic, Nylon, Polyamide และ Polystyrene โดยวัสดุที่ใช้จะอยู่ในรูปผง เช่น โลหะ พลาสติก เซรามิก และแก้ว เครื่องจะใช้เลเซอร์วาดรูปแบบขึ้นทีละชั้นและพ่นผงเหล่านีกลงไปหลอมละลายกระทั่งกลายเป็นชิ้นงาน วิธีนี้ใช้ในอุตสาหกรรมเท่านั้น กระบวนการ Solid Ground Curing (SGC) การสร้างชิ้นงานใช้หลักการคล้ายกับเครื่องถ่ายภาพเอกซเรย์คือ นำแผ่นกระจกใสมาประจุไฟฟ้าตามรูปภาคตัดขวางในส่วนที่ไม่ใช่เนื้อชิ้นงาน จากนั้นนำผงหมึกสีดำมาโรยให้ทั่วผงหมึกนี้จะยึดติดกับกระจกในส่วนที่มีประจุไฟฟ้า กระจกจะให้แสงผ่านได้ในส่วนที่ไม่มีผงหมึกซึ่งก็คือส่วนที่เป็นเนื้อชิ้นงาน ในส่วนที่มีผงหมึกติดอยู่จะทึบแสง จากนั้นนำแผ่นกระจกนี้มาวางเหนือผิวเรซินเหลวและฉายแสงอัลตราไวโอเลตเพื่อให้เรซินแข็งตัว เรซินเหลวที่เหลือจะถูกกำจัดออกแล้วลงแว็กซ์เหลวบนชิ้นงานและบริเวรรอบๆ จากนั้นทำการปาดผิวหน้าให้ชั้นของชิ้นงานได้ความเรียบและสร้างชั้นต่อไป เมื่อชิ้นงานเสร็จสมบูรณ์แล้ว นำมาล้างแว็กซ์ออกด้วยกรด citric ผสมน้ำที่อุณหภูมิสูงและ กระบวนการ Laminated Object Manufacturing (LOM) ใช้เลเซอร์หรือมีดตัดวัสดุที่ใช้สร้าง โดยวัสดุจะมีลักษณะเป็นแผ่นซึ่งมีสารยึดติดอยู่ด้านหนึ่ง เช่น กระดาษ พลาสติก เซรามิก ไวนิล เป็นต้น ตัดวัสดุตามเส้นรอบรูปของชิ้นงานทีละชั้น ส่วนที่ไม่ใช่เนื้อชิ้นงานจะถูกตัดเป็นลูกบาศก์เล็กๆ เพื่อง่ายต่อการแกะออกจากชิ้นงาน หลังการจากตัดแต่ละชั้นจะมีลูกกลิ้งร้อน (heated bonding roller) กลิ้งทับด้วยความดันเพื่อให้สารยึดติดหลอมเหลวเพื่อยึดติดกับชิ้นงานชั้นด้านล่าง แท่นรองต้นแบบจะถูกเลื่อนต่ำลงเท่ากับขนาดของแต่ละชั้น และวัสดุแผ่นใหม่จะถูกดึงเข้ามาแทนที่และพร้อมที่จะสร้างชิ้นงานชั้นต่อไป (เทคโนโลยีการพิมพ์แบบสามมิติ 3D printing, 2013) สามารถแบ่งประเภทของ RP ตามลักษณะวัสดุตั้งต้นคือ ของแข็ง ของเหลว และผง

การศึกษาเกี่ยวกับการพัฒนากระบวนการ RP จะมีแนวทางการพัฒนา 4 ด้าน คือ การออกแบบและขึ้นรูปชิ้นงานต้นแบบจาก STL ไฟล์ (STL File Modeling), การวิเคราะห์การแบ่งชิ้นงาน (Slicing Analysis), การวิเคราะห์แนวทางการเคลื่อนที่ของเครื่องมือสร้างต้นแบบ

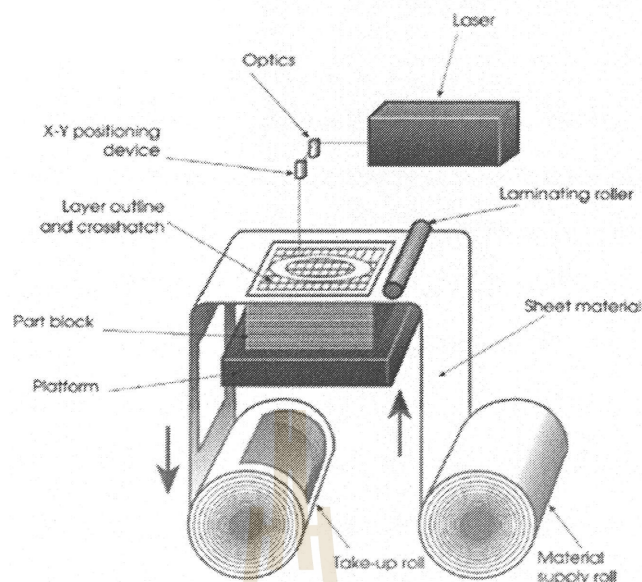
(Tool path planning of RP process) และการวิเคราะห์ทิศทาง การจัดวางวัตถุ (Object Analysis) (ปภากร พิทยชาวล และคณะ, 2556)

กระบวนการสร้างต้นแบบรวดเร็วในปัจจุบันมีแนวโน้มที่จะถูกนำมาใช้อย่างสูงขึ้นในอนาคต โดยจะเห็นได้จากการพัฒนากระบวนการเกิดขึ้นอย่างรวดเร็วและต่อเนื่องในช่วง 2-3 ปีที่ผ่านมา โดยกระบวนการสร้างต้นแบบรวดเร็วมีการเติบโตและนำไปประยุกต์ใช้ได้หลากหลายขึ้น ไม่ว่าจะเป็นการนำไปใช้ในการแพทย์, แฟชั่นดีไซน์, รถยนต์, ที่อยู่อาศัย, อุปกรณ์อิเล็กทรอนิกส์, อาหาร และอื่นๆ (ชัยวัฒน์ พุดพิงศ์พิบูลย์, 2558)

ในการขึ้นรูปชิ้นงานต้นแบบด้วยกระบวนการ RP แต่ละประเภทนั้น จะมีข้อดีและข้อเสียที่แตกต่างกัน ในงานวิจัยนี้สนใจศึกษากระบวนการสร้างต้นแบบชิ้นงานแบบ Laminate object manufacturing (LOM) โดยกระบวนการนี้มีข้อดี เช่น ถึงแม้ว่าชิ้นงานจะมีขนาดเล็กหรือขนาดใหญ่ จะไม่ต้องการเพิ่มขั้นตอนในการสร้างชิ้นงาน เป็นกระบวนการที่มีต้นทุนต่ำ เพราะวัสดุตั้งต้นที่นำมาใช้มีราคาไม่สูง เช่น กระดาษ หรือโฟม เป็นต้น การขึ้นรูปชิ้นงานด้วยกระบวนการ RP ประเภท SLA หรือ FDM จำเป็นที่จะต้องสร้างฐานรองรับชิ้นงานขึ้นมาเพื่อป้องกันไม่ให้ชิ้นงานเกิดการเสียรูป หรือโก่งงอในระหว่างการสร้างชิ้นงานที่มีจำนวนชั้นของชิ้นงานเพิ่มขึ้น แต่การสร้างชิ้นงานแบบ LOM ไม่ต้องการฐานรองรับชิ้นงานเพราะวัสดุที่ใช้สร้างชิ้นงานมีลักษณะเป็นแผ่นจึงมีทั้งส่วนที่เป็นชิ้นงานและส่วนที่ไม่ใช่ชิ้นงานที่คอยรองรับตัวชิ้นงานอยู่แล้ว ชิ้นงานจึงไม่เกิดการเสียรูปหรือโก่งงอขึ้น (I. Cho et al., 2000) ดังจะเห็นได้จากรูปที่ 1.5 และรูปที่ 1.6

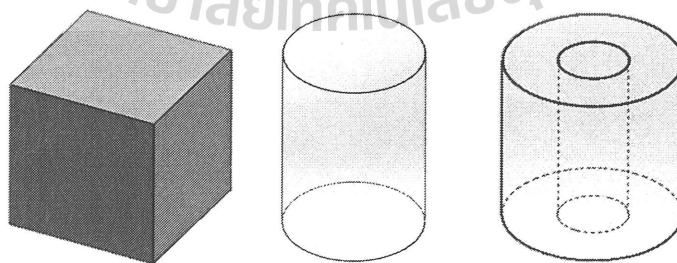


รูปที่ 1.5 แผนผังกระบวนการของ LOM (I. Cho et al., 2000)

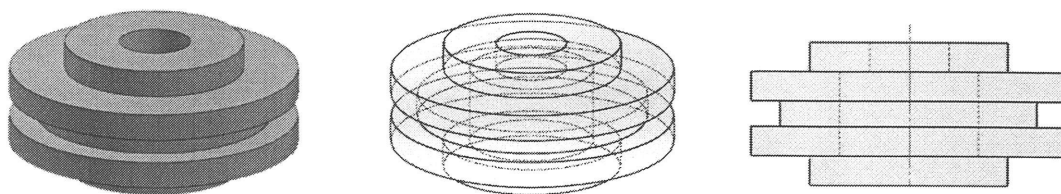


รูปที่ 1.6 แสดงกระบวนการ Laminate Object Manufacturing
(ที่มา: courtesy Helisys, Inc., <http://www.helisys.com>)

ในการสร้างต้นแบบชิ้นงานด้วยวิธีการ LOM นั้น สามารถสร้างชิ้นงานได้หลายรูปแบบ เช่น รูปทรงเรขาคณิตต่างๆ ทรงตัน ทรงกระบอก หรือทรงกลวง เป็นต้น ดังรูปที่ 1.7 หากชิ้นงานมีความซับซ้อนมากขึ้น ดังตัวอย่างที่แสดงในรูปที่ 1.8 จะต้องมีการวางแผนเกี่ยวกับการกำหนดตำแหน่งและการวิเคราะห์แนวทางการเคลื่อนที่ของเครื่องมือสร้างต้นแบบ (Tool path planning of RP process) ของการขึ้นรูปชิ้นงานที่มีลักษณะเป็นทรงดังกล่าว และต้องทราบส่วนประกอบของชิ้นงานที่จะสร้างว่าส่วนใดคือส่วนที่เป็นเนื้อชิ้นงาน และส่วนที่ไม่ใช่ส่วนของเนื้อชิ้นงานที่จะต้องทำการตัดออก



รูปที่ 1.7 แสดงชิ้นงานรูปทรงต่างๆ

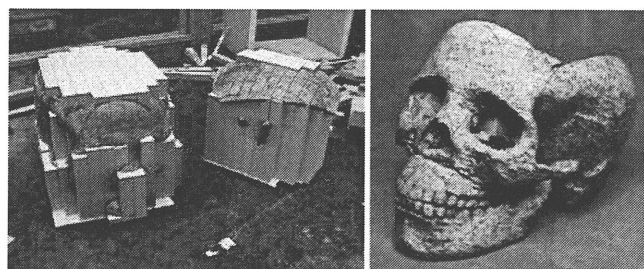


รูปที่ 1.8 แสดงชิ้นงานที่มีลักษณะซับซ้อน

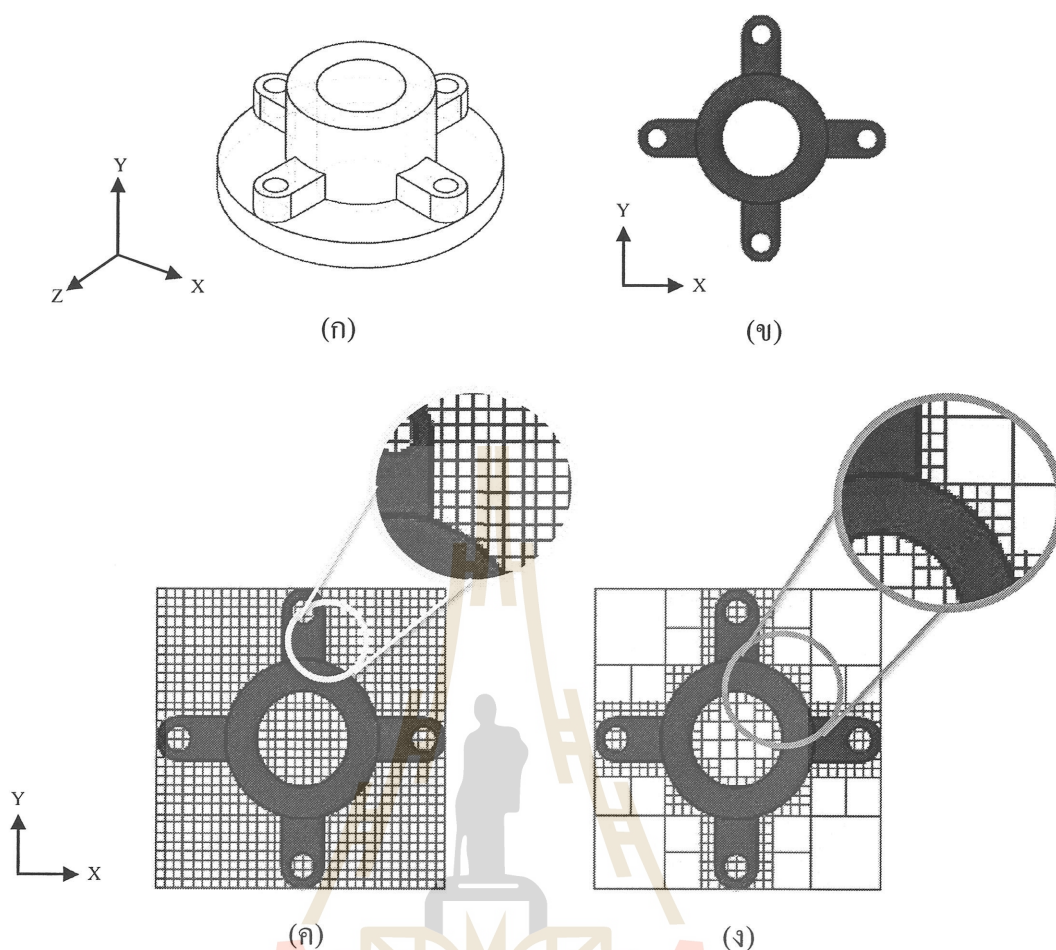
ในการขึ้นรูปชิ้นงานโดยกระบวนการ LOM นั้น มีนักวิจัยได้เสนอแนวทางการตัดที่เรียกว่า “Bond then cut” (Y.Y. Chiu et al., 2015) ซึ่งเป็นการขึ้นรูปชิ้นงานแบบเชื่อมติดแผ่นวัสดุที่จะใช้สร้างชิ้นงานทั้งแผ่นเข้าด้วยกันแล้วจึงทำการตัดส่วนของชิ้นงานตามแบบ ซึ่งในทุกชั้นของชิ้นงานก็จะประกอบไปด้วยส่วนที่เป็นชิ้นงานและส่วนที่ไม่ใช่ชิ้นงาน ดังนั้นเมื่อเสร็จสิ้นกระบวนการขึ้นรูปชิ้นงานต้นแบบแล้วคนงานจะต้องทำการตัดส่วนที่ไม่ใช่ชิ้นงานออกเป็นลูกบาศก์ ซึ่งขั้นตอนนี้เรียกว่า De-cubing ดังจะเห็นได้จากรูปที่ 1.9 และ 1.10 โดยกระบวนการ de-cubing จะแบ่งเป็น 2 ประเภทคือ Uniform de-cubing และ Adaptive de-cubing



รูปที่ 1.9 แสดงกระบวนการ de-cubing (ที่มา: courtesy Helisys, Inc., <http://www.helisys.com>)



รูปที่ 1.10 แสดงชิ้นงานก่อนการทำ De-cubing (ซ้าย) และ ชิ้นงานหลังการทำ De-cubing (ขวา)
(ที่มา: <https://www.researchgate.net>)



รูปที่ 1.11 (ก) ชิ้นงานสามมิติ (ข) ภาพหน้าตัดชิ้นงาน (ค) Uniform de-cubing
(ง) Adaptive de-cubing

โดยลักษณะของ Uniform de-cubing จะเป็นการตัดส่วนที่ไม่ใช่เนื้อชิ้นงานออกในลักษณะ ลูกบาศก์เท่าๆกันทั้งหมด ดังแสดงในรูปที่ 1.13 และ Adaptive de-cubing จะเป็นการตัดส่วนที่ไม่ใช่เนื้อชิ้นงานออกเป็นช่องสี่เหลี่ยมที่มีขนาดเล็กหรือใหญ่ไม่เท่ากัน ดังแสดงในรูปที่ 1.14 ซึ่งในกระบวนการ de-cubing นี้ไม่เพียงแต่จะทำให้เสียทั้งกำลังคนซึ่งต้องอาศัยคนงานที่มีทักษะพอสมควรและทำให้เสียชั่วโมงการทำงานของคนงานที่ค่อนข้างมากแล้ว ยังสร้างความเสียหายกับบริเวณผิวของชิ้นงานอีกด้วย ทำให้ไม่สามารถใช้กับชิ้นงานที่มีรูปทรงค่อนข้างซับซ้อนได้ เช่น ชิ้นงานที่มีรูปร่างกลวง หรือชิ้นงานที่มีรูปทรงแฉก เป็นต้น ดังนั้นเพื่อแก้ปัญหา de-cubing ต่อมา จึงได้มีการคิดค้นวิธีการเพื่อมาแก้ปัญหาดังกล่าวคือวิธีการที่เรียกว่า “cut then bond” (Y.Y. Chiu et al., 2015) คือการตัดชิ้นงานตามที่ได้แบ่งชั้นชิ้นงานในไฟล์ STL ทีละชั้น ทั้งที่เป็นส่วนของชิ้นงาน และส่วนที่ไม่ใช่ส่วนของชิ้นงาน ซึ่งต้องทำการ de-cubing ออก แล้วจึงติดชิ้นงานเข้าด้วยกันทีละชั้น จะทำให้วัสดุส่วนที่ไม่ใช่ชิ้นงานทั้งหมดถูกตัดทุกชั้นของชิ้นงานซึ่งง่ายต่อการแยกออกจาก

ส่วนที่เป็นชิ้นงานเมื่อการขึ้นรูปชิ้นงานเสร็จสมบูรณ์ แต่วิธีการนี้ยังทำให้เกิดปัญหา คืออาจเกิดความคลาดเคลื่อนระหว่างที่ติดชิ้นงานซึ่งจะทำให้ชิ้นงานที่ได้ไม่ตรงตามต้องการ และบริเวณขอบของชิ้นงานยังอาจเกิดความเสียหายระหว่างกระบวนการ de-cubing ซึ่งปัญหานี้ถือว่าเป็นปัญหาที่มีความสำคัญในการสร้างชิ้นงานต้นแบบ

งานวิจัยนี้นำเสนอแนวทางการวิเคราะห์เพื่อกำหนดตำแหน่งการตัดของเครื่องมือตัดบริเวณส่วนที่ไม่ใช่ชิ้นงาน สำหรับกระบวนการ De-cubing ในการขึ้นรูปชิ้นงานต้นแบบรวดเร็วด้วยเทคโนโลยี Laminate Object Manufacturing (LOM) โดยศึกษากระบวนการสร้างชิ้นงานต้นแบบด้วยวิธีการ “cut then bond” และศึกษาเกี่ยวกับการแบ่งพื้นที่ในการตัดส่วนที่ไม่ใช่เนื้อชิ้นงานออกเป็นรูปทรงสี่เหลี่ยมที่มีขนาดไม่เท่ากันแบบ Adaptive de-cubing ของการขึ้นรูปในแต่ละชั้น เพื่อให้ความเสียหายระหว่างการแยกส่วนที่ไม่ใช่ชิ้นงานออกจากส่วนของชิ้นงาน ที่อาจเกิดขึ้นบริเวณขอบของชิ้นงาน (contour) ลดลง และวางแผนการเคลื่อนที่ของเลเซอร์ หรือเครื่องมือตัด เป็นการหลีกเลี่ยงการชนกันระหว่างเครื่องมือกับชิ้นงานต้นแบบ ส่งผลให้สามารถสร้างชิ้นงานออกมาได้ตรงตามแบบมากยิ่งขึ้น ใช้เวลาในการสร้างต้นแบบรวดเร็วมากยิ่งขึ้น และชิ้นงานไม่เกิดการเสียหายหรือเกิดการเสียหายน้อยลง เพื่อจะสามารถทำให้เราสร้างต้นแบบชิ้นงานออกมาได้ดี ตรงตามความต้องการ ประหยัดเวลาในการสร้างต้นแบบและต้นทุนในการผลิต ส่งผลให้เกิดประสิทธิภาพในการสร้างต้นแบบชิ้นงานที่สูงขึ้นได้

1.2 วัตถุประสงค์ของงานวิจัย

เพื่อกำหนดตำแหน่งการตัดส่วนที่ไม่ใช่ชิ้นงาน สำหรับกระบวนการ De-cubing ในกระบวนการขึ้นรูปชิ้นงานต้นแบบรวดเร็วด้วยเทคโนโลยี Laminate Object Manufacturing (LOM)

1.3 สมมติฐานการวิจัย

สมมติฐานการวิจัยคือ รูปภาพหน้าตัดชิ้นงานที่เป็นข้อมูลนำเข้า (In put) เป็นข้อมูลเพื่อการวางแผนการเคลื่อนที่ของเครื่องมือตัดในการสร้างชิ้นงานต้นแบบด้วยเทคโนโลยี Laminate Object Manufacturing (LOM)

1.4 ขอบเขตการวิจัย

1.4.1 การวิจัยนี้ศึกษาการขึ้นรูปชิ้นงานต้นแบบ โดยอ้างอิงวิธีการจากเทคโนโลยี Laminated Object Manufacturing (LOM)

1.4.2 พัฒนาโปรแกรมทางคอมพิวเตอร์เพื่อคำนวณหาเส้นทางการตัดในบริเวณส่วนที่ไม่ใช่เนื้อของชิ้นงาน (De-cube) ที่ละชั้นของแบบจำลอง

1.4.3 การวิจัยนี้ศึกษาวิธีการแบ่งส่วนพื้นที่บริเวณส่วนที่ไม่ใช่เนื้อของชิ้นงาน เพื่อใช้ในการวางแผนการเคลื่อนที่ของเครื่องมือตัดในการสร้างต้นแบบด้วยเทคโนโลยี Laminated Object Manufacturing (LOM) โดยทำการศึกษาด้านแบบที่มีลักษณะเป็นรูปทรงเรขาคณิต หรือต้นแบบที่มีลักษณะกลวง

1.4.4 ข้อมูลนำเข้า (Input) สำหรับงานวิจัยนี้คือ เส้นโครงร่างของชิ้นงานแต่ละชั้น (Layer contour) ที่ใช้ในการสร้างชิ้นงานต้นแบบซึ่งอยู่ในรูปแบบของข้อมูลภาพ (Image)

1.4.5 ข้อมูลนำออก (Output) สำหรับงานวิจัยนี้คือ เส้นทางการเคลื่อนที่ของเครื่องมือตัดในบริเวณส่วนที่ไม่ใช่ส่วนของชิ้นงาน (De-cube) แบบ Adaptive de-cube

1.4.6 การวิจัยนี้ไม่คำนึงถึงการตัดในบริเวณที่เป็นส่วนของชิ้นงาน

1.4.7 การวิจัยนี้ไม่คำนึงถึงจำนวนของชั้น ความหนาของชั้น (Layer Thickness) และทิศทางการวางวัตถุ (Object Orientation)

1.4.8 การวิจัยนี้กำหนดให้แบบจำลองที่ศึกษาผ่านขั้นตอนการแบ่งชั้นแล้ว

1.4.9 การวิจัยนี้กำหนดให้มีการนำเข้าข้อมูลเส้นโครงร่างทีละชั้น (layer by layer)

1.4.10 งานวิจัยนี้กำหนดให้เครื่องมือตัดมีการเคลื่อนที่ในแนวแกน X และแกน Y

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ได้กระบวนการในการสร้างเส้นทางการเคลื่อนที่ของเครื่องมือตัด (tool path) ในกระบวนการ De-cubing ของการสร้างชิ้นงานต้นแบบด้วยเทคนิค Laminated Object Manufacturing (LOM)

1.5.2 ได้แนวทางการวิจัยทางด้านการสร้างและออกแบบชิ้นงานต้นแบบรวดเร็วด้วยเทคนิค Laminated Object Manufacturing (LOM)

1.6 ส่วนประกอบของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ประกอบด้วยเนื้อหาทั้งหมด 5 บท บทที่ 1 กล่าวถึงที่มาและความสำคัญของปัญหา, วัตถุประสงค์, ของเขตของงานวิจัย และประโยชน์ที่คาดว่าจะได้รับ บทที่ 2 กล่าวถึงหลักการ, ทฤษฎีต่างๆ และทบทวนวรรณกรรมที่เกี่ยวข้องกับการสร้างต้นแบบรวดเร็ว (Rapid Prototype) และเทคโนโลยี Laminate Object Manufacturing บทที่ 3 กล่าวถึงรายละเอียดขั้นตอนในการดำเนินงานวิจัยทั้งหมด บทที่ 4 แสดงผลลัพธ์ของการดำเนินงานวิจัย และกรณีศึกษา และบทที่ 5 กล่าวถึงผลสรุปของการดำเนินงานวิจัย และข้อเสนอแนะในการนำงานวิจัยไปประยุกต์ใช้ในอนาคต



บทที่ 2

ทบทวนวรรณกรรม

เนื้อหาในบทนี้กล่าวถึงทฤษฎีที่เกี่ยวข้องกับการสร้างชิ้นงานต้นแบบรวดเร็ว และการสร้างต้นแบบรวดเร็วด้วยเทคโนโลยี Laminated Object Manufacturing (LOM) ได้แก่ ทฤษฎีการสร้างชิ้นงานต้นแบบรวดเร็ว, ทฤษฎีการสร้างต้นแบบรวดเร็วด้วยเทคโนโลยี Laminated Object Manufacturing (LOM), ทฤษฎีการประมวลผลภาพ (Image processing) ในการทบทวนวรรณกรรมที่เกี่ยวข้องได้กล่าวถึง การสร้างชิ้นงานต้นแบบรวดเร็ว และการสร้างต้นแบบรวดเร็วด้วยเทคโนโลยี Laminated Object Manufacturing (LOM) วิธีการและเทคนิคเกี่ยวกับแก้ไขปัญหาที่เกิดขึ้นในการสร้างชิ้นงานด้วยเทคโนโลยี LOM

2.1 กรรมวิธีการสร้างต้นแบบ (Prototype)

การสร้างต้นแบบ (Prototype) ถือเป็นหัวใจสำคัญในการผลิตชิ้นงาน สินค้าหรืออุปกรณ์ต่างๆ เพราะก่อนที่จะทำการผลิตชิ้นงานจริงเราต้องแน่ใจก่อนว่าชิ้นงานที่เราออกแบบหรือพัฒนาจะสามารถนำไปใช้ประโยชน์ได้จริงตรงตามความต้องการและการใช้งาน สามารถตอบสนองความต้องการของผู้ใช้งานได้อย่างสมบูรณ์ ผู้ผลิตจึงต้องทำการสร้างต้นแบบชิ้นงานขึ้นมาเพื่อศึกษาถึงคุณสมบัติ ฟังก์ชันการใช้งานลักษณะต่างๆ ความเหมาะสมในการใช้งาน รวมถึงความสวยงามภายนอกอีกด้วย ในการสร้างต้นแบบนี้จะมีการปรับเปลี่ยนต้นแบบหลายต่อหลายครั้งจึงจะได้ชิ้นงานที่สมบูรณ์

กรรมวิธีการสร้างต้นแบบโดยทั่วไปแบ่งได้เป็น 3 กรรมวิธีคือ กรรมวิธีหักออก (Subtractive process), กรรมวิธีเพิ่มเข้า (additive process) และกรรมวิธีอัดขึ้นรูป (Compressive process)

(1) กรรมวิธีหักออก (Subtractive process) เป็นกรรมวิธีที่ใช้กันมากและใช้กันมานาน การสร้างต้นแบบเริ่มจากการนำก้อนวัสดุมากัดเอาเนื้อวัสดุออกให้ได้รูปร่างตามที่ต้องการโดยใช้เครื่องมือทั่วไป เช่น เครื่องกลึง เครื่องกัด เครื่องเลื่อย เครื่องเจาะ เป็นต้น กรรมวิธีนี้ค่อนข้างใช้เวลานานในการสร้างต้นแบบ แต่ปัจจุบันได้มีเครื่อง CNC (Computer numerical control) ที่ช่วยในการทำงานให้เร็วขึ้นและแม่นยำขึ้น แต่เครื่อง CNC ยังมีข้อจำกัดคือ ไม่สามารถสร้างต้นแบบที่มีรูปร่างซับซ้อนมากๆ ได้ โดยเฉพาะเมื่อพื้นผิวที่ต้องการเอาออกมีส่วนอื่นบดบังอยู่

(2) กรรมวิธีเพิ่มเข้า (Additive process) เป็นกรรมวิธีที่สร้างต้นแบบโดยการเติมวัสดุลงตามพื้นที่ภาคตัดขวางในแนวนอนของต้นแบบทีละชั้น โดยเริ่มจากชั้นล่างสุด (พื้นที่ภาคตัดขวางด้านล่างสุดของต้นแบบ) ขึ้นมาถึงชั้นบนสุด (พื้นที่ภาคตัดขวางด้านบนสุดของต้นแบบ) ซึ่งโดยทั่วไปแล้วแต่ละชั้นจะมีความหนาประมาณ 0.1 มิลลิเมตร (ความหนาแต่ละชั้นขึ้นอยู่กับกรรมวิธีและวัสดุที่ใช้) กรรมวิธีนี้สามารถสร้างต้นแบบได้รวดเร็วและมีรูปร่างซับซ้อนได้ แต่มีข้อจำกัดเรื่องวัสดุที่ใช้สร้างต้นแบบ

(3) กรรมวิธีอัดขึ้นรูป (Compressive process) เป็นกรรมวิธีการสร้างต้นแบบโดยการหล่อการทาบ หรือปั๊มขึ้นรูปวัสดุโดยใช้แม่พิมพ์เป็นแบบ กรรมวิธีนี้ค่อนข้างใช้เวลามากเพราะต้องทำแม่พิมพ์และไม่สามารถสร้างต้นแบบที่มีรูปร่างซับซ้อนได้ (ฉัตรชัย จันทร์เด่นดวง, 2546)

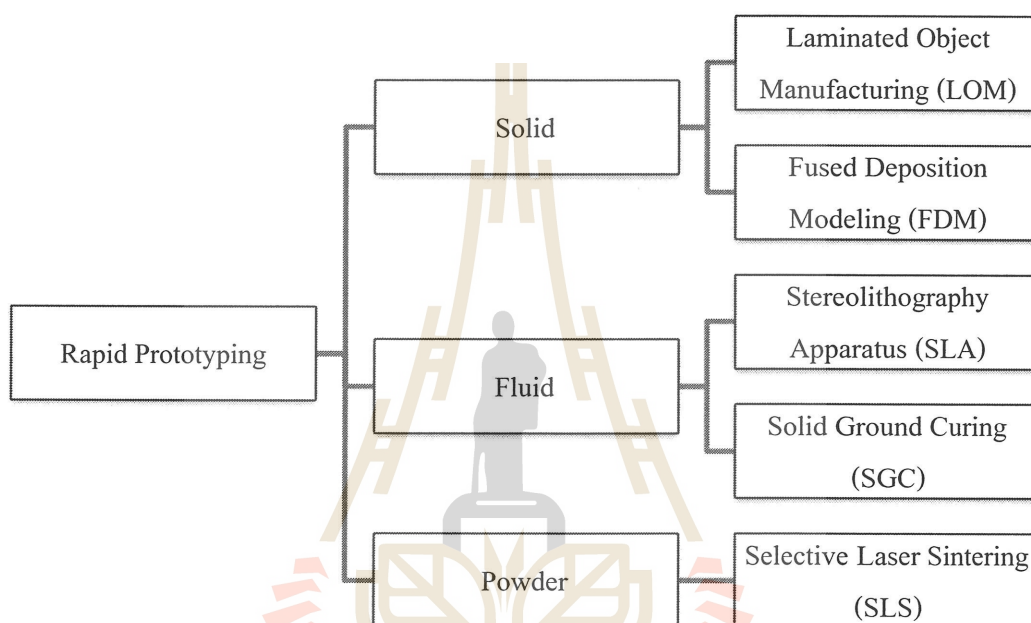
การสร้างต้นแบบขึ้นงานมีด้วยกันหลายวิธี โดยจะแตกต่างกันไปตามชนิด ขนาด และความต้องการอื่นๆ ของผลิตภัณฑ์นั้นๆ ต้นแบบมีทั้งแบบที่เท่าขนาดจริง และต้นแบบที่ถูกย่อ หรือขยายตามสัดส่วนที่ต้องการตามเงื่อนไขของการนำต้นแบบนั้นๆ ไปใช้ การสร้างต้นแบบที่ไม่ละเอียดหรือไม่ซับซ้อนมาก เห็นเพียงภาพรวม เช่นต้นแบบอาคารขนาดเล็ก จะใช้วัสดุที่นำมาสร้างต้นแบบที่หาได้ง่าย ขึ้นรูปง่าย และราคาถูก จะเรียกต้นแบบที่ทำออกมาเพื่อจุดประสงค์ในการสร้างต้นแบบแบบนี้จะเรียกว่า Conceptual Prototype ในทางกลับกันถ้าสร้างต้นแบบเพื่อให้เห็นถึงฟังก์ชันในการใช้งานที่เหมือนจริง รวมถึงขนาดและวัสดุที่ใช้เป็นวัสดุที่ใกล้เคียงหรือวัสดุเดียวกับชิ้นงานจริง เพื่อให้ผู้ผลิตได้สัมผัสถึงชิ้นงานจริงได้มากที่สุด เช่นการสร้างต้นแบบเครื่องมือ อุปกรณ์ทางการแพทย์ และชิ้นงานที่มีกลไกหรือชิ้นงานที่ต้องนำมาประกอบเข้าด้วยกันเป็นต้น การสร้างต้นแบบในลักษณะนี้จะเรียกว่า Functional Prototype (Charles L. Thomas, 1996)

2.2 การสร้างชิ้นงานต้นแบบรวดเร็ว (Rapid Prototyping: RP)

การสร้างต้นแบบอีกวิธีหนึ่งที่ได้ความนิยมน้อยแต่แพร่หลายในปัจจุบัน ไม่ว่าจะเป็นทางด้านอุตสาหกรรมการผลิตหรือทางการแพทย์ คือ การสร้างต้นแบบรวดเร็ว (Rapid Prototyping: RP) คือการสร้างต้นแบบจากแบบที่สร้างขึ้นมาจากโปรแกรมการออกแบบชิ้นงาน 3 มิติ (เช่น Auto CAD, Solid works) ขึ้นมาโดยตรง โดยข้อมูลด้านรูปทรงต่างๆจะถูกส่งไปยังเครื่อง RP เพื่อทำการสร้างชิ้นงานขึ้นมาทีละชั้น ข้อดีของเทคโนโลยี RP คือสามารถสร้างชิ้นงานที่มีรูปร่างแตกต่างกันได้อย่างไม่มีที่สิ้นสุด มีการขึ้นรูปอย่างรวดเร็ว สามารถแก้ไขปรับเปลี่ยนได้ง่ายเพราะสามารถปรับเปลี่ยนได้จากโปรแกรมได้แล้วสร้างใหม่ได้เลย สามารถลดเวลาในการสร้างต้นแบบลงได้ และสามารถเห็นภาพรวมของชิ้นงานได้ทั้งหมด ข้อเสียของกระบวนการ RP คือ สร้างชิ้นงานที่มีขนาดจำกัด ไม่สามารถสร้างชิ้นงานที่มีขนาดใหญ่ได้ มีความแข็งแรงของต้นแบบค่อนข้างน้อย และค่อนข้างจะมีข้อจำกัดของวัสดุที่นำมาใช้ (Ovidiu MOLDOVAN, 2011) วัสดุที่นำมาใช้กับ

เครื่อง RP สามารถแบ่งออกเป็น 3 ชนิด ได้แก่ ของแข็ง (Solid), ของเหลว (Liquid), และผง (Powder) วัสดุที่ใช้ในการสร้างต้นแบบอาจอยู่ในรูปแบบของ เม็ดกลมๆ, เม็ดพลาสติก, เส้น หรือ แผ่น ในปัจจุบันวัสดุที่นิยมใช้สร้างต้นแบบได้แก่ กระจก, ไนลอน, จีพิ้ง, เรซิน, โลหะ และเซรามิก (Ludmila Novakova-Marcincinova and Ivan Kuric, 2012)

Rapid Prototyping (RP) แต่ละประเภทจะมีลักษณะการทำงาน กระบวนการและวัสดุตั้งต้นที่ใช้ดังแสดงในรูปที่ 2.1

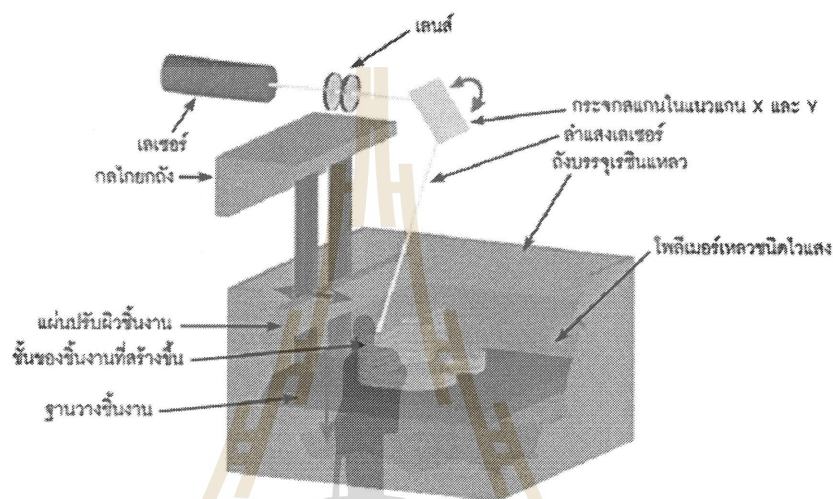


รูปที่ 2.1 การแบ่งประเภทของ RP ตามวัสดุตั้งต้นที่ใช้ในกระบวนการ

โดยในแต่ละกระบวนการจะมีแตกต่าง ข้อดี และข้อเสียที่ต่างกัน ได้แก่

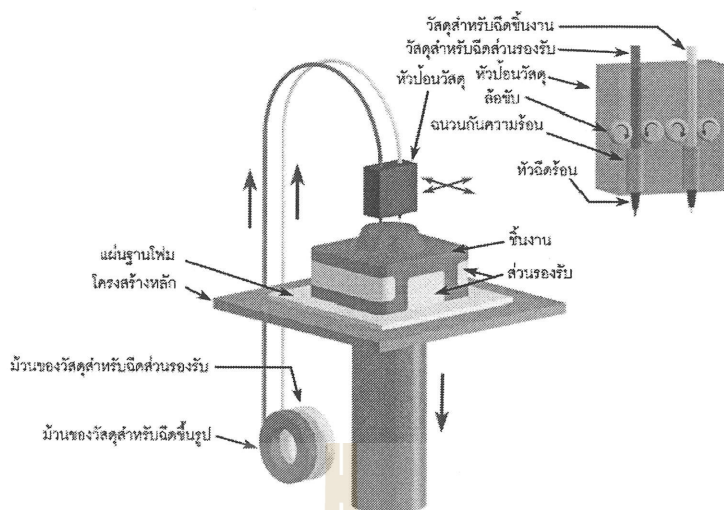
1. กระบวนการ Stereolithography (SLA) ดังรูปที่ 2.2 ซึ่งถือว่าเป็นเทคโนโลยี rapid prototype แรกที่เกิดขึ้นขึ้นมา โดยมีหลักการทำงานคือ ใช้เลเซอร์ยิงลงบนวัสดุของเหลวที่ไวต่อแสง liquid photopolymer (resin) เรซินที่ถูกแสงเลเซอร์จะแข็งตัว การสร้างต้นแบบด้วยวิธีนี้เริ่มจากการจุ่มแท่นรองต้นแบบ (platform) ลงในอ่างบรรจุเรซินเหลวโดยให้เรซินเหลวท่วมแท่นรองต้นแบบ สูงเท่ากับความหนาของภาคตัดขวางชิ้นแรกของชิ้นงาน จากนั้นยิงแสงเลเซอร์ตามขอบ (boundary) ของชิ้นงานก่อนและตามด้วยส่วนที่เป็นเนื้อชิ้นงาน เมื่อเรซินแข็งตัวแล้ว แท่นรองจะเลื่อนต่ำลงเท่ากับความหนาของชั้นต่อไป เพื่อให้เรซินท่วมชิ้นงานชั้นแรก และดำเนินการยิงเลเซอร์ ทำเช่นนี้ต่อไปเรื่อยๆจนได้ชิ้นงานต้นแบบที่เสร็จสมบูรณ์ วิธีการนี้เป็นการทำต้นแบบทีละชั้นจากข้างล่างขึ้นข้างบน เมื่อได้ชิ้นงานที่เสร็จสมบูรณ์แล้วจะต้องนำมาทำการบ่ม (post-curing) ด้วยแสงอัลตรา

ไวโอเลต เพื่อเพิ่มความแข็งแรง จากนั้นสามารถนำมาตกแต่งผิวด้วยวิธีการทั่วไป เช่น การพ่นทราย (sanding) การขัดเงา (polishing) เป็นต้น วิธีการนี้มีข้อดีคือ ให้ความละเอียดแม่นยำถึง 0.001 มิลลิเมตร ให้รายละเอียดและผิวเรียบดี และสามารถผลิตต้นแบบได้ทุกรูปร่าง ข้อเสียคือ ชิ้นงานค่อนข้างจะบิดงอง่าย เรซินมีความเป็นพิษ ต้นแบบที่ทำจากโพลิเมอร์บางชนิดจะบวมเมื่อโดนความชื้น ฐานรองรับชิ้นงานต้องถูกออกแบบและสร้างพร้อมกันกับต้นแบบ และราคาเครื่องจักรรวมทั้งค่าใช้จ่ายในการดูแลรักษามีราคาสูง



รูปที่ 2.2 กระบวนการขึ้นรูปชิ้นงานด้วยเทคนิค SLA
(ที่มา: <http://www.custompartnet.com>)

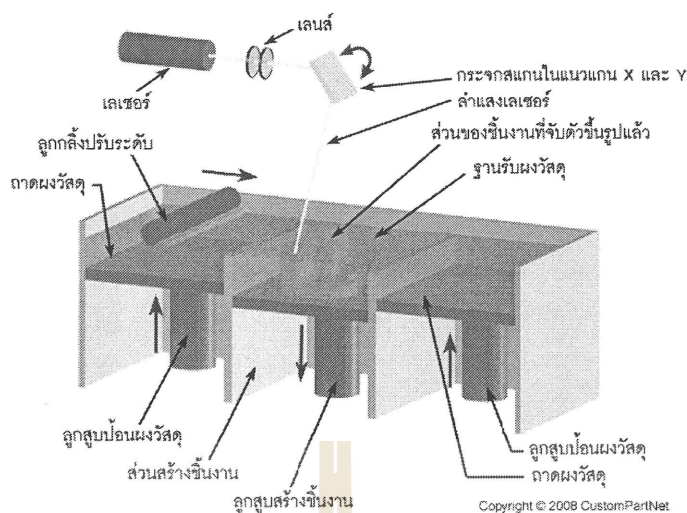
2. กระบวนการ Fused Deposition Modeling (FDM) ดังรูปที่ 2.3 ใช้วัสดุหลักเป็นเส้นพลาสติกหรือโลหะที่พันเป็นม้วน มีหลักการทำงานคือ เส้นวัสดุจะถูกดันไปยังหัวอัดรีด (extrusion nozzle) ซึ่งจะหลอมให้พลาสติกละลายก่อนนำมาพันลงบนแท่นวาง ซึ่งเมื่อพันออกมาแล้วจะแข็งตัวอย่างรวดเร็ว เครื่องจะพันทีละชั้นกระทั่งกลายเป็นโมเดล 3 มิติตามรูปทรงที่ต้องการ วัสดุที่ใช้ในการขึ้นรูปด้วยวิธีนี้ได้แก่ เทอร์โมพลาสติก เช่น ABS, โพลีคาร์บอเนต และโพลิฟีนิลซัลโฟน จำพวกอีลาสโตเมอร์ ข้อดีของวิธีนี้คือ สามารถใช้วัสดุได้หลากหลาย และสามารถเปลี่ยนวัสดุต่างชนิดระหว่างการผลิตได้ง่าย ไม่มีก๊าซพิษ เป็นวิธีที่ง่าย และประหยัด ไม่มีวัสดุของเสีย ไม่ต้องทำความสะอาดชิ้นงานเมื่อสร้างชิ้นงานเสร็จสมบูรณ์ และสามารถผลิตชิ้นงานที่มีผิวบางหรือรูกลวงได้เร็ว ข้อเสียคือ ต้องสร้างฐานรองรับชิ้นงาน ชิ้นงานไม่แข็งแรง และใช้เวลานานสำหรับชิ้นงานที่มีขนาดใหญ่



รูปที่ 2.3 กระบวนการขึ้นรูปชิ้นงานด้วยเทคนิค FDM

(ที่มา: <http://www.custompartnet.com>)

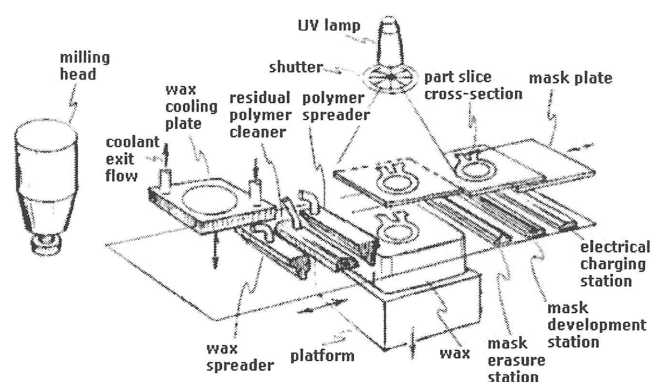
3. กระบวนการ Selective Laser Sintering (SLS) ดังรูปที่ 2.4 โดยมีหลักการทำงานคือ ใช้เลเซอร์หลอมวัสดุที่เป็นผงให้เป็นรูปร่างของต้นแบบ โดยวัสดุผงที่เก็บไว้ในกระบอบสูบจะถูกเพิ่มอุณหภูมิจนใกล้จุดหลอมเหลวแล้วถูกลูกกลิ้งดันมายังกระบอบสูบที่ใช้สร้างชิ้นงาน ซึ่งลูกกลิ้งจะเกลี่ยทับผงวัสดุเป็นชั้นบางๆ สม่ำเสมอ จากนั้นแสงเลเซอร์จะหลอมอนุภาคของผงวัสดุให้ติดกัน การสร้างชั้นต่อไปลูกสูบในกระบอบสูบที่เก็บผงวัสดุจะเลื่อนสูงขึ้นและลูกสูบในกระบอบสูบที่สร้างชิ้นงานจะเลื่อนต่ำลงเท่ากับความหนาของชิ้นงานในชั้นต่อไป การสร้างต้นแบบด้วยวิธีการนี้จะต้องทำในบรรยากาศก๊าซไนโตรเจน 98% เพื่อป้องกันผงวัสดุทำปฏิกิริยากับออกซิเจนขณะหลอม ความหนาของแต่ละชั้นจะขึ้นอยู่กับผงวัสดุที่ใช้ เช่น Thermoplastic, Nylon, Polyamide, Polystyrene, โลหะ, พลาสติก, เซรามิก และแก้ว ข้อดีของกระบวนการนี้คือ ไม่ต้องทำการบ่ม (post-curing) ไม่ต้องสร้างฐานรองรับชิ้นงาน สามารถใช้ผงเคลือบเซรามิกและโลหะ และชิ้นส่วนมีความแข็งแรง ข้อเสียคือ มีก๊าซพิษเกิดขึ้นในกระบวนการผลิต พื้นผิวเป็นรูพรุนไม่เรียบ กระบวนการผลิตใช้เวลามาก และวิธีนี้ใช้ในอุตสาหกรรมเท่านั้น



รูปที่ 2.4 กระบวนการขึ้นรูปชิ้นงานด้วยเทคนิค SLS

(ที่มา: <http://www.custompartnet.com>)

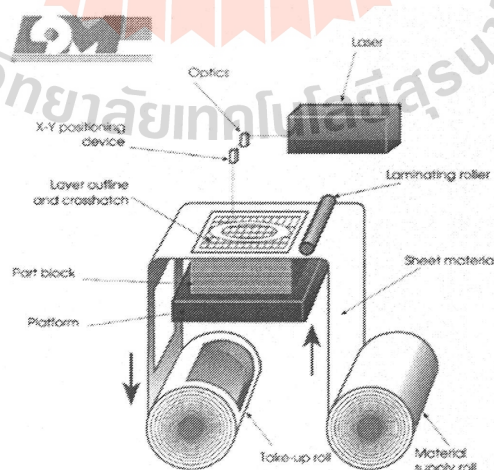
4. กระบวนการ Solid Ground Curing (SGC) ดังรูปที่ 2.5 หลักการทำงานคือ เริ่มจากไฟล์ STL ถูกแปลงเป็น CFL (Cubital Facet List) ซึ่งจะเก็บข้อมูลภาคตัดขวางแต่ละชั้นของต้นแบบ การสร้างชิ้นงานใช้หลักการคล้ายกับเครื่องถ่ายภาพเอกซเรย์คือ นำแผ่นกระจกใสมาประจุไฟฟ้าตามรูปภาคตัดขวางในส่วนที่ไม่ใช่เนื้อชิ้นงาน จากนั้นนำผงหมึกสีดำมาโรยให้ทั่วผงหมึกนี้จะยึดติดกับกระจกในส่วนที่มีประจุไฟฟ้า กระจกจะให้แสงผ่านได้ในส่วนที่ไม่มีผงหมึกซึ่งก็คือส่วนที่เป็นเนื้อชิ้นงาน ในส่วนที่มีผงหมึกติดอยู่จะทึบแสง จากนั้นนำแผ่นกระจกนี้มาวางเหนือผิวเรซินเหลวและฉายแสงอัลตราไวโอเลตเพื่อให้เรซินแข็งตัว เรซินเหลวที่เหลือจะถูกกำจัดออกแล้วลงแนวซ์เหลวบนชิ้นงานและบริเวณรอบๆ แล้วทำให้แนวซ์แข็งตัวด้วยแผ่นความเย็น จากนั้นทำการปาดผิวหน้าให้ชั้นของชิ้นงานมีความเรียบและความหนาตามต้องการเพื่อเตรียมผิวสำหรับการสร้างชั้นต่อไป เมื่อชิ้นงานเสร็จสมบูรณ์แล้ว นำมาล้างแนวซ์ออกด้วยกรด citric ผสมน้ำที่อุณหภูมิสูง กระบวนการนี้มีข้อดีคือ ผลิตได้สูง ไม่ต้องสร้างฐานรองรับชิ้นงาน และสามารถคาดการณ์เวลาที่ใช้ในการผลิตได้ ข้อเสียคือ ต้องคอยเฝ้าดูตลอดเวลาระหว่างการผลิต และสิ้นเปลืองวัสดุมาก



รูปที่ 2.5 กระบวนการขึ้นรูปชิ้นงานด้วยกระบวนการ SGC

(ที่มา: <http://www.bloggang.com>)

5. กระบวนการ Laminated Object Manufacturing (LOM) ดังรูปที่ 2.6 มีหลักการทำงานคือ ใช้เลเซอร์หรือมีดตัดวัสดุที่ใช้จะมีลักษณะเป็นแผ่นซึ่งมีสารยึดติดอยู่ด้านหนึ่ง เช่น กระดาษพลาสติก เซรามิกส์ ไวนิล เป็นต้น ตัดวัสดุตามเส้นรอบรูปของชิ้นงานทีละชั้น ส่วนที่ไม่ใช่เนื้อชิ้นงานจะถูกตัดเป็นลูกบาศก์เล็กๆ เพื่อง่ายต่อการแกะออกจากชิ้นงาน หลังจากการจากตัดแต่ละชั้นจะมีลูกกลิ้งร้อน (heated bonding roller) กลิ้งทับด้วยความดันเพื่อให้สารยึดติดหลอมเหลวเพื่อยึดติดกับชิ้นงานชั้นด้านล่าง แผ่นรองต้นแบบจะถูกเลื่อนต่ำลงเท่ากับความหนาของแต่ละชั้น และวัสดุแผ่นใหม่จะถูกดึงเข้ามาแทนที่และพร้อมที่จะสร้างชิ้นงานชั้นต่อไป ข้อดีของกระบวนการนี้คือ ไม่ต้องทำการบ่ม (post-curing) ไม่ต้องสร้างฐานรองชิ้นงาน เป็นวิธีที่ง่าย รวดเร็ว และประหยัด เหมาะสำหรับการสร้างชิ้นงานที่มีขนาดใหญ่และสามารถใช้วัสดุได้หลายชนิด ข้อเสียคือ ต้นแบบดูดซับความชื้น ผิวชิ้นงานไม่เรียบดีเมื่อเทียบกับวิธีอื่น มีข้อจำกัดในการสร้างต้นแบบที่ค่อนข้างซับซ้อน

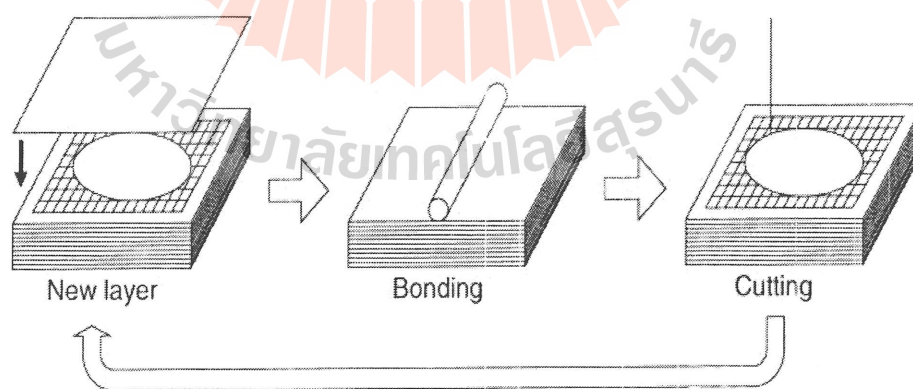


รูปที่ 2.6 แสดงกระบวนการ Laminated Object Manufacturing

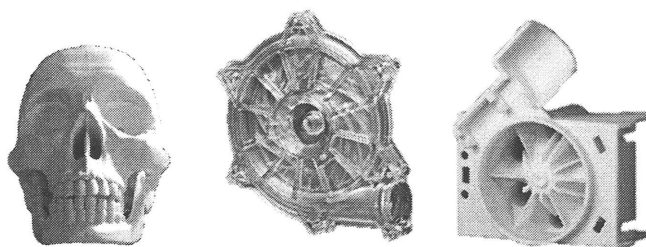
(ที่มา: courtesy Helisys, Inc., <http://www.helisys.com>)

กระบวนการสร้างต้นแบบรวดเร็วในปัจจุบันมีแนวโน้มที่จะถูกนำมาใช้อย่างสูงขึ้นในอนาคต โดยจะเห็นได้จากการพัฒนากระบวนการเกิดขึ้นอย่างรวดเร็วและต่อเนื่องในช่วง 2-3 ปีที่ผ่านมา โดยกระบวนการสร้างต้นแบบรวดเร็วมีการเติบโตและนำไปประยุกต์ใช้ได้หลากหลายขึ้น ไม่ว่าจะเป็นการนำไปใช้ในการแพทย์, แฟชั่นดีไซน์, รถยนต์, ที่อยู่อาศัย, อุปกรณ์อิเล็กทรอนิกส์, อาหาร และอื่นๆ (ชัยวัฒน์ พลพิพงษ์พิบูลย์, 2558)

ในการขึ้นรูปชิ้นงานต้นแบบด้วยกระบวนการ RP แต่ละประเภทนั้น จะมีข้อดีและข้อเสียที่แตกต่างกัน ในงานวิจัยนี้สนใจกระบวนการสร้างต้นแบบชิ้นงานแบบ Laminate object manufacturing (LOM) โดยกระบวนการนี้มีข้อดี เช่น ถึงแม้ว่าชิ้นงานจะมีขนาดเล็กหรือขนาดใหญ่ จะไม่ต้องการเพิ่มขั้นตอนในการสร้างชิ้นงาน เป็นกระบวนการที่มีต้นทุนต่ำ เพราะวัสดุตั้งต้นที่นำมาใช้มีราคาไม่สูง เช่น กระดาษ หรือ ใยแก้ว เป็นต้น ส่วนข้อด้อยของการขึ้นรูปชิ้นงานด้วยเทคโนโลยี LOM นั้นคือต้นแบบที่ได้นั้น ไม่แข็งแรงและได้มีการวิจัยในการนำวัสดุประเภทโลหะหรือเซรามิกเข้ามาใช้ในเทคโนโลยี LOM ซึ่งทำให้ต้นแบบที่ได้มีความแข็งแรงเพิ่มมากขึ้น แต่ก็ทำให้ขั้นตอนการทำ De-cube ยากขึ้นเช่นกัน [Long Peak Engineering Inc.,] ในการขึ้นรูปชิ้นงานด้วยกระบวนการ RP ประเภท SLA หรือ FDM จำเป็นที่จะต้องสร้างฐานรองรับชิ้นงานขึ้นมาเพื่อป้องกันไม่ให้ชิ้นงานเกิดการเสียรูป หรือโก่งงอในระหว่างการสร้างชิ้นงานที่มีจำนวนชั้นของชิ้นงานเพิ่มขึ้น แต่การสร้างชิ้นงานแบบ LOM ไม่ต้องการฐานรองรับชิ้นงานเพราะวัสดุที่ใช้สร้างชิ้นงานมีลักษณะเป็นแผ่นจึงมีทั้งส่วนที่เป็นชิ้นงานและส่วนที่ไม่ใช่ชิ้นงานที่คอยรองรับตัวชิ้นงานอยู่แล้ว ชิ้นงานจึงไม่เกิดการเสียรูปหรือโก่งงอขึ้น (I. Cho et al., 2000) ดังจะเห็นได้จากรูปที่ 2.7 และ 2.8



รูปที่ 2.7 แผนผังกระบวนการของ LOM (I. Cho et al., 2000)



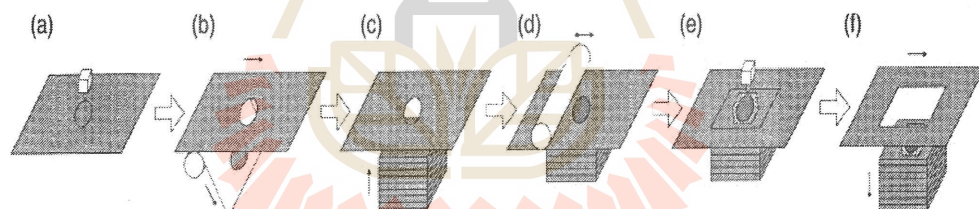
รูปที่ 2.8 แสดงตัวอย่างชิ้นงานต้นแบบที่สร้างด้วยวิธีการ Rapid Prototyping

การสร้างต้นแบบด้วยวิธี RP ได้นำไปใช้อย่างหลากหลายไม่ว่าจะเป็นทางด้านการแพทย์ เช่น การสร้างต้นแบบของใบหู โดยใช้ซีพียูเพื่อช่วยเด็กที่เกิดมาที่มีความผิดปกติที่เป็นโรคใบหูเล็กแต่กำเนิด (Microtia) หรือผู้ที่ประสบอุบัติเหตุสูญเสียใบหู (Majd Al Mardini et al., 2005) ช่วยสร้างโครงร่างเพื่อปลูกถ่ายอวัยวะของผู้ป่วย กระดูกนิ้ว ชี้นส่วนกะโหลก ขาเทียม ไปจนถึงการสร้างไตเทียมและหลอดเลือดเทียมซึ่งมีระบบโครงสร้างที่ไม่ซับซ้อนเท่าอวัยวะอื่น ทางด้านอุตสาหกรรม ได้มีการสร้างต้นแบบด้วยกระบวนการ RP ในการผลิตเครื่องบินบังคับขนาดเล็กที่สร้างเสร็จภายใน 7 วัน, รถยนต์ ECO CAR ที่วิ่งได้จริงบนถนน, สร้างแม่พิมพ์ในการหล่อเครื่องยนต์ชนิดดีเซลที่มีความซับซ้อนสูง (ZHANG Yu and LIU Hongwu, 2009), การหล่อชิ้นส่วนอุปกรณ์อิเล็กทรอนิกส์, เครื่องประดับเงินและทองคำจากไฟล์ดิจิทัล, รองเท้า, เสื้อผ้า ไปจนถึงการพิมพ์อะไหล่ปืนที่สามารถนำมาประกอบเพื่อยิงกระสุนจริงได้ และในอนาคต RP มีแนวโน้มที่จะได้รับความนิยมในการนำไปใช้อย่างแพร่หลายมากขึ้น

2.3 ทบทวนวรรณกรรม

ในการสร้างต้นแบบชิ้นงานนั้น แต่เดิมอาจใช้การสร้างจากการแกะสลักไม้ การสร้างแม่พิมพ์ หรือการตัดโฟม เป็นต้น ซึ่งการสร้างต้นแบบชิ้นงานแบบนี้จะใช้เวลาค่อนข้างมากในการสร้างแม่พิมพ์ รวมถึงหากต้นแบบมีการปรับปรุงแก้ไขจะต้องสร้างแม่พิมพ์ขึ้นมาใหม่ ทำให้เสียทั้งเวลาและต้นทุนในการสร้างต้นแบบมากขึ้น (Y.S. Liao, L.C. Chiu, Y.Y. Chiu, 2003) รวมถึงยังไม่ได้มีการนำโปรแกรมทางคอมพิวเตอร์ (Computer Aided Design: CAD) เข้ามาช่วยในการออกแบบชิ้นงาน เนื่องจากปัญหาในการสร้างต้นแบบชิ้นงานแบบเก่าจึงได้มีการนำเทคโนโลยีใหม่ที่เข้ามาช่วยให้การสร้างต้นแบบชิ้นงานรวดเร็ว รวมถึงใช้ต้นทุนในการผลิตต่ำอีกด้วย เรียกวิธีการนี้ว่า การสร้างชิ้นงานต้นแบบรวดเร็ว (Rapid Prototype : RP) เป็นการสร้างต้นแบบชิ้นงานโดยอาศัยการขึ้นรูปจากไฟล์ CAD หรือ STL ไฟล์ ซึ่งจะทำให้การสร้างต้นแบบชิ้นงานมีความรวดเร็ว และใช้ต้นทุนต่ำ รวมถึงสามารถทำการแก้ไขต้นแบบได้ง่ายมากยิ่งขึ้นอีกด้วย กระบวนการสร้างต้นแบบชิ้นงาน

ด้วยเทคโนโลยี RP เริ่มจากการออกแบบชิ้นงานสามมิติ แล้วนำข้อมูลที่ได้ส่งไปยังเครื่อง RP เพื่อทำการแปลงข้อมูลจากไฟล์ CAD ให้เป็น STL ไฟล์ แล้วจึงทำการแบ่งชั้น (Slicing) เพื่อสร้างชิ้นงานต้นแบบขึ้นมาทีละชั้นๆ (layer by layer) จนได้ชิ้นงานต้นแบบที่สมบูรณ์ ซึ่งในงานวิจัยนี้วิจัยนี้จะกล่าวถึงการสร้างชิ้นงานต้นแบบด้วยเทคโนโลยี Laminated Object Manufacturing (LOM) โดย Cho and Lee (ZHANG Yu and LIU Hongwu, 2009) ได้สนใจการสร้างชิ้นงานต้นแบบที่มีลักษณะกลวงหรือเป็นโพรง ซึ่งลักษณะชิ้นงานแบบนี้จะไม่สามารถนำส่วนที่ไม่ใช่ชิ้นงานออกมาได้ จึงเสนอกระบวนการใหม่ในการเอาแผ่นชิ้นงานออกของระบบ RP โดยมีหลักการคือ นำแผ่นวัสดุที่มีลักษณะเป็นกาวด้านหนึ่งเพื่อยึดติดชิ้นงานในแต่ละชั้น ทำการตัดชิ้นงานในชั้นที่ 1 โดยจะแบ่งการตัดชิ้นงานออกเป็น 2 ช่วง คือ ช่วงแรกทำการตัดชิ้นงาน 1 ชั้น ในขั้นตอนนี้ส่วนที่ไม่ใช่เนื้อชิ้นงานจะถูกตัดออกไปด้วย (de-cubing) แล้วจึงนำไปเชื่อมติดกับชิ้นงานชั้นอื่นๆ ที่เป็นฐานชิ้นงาน และทำการตัดชิ้นงานชั้นถัดไปเรื่อยๆ จนกระทั่งชิ้นงานเสร็จสมบูรณ์ ดังรูปที่ 2.9 โดยแบ่งเป็น 6 ขั้นตอน คือ (a) ทำการตัดครั้งแรกตามเส้น Contour ของชิ้นงาน (b) ลอกด้านที่มีกาวออก (c) ส่วนที่เป็นฐานชิ้นงานจะเลื่อนขึ้นมา (d) ให้ความร้อนเพื่อติดฐานชิ้นงานและส่วนที่ตัดครั้งแรก (e) ทำการตัดครั้งที่ 2 (f) ส่วนที่เป็นชิ้นงานจะเคลื่อนที่ลง (Cho and Lee, 2000)



รูปที่ 2.9 วิธีการตัดชิ้นงานแบบใหม่

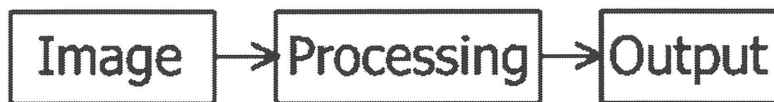
ในการศึกษาเกี่ยวกับการตัดส่วนที่ไม่ใช่ชิ้นงานออกเป็นลูกบาศก์ (de-cubing) แบ่งได้เป็น 2 แบบ คือ Uniform de-cubing และ Adaptive de-cubing มีลักษณะที่แตกต่างกันคือ Uniform de-cubing จะเป็นการตัดส่วนที่ไม่ใช่เนื้อชิ้นงานออกในลักษณะลูกบาศก์เท่าๆกันทั้งแผ่น ส่วน Adaptive de-cubing จะเป็นการตัดส่วนที่ไม่ใช่เนื้อชิ้นงานออกเป็นช่องสี่เหลี่ยมที่มีขนาดเล็กหรือใหญ่ไม่เท่ากัน Hur and Lee ได้ศึกษาเกี่ยวกับปัญหาที่พบจากการใช้วิธีการ de-cubing โดยแบ่งออกเป็นปัญหา 2 ข้อคือ (1) กระบวนการแบบตัดแล้วนำไปติด (Cut-then-bond) จะทำให้เกิดความคลาดเคลื่อนสะสมของชิ้นงานขึ้น และจะนำไปสู่ความผิดพลาดร้ายแรงของชิ้นงานที่ได้ เมื่อชั้นของชิ้นงานมีปริมาณมากขึ้น (2) ยังมีเนื้อวัสดุส่วนที่ไม่ใช่ชิ้นงานอยู่จำนวนมากที่ยังติดค้างอยู่กับบริเวณที่เป็นส่วนของชิ้นงานอีกด้วย ต่อมา ศรีนยา พุททวงศ์และคณะได้พัฒนาวิธีการและ

โปรแกรมสำหรับใช้สร้าง Adaptive de-cubing โดยทำการประมวลผลผ่านโปรแกรม Lab VIEW โดยเริ่มต้นโปรแกรมจากเมื่อชิ้นงาน 3 มิติ ผ่านกระบวนการแบ่งชิ้นงานแล้ว โดยทำการเรียกข้อมูลจากชั้นล่างสุดมาก่อน จากนั้นมีการสร้าง tiles ทั้งหมด 3 ขนาด โดยทำการสร้าง Uniform tiles (Main tiles: H_{SM}) ขึ้นมาก่อน โดยใช้อัตราส่วนระหว่างพื้นที่ของเส้น contour ที่อยู่นอกสุดและเส้นรอบรูปของเส้น contour ที่อยู่นอกสุด จากนั้นสร้างกล่องขอบเขตขึ้นมา หากพบว่าใน Main tiles ใดมีพิกเซลที่เป็นเส้น contour อยู่อย่างน้อย 1 ใน 4 ส่วน Main tiles นั้นจะถูกแทนที่ด้วย 4 Small tiles (H_{SS}) และ H_{SS} มีขนาดเท่ากับ $0.5 H_{SM}$ หากพบว่า Main tiles ไม่มีพิกเซลที่เป็นเส้น contour อยู่เลย Main tiles นั้นจะถูกแทนที่ด้วย Large tile (H_{SL}) และ H_{SL} มีขนาดเท่ากับ $2 H_{SM}$ เมื่อพิจารณาเสร็จสิ้นจะได้เส้นทางการทำ De-cubing สำหรับการสร้างชิ้นงานต้นแบบ จะเห็นได้ว่ากระบวนการทำ de-cubing ยังมีปัญหาที่ต้องได้รับการพัฒนาและแก้ไข เพื่อให้สามารถลดระยะเวลาในกระบวนการ De-cubing และเพื่อให้ชิ้นงานต้นแบบที่สร้างขึ้นมานั้นมีลักษณะใกล้เคียงกับที่ได้ออกแบบไว้มาก ในงานวิจัยนี้จึงต้องการวิเคราะห์แนวทางการเคลื่อนที่ของเครื่องมือสร้างต้นแบบ (Tool path planning of RP process) และศึกษาเกี่ยวกับการแบ่งพื้นที่ในการตัดส่วนที่ไม่ใช่เนื้อชิ้นงาน (de-cubing) ของการขึ้นรูปชิ้นงานต้นแบบทีละชั้น เพื่อให้ความเสียหายที่อาจเกิดขึ้นระหว่างผิวขอบของชิ้นงาน (contour) เมื่อมีการสร้างชิ้นงานต้นแบบด้วยกระบวนการ LOM ให้ลดลงเพื่อลดระยะเวลาในกระบวนการ De-cubing และสามารถวางแผนการเคลื่อนที่ของเลเซอร์ หรือมีดตัด เป็นการหลีกเลี่ยงการชนกันระหว่างเครื่องมือกับชิ้นงานต้นแบบและสามารถวางแผนการเคลื่อนที่ที่ให้ระยะทางที่สั้นที่สุด ส่งผลให้สามารถสร้างชิ้นงานออกมาได้ตรงตามแบบมากที่สุด รวดเร็วที่สุด และชิ้นงานเกิดการเสียหายน้อยลง ส่งผลให้ประหยัดเวลา ลดต้นทุนในการผลิต และเกิดประสิทธิภาพในการสร้างชิ้นงานต้นแบบที่สูงขึ้นได้

2.4 ทฤษฎี Image Processing

ทฤษฎี Image Processing คือการนำภาพมาประมวลผล เอมาคิดคำนวณทางคอมพิวเตอร์ การคิดคำนวณนั้นมีหลายวิธี ซึ่งแต่ละวิธีก็มีประโยชน์แตกต่างกัน ไม่ว่าจะเป็นการนำสี (Color) ของแต่ละจุด (Pixel) มาคิด, การคิดคำนวณเป็นบริเวณหลายจุดรวมกัน (Area) เช่น การดูลวดลาย (Pattern, Texture), การวิเคราะห์หารูปร่าง (Shape) และการวิเคราะห์แบบอื่นๆ โดยที่มาของรูปภาพนั้นอาจมาจากกล้องดิจิทัล หรือจากสื่อดิจิทัลต่างๆ แล้วนำไปผ่านกระบวนการเพื่อให้เกิดเป็นภาพใหม่ เช่น การทำภาพเบลอ (Blured Image) การทำภาพนูน (Emboss Image) การตรวจหาขอบภาพ (Edge Detector) การประมวลผลภาพดิจิทัล (digital image processing) เทคนิคและอัลกอริทึมต่างๆ ที่ใช้การประมวลผลภาพที่อยู่ในรูปแบบดิจิทัล (ภาพดิจิทัล) ภาพในที่นี้รวมถึง

ประมวลผลภาพ



รูปที่ 2.10 แสดงกระบวนการประมวลผลภาพ

สัญญาณดิจิทัลใน 2 มิติอื่นๆ โดยทั่วไปคำนึงเมื่อใช้อย่างกว้างๆ จะครอบคลุมถึงสัญญาณวิดีโอ (video) หรือภาพเคลื่อนไหว ซึ่งจะเป็นชุดของภาพนิ่ง เรียกว่า เฟรม (frame) หลายๆ ภาพต่อกันไปตามเวลา ซึ่งก็คือสัญญาณ 3 มิติ เมื่อนับเวลาเป็นมิติที่ 3 หรือ อาจจะครอบคลุมถึงสัญญาณ 3 มิติอื่นๆ เช่น ภาพ 3 มิติทางการแพทย์ หรือ อาจจะมากกว่านั้น เช่น ภาพ 3 มิติ และ หลายชนิด (multimodal image) ทฤษฎี Image Processing สามารถนำไปใช้ประโยชน์ได้มากมายหลายด้าน ไม่ว่าจะเป็นระบบรู้จำลายนิ้วมือ, ระบบคัดแยกเกรดหรือคุณภาพของพืชผลทางการเกษตร, ระบบอ่านรหัสไปรษณีย์อัตโนมัติเพื่อคัดแยกปลายทางของจดหมาย, ระบบเก็บข้อมูลรถที่เข้าและออกอาคาร โดยใช้ภาพถ่ายของป้ายทะเบียนรถ, ระบบรู้จำใบหน้าเพื่อเฝ้าระวังผู้ก่อการร้ายในอาคารสถานที่สำคัญ, ระบบคัดแยกสินค้าตามสีของผลิตภัณฑ์ และระบบคัดแยกสินค้ามีตำหนิ เป็นต้น

ในการประมวลผลภาพนั้น การกำหนดขอบเขตของภาพทุกภาพให้อยู่ในรูปสี่เหลี่ยม (Rectangular image model) เป็นวิธีที่นิยมใช้กันมากที่สุด เนื่องจากทำให้การอ่านภาพ การจัดเก็บข้อมูลภาพในหน่วยความจำ และการแสดงผลเป็นไปได้อย่างมีประสิทธิภาพ (สันสนีย์ เอื้อพันธ์ วิริยะกุล, 2555) การเก็บข้อมูลภาพลงหน่วยความจำของคอมพิวเตอร์สามารถทำได้โดยการจองหน่วยความจำของเครื่องไว้ในรูปของตัวแปรอะเรย์ (array) โดยค่าในแต่ละช่องของอะเรย์แสดงถึงคุณสมบัติของจุดภาพ (Pixel) และตำแหน่งของช่องอะเรย์เป็นตัวกำหนดตำแหน่งของจุดภาพ

มาตรฐานของสีในปัจจุบันมีอยู่หลายระบบด้วยกันขึ้นอยู่กับนำไปใช้ แต่โดยทั่วไปแล้วทุกมาตรฐานจะมีแนวคิดเดียวกันคือ การแทนสีด้วยจุดที่อยู่ภายในสเปส 3 มิติ โดยจะมีแกนอ้างอิงสำหรับจุดสีนั้นในสเปสซึ่งแต่ละแกนจะมีความเป็นอิสระต่อกัน ระบบสีที่นิยมใช้ได้แก่ ระบบ RGB HSV (Hue Saturation Value) และ HLS (Hue Lightness Saturation) (บุญธรรม ภัทราจารุกุล, 2556)

ระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง เขียว และน้ำเงิน โดยปกติจะนำไปใช้ในจอภาพแบบ CRT (Cathode ray tube) ระบบ RGB ที่นิยมใช้งานได้แก่ RGB_{CIE} RGB_{NTSC}

ระบบสี RGB ของ CIE ซึ่งอ้างอิงด้วยสีแดงที่ 700 nm สีเขียวเท่ากับ 546.1 nm และสีน้ำเงิน 435.8 nm

ระบบสี RGB ของ NTSC จะใช้สำหรับแสดงภาพของจอภาพแบบ CRT ระบบสี HSV (Hue Saturation Value) เป็นการพิจารณาสีโดยใช้ Hue Saturation และ Value ซึ่ง Hue คือ ค่าสีของสีหลัก (แดง เขียวและน้ำเงิน) ในทางปฏิบัติจะอยู่ระหว่าง 0 และ 255 ซึ่งถ้า Hue มีค่าเท่ากับ 0 จะแทนสีแดง และเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อยๆ สีก็จะเปลี่ยนแปลงไปตามสเปกตรัมของสีจนถึง 256 ก็จะกลับมาเป็นสีแดงอีกครั้ง ซึ่งสามารถแทนให้อยู่ในรูปขององศาได้ดังนี้คือ สีแดง = 0 องศา สีเขียวเท่ากับ 120 องศา สีน้ำเงินเท่ากับ 240 องศา Saturation คือความบริสุทธิ์ของสีซึ่งถ้า Saturation มีค่าเท่ากับ 0 แล้วสีที่ได้จะไม่มี Hue ซึ่งจะเป็นสีขาวล้วนแต่ถ้า Saturation มีค่าเท่ากับ 255 แสดงว่า จะไม่มีแสงสีขาวผสมอยู่เลย Value คือความสว่างของสี ซึ่งสามารถวัดได้โดยค่าความเข้มของความสว่างของแต่ละสีที่ประกอบกัน

ระบบสีแบบ HLS (Hue lightness saturation) จะมีลักษณะคล้ายกับ HSV ดังนี้คือสีของระบบจะขึ้นอยู่กับ Hue Lightness และ Saturation Hue คือค่าของสีหลักซึ่งมีสีน้ำเงินอยู่ที่ 0 องศา สีเขียวอยู่ที่ 120 องศาและสีแดงอยู่ที่ 240 องศา Lightness คือค่าความสว่างซึ่งจะมีค่าเปลี่ยนแปลงตามแนวแกน L โดยที่ $L = 0$ จะเป็นสีดำ $L = 1$ จะเป็นสีขาว ระบบสีแบบ CMY เป็นระบบสีที่พัฒนาขึ้นมาใช้สำหรับการพิมพ์ภาพสีโดยมีสีหลักคือสี Cyan Magenta และ Yellow ซึ่งเรียกว่า Subtractive primaries Color (สีแดง เขียวและน้ำเงิน เรียกว่า Additive primaries Color) ระบบสีแบบ CMY สามารถหาได้โดยการนำเอาสีในระบบ RGB ลบกับสีขาว การแปลงค่าสีระหว่างระบบสีต่าง ๆ ระหว่างระบบสามารถทำได้โดยการใช้ Matrix

2.5 โปรแกรม MATLAB

MATLAB ย่อมาจาก MATRIX LABORATORY เป็น โปรแกรมคำนวณทางคณิตศาสตร์ ถูกพัฒนาโดยบริษัท The Mathworks, Inc โปรแกรม MATLAB สามารถเขียนโปรแกรมและแสดงผลออกมาได้ในโปรแกรมตัวเดียว โดย MATLAB เป็นโปรแกรมที่นิยมใช้ในงานด้านวิศวกรรมศาสตร์ เพราะสามารถเลือกใช้ฟังก์ชันทางคณิตศาสตร์ได้อย่างหลากหลายทำให้สามารถพัฒนาโปรแกรมได้เป็นอย่างดี MATLAB นั้นสามารถใช้งานได้อย่างหลากหลายไม่ว่าจะเป็นการคำนวณทั่วไป การวิเคราะห์และการประมวลผลข้อมูล การสร้างและทดสอบแบบจำลอง เป็นต้น การประมวลผลของ MATLAB จะมีลักษณะการทำงานคือ เขียนคำสั่งเข้าไป เพื่อให้ MATLAB ประมวลผลไปที่ละคำสั่ง หรือสามารถรวบรวมชุดคำสั่งให้เป็นโปรแกรมได้ MATLAB นั้นได้ถูกสร้างขึ้นเพื่อต้องการแก้ปัญหาตัวแปรที่มีลักษณะเป็น Matrix ได้ง่ายขึ้น นอกจาก

โปรแกรม MATLAB แล้ว บริษัท MathWorks ผู้ผลิต MATLAB ยังได้สร้างเครื่องมือหรือที่เรียกว่า toolbox ซึ่งเป็นโปรแกรมที่เขียนขึ้นเพื่อประกอบกับการใช้ MATLAB สำหรับงานที่จำเพาะเจาะจงหลายประเภท Toolbox นั้นเป็นการนำเอาโปรแกรมที่เขียนขึ้นเป็นฟังก์ชันสำหรับ MATLAB เพื่อรวมเข้าเพื่อให้ผู้ใช้งานมีความสะดวกในการเรียกใช้มากขึ้น ทำให้ผู้ใช้ไม่จำเป็นต้องสร้างโปรแกรมขึ้นมาใช้งานเอง โดย toolbox ที่สร้างขึ้นจะครอบคลุมการทำงานในหลายด้าน เช่น signal processing, control systems, neural networks, fuzzy logic, simulation เป็นต้น

การป้อนคำสั่งสำหรับการประมวลผลของโปรแกรม MATLAB สามารถป้อนได้ 2 วิธี (โศรฐา แจ้งการ,) คือ 1. เป็นการป้อนคำสั่ง หรือค่าตัวแปร บนหน้าต่างคำสั่งตามขั้นตอนของการแก้ปัญหา 2. เป็นการกำหนดค่าตัวแปร และชุดคำสั่ง ที่ใช้ทำการประมวลผลตามขั้นตอนของการคำนวณลงในไฟล์ ซึ่งจะต้องเก็บไฟล์ไว้ด้วยนามสกุล .m อาจเรียกชุดคำสั่งนี้ว่า M-file หรือ Script-file จะเห็นได้ว่าโปรแกรม MATLAB สามารถทำงานได้หลากหลาย จึงทำให้เหมาะสำหรับผู้ที่ต้องการใช้งานเฉพาะทางหรืองานด้านวิศวกรรมขั้นสูงเป็นอย่างยิ่ง

เนื่องจาก MATLAB เป็นโปรแกรมสำเร็จรูป ดังนั้นในการเขียนชุดคำสั่งต่างๆ MATLAB ได้พยายามรวบรวม function ที่ใช้อยู่เป็นประจำเข้าไว้ด้วยกันเพื่อสะดวกในการใช้งาน function เหล่านี้จะอยู่ภายใน sub-directories ซึ่งมักจะเขียนเป็น file ที่เรียกว่า M-file แม้ว่า MATLAB จะเป็นโปรแกรมที่มีความสามารถในการทำงานในการคำนวณได้มากมาย แต่ในหลายๆ กรณีก็มีความจำเป็นที่จะต้องติดต่อ ส่งผ่านข้อมูลกับโปรแกรมอื่น หรือต้องมีการเก็บข้อมูลเพื่อที่จะนำไปใช้ต่อไป ดังนั้นจึงจำเป็นอย่างยิ่งที่จะต้องมียระบบการบันทึกและรับส่งข้อมูล

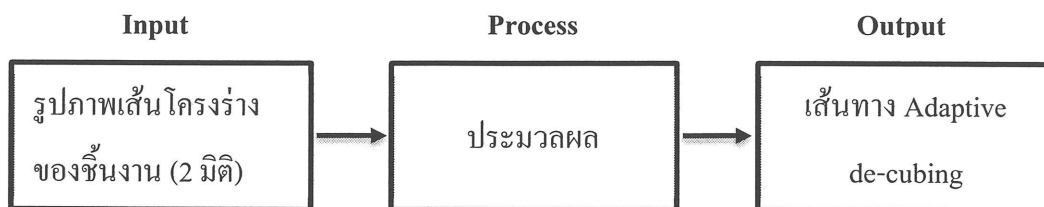
บทที่ 3

การดำเนินงานวิจัย

ในบทนี้ได้นำเสนอเกี่ยวกับวิธีการดำเนินงาน รวมถึงรายละเอียดของขั้นตอนการดำเนินงานที่ใช้ในการทำวิจัยในครั้งนี้ โดยอธิบายรายละเอียดขั้นตอนการนำเข้าข้อมูลเป็นรูปภาพ โดยการประยุกต์ใช้เทคโนโลยีการประมวลผลภาพ (Image processing) การกำหนดขอบเขตของรูปภาพหน้าตัดชิ้นงานโดยการประยุกต์ใช้หลักการสร้างกล่องขอบเขต (Bounding box) การกำหนดเงื่อนไขการประมวลผล และโครงสร้างของโปรแกรม

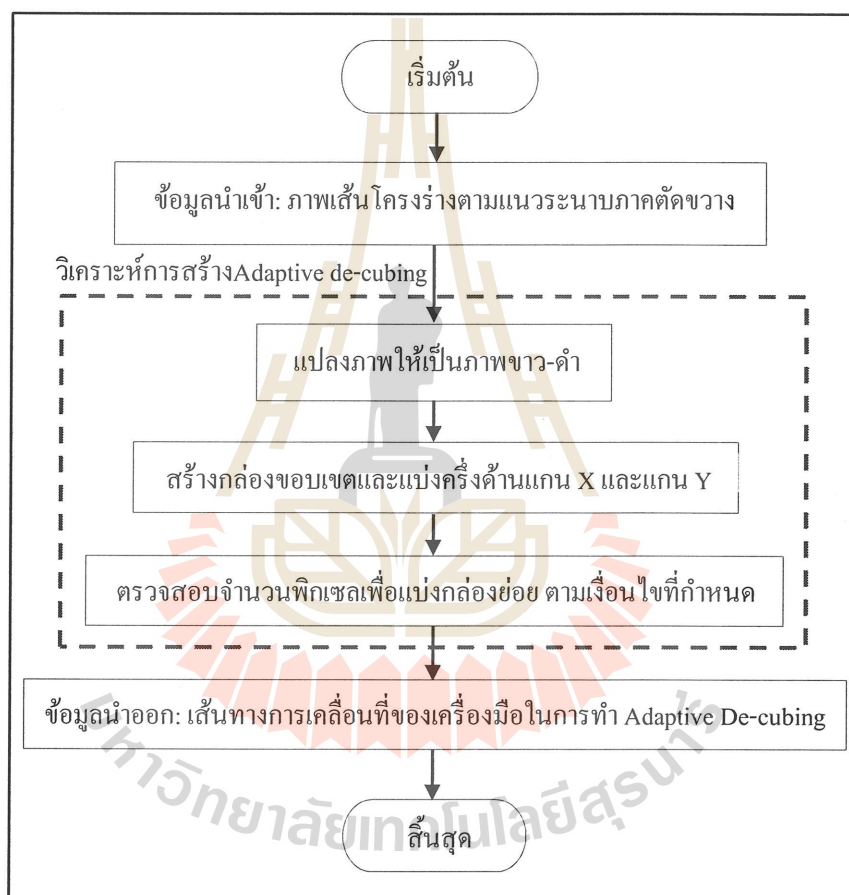
3.1 วิธีการดำเนินงานวิจัย

การดำเนินงานวิจัยเริ่มต้นจากการศึกษาข้อมูลจากงานวิจัยที่ผ่านมาเกี่ยวกับเทคโนโลยีการสร้างชิ้นงานต้นแบบรวดเร็วแต่ละประเภท และได้ศึกษาการสร้างชิ้นงานต้นแบบรวดเร็วด้วยเทคนิค Laminated Object Manufacturing ทำให้ทราบเกี่ยวกับกระบวนการ และข้อจำกัดของ LOM แล้วทำให้พบว่ากระบวนการที่สำคัญของ LOM อย่างหนึ่งคือกระบวนการ De-cubing แบบ Adaptive De-cubing (ศรินยา พุทธวงศ์และคณะ, 2559) จากนั้นจึงได้ทำการศึกษาเกี่ยวกับกระบวนการ De-cubing แล้วจึงได้ทำการศึกษาวิธีการและพัฒนาโปรแกรมทางคอมพิวเตอร์ เพื่อมาช่วยในการคำนวณเส้นทางการตัดชิ้นงานในกระบวนการ De-cubing เพื่อลดระยะเวลาในการสร้างชิ้นงานต้นแบบ กรอบแนวความคิดในการดำเนินการวิจัยประกอบด้วยข้อมูลนำเข้าซึ่งเป็นรูปภาพเส้นโครงร่าง (contour) ของชิ้นงานต้นแบบ (2 มิติ) ทำการประมวลผลข้อมูลจะได้ข้อมูลนำออกเป็นเส้นทางการทำ Adaptive de-cubing ดังแสดงในรูปที่ 3.1 และกระบวนการดำเนินงานวิจัยทั้งหมดสามารถแสดงได้ในรูปที่ 3.2 โดยเริ่มต้นมีข้อมูลนำเข้าเป็นภาพเส้นโครงร่าง (contour) ตามแนวระนาบภาคตัดขวาง (cross-section) โดยมีการนำเข้าข้อมูลเส้นโครงร่างทีละชั้น (layer by layer) ทำการแปลงภาพให้เป็นภาพขาวดำเพื่อให้แยกส่วนที่เป็นชิ้นงานและส่วนที่ไม่ใช่เนื้อชิ้นงานแล้วสร้างกล่องขอบเขต (Bounding box) เพื่อกำหนดขอบเขตในการพิจารณา แบ่งครึ่งกล่องขอบเขต (Bounding box) ทั้งด้านแกน X และแกน Y จะได้กล่องขอบเขตย่อย 4 กล่อง เพื่อใช้เป็นขอบเขตเริ่มต้นในการพิจารณาต่อไป จากนั้นทำการพิจารณาจำนวนพิกเซลในกล่องย่อยที่ละกล่อง ทั้งด้านแกน X และแกน Y ว่าควรทำการแบ่งกล่องย่อยอีกหรือไม่ เมื่อพิจารณาตามเงื่อนไขเสร็จสิ้น จะได้ข้อมูลนำออกเป็นเส้นทางการเคลื่อนที่ของเครื่องมือตัดในส่วนของการทำ Adaptive De-cubing



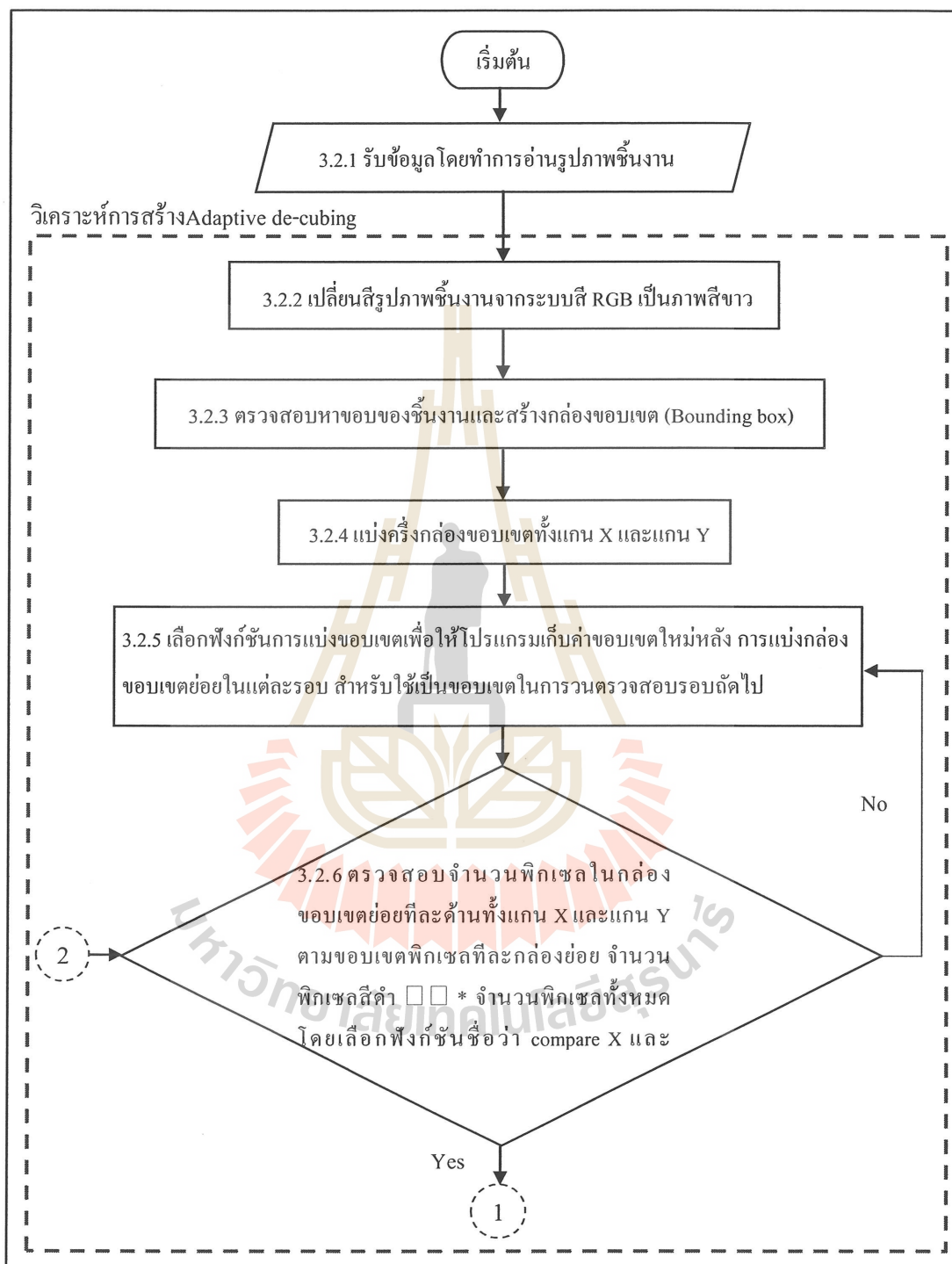
รูปที่ 3.1 แสดงกรอบแนวคิดงานวิจัย

ขั้นตอนการดำเนินงานวิจัยทั้งหมดแสดงได้ดังนี้

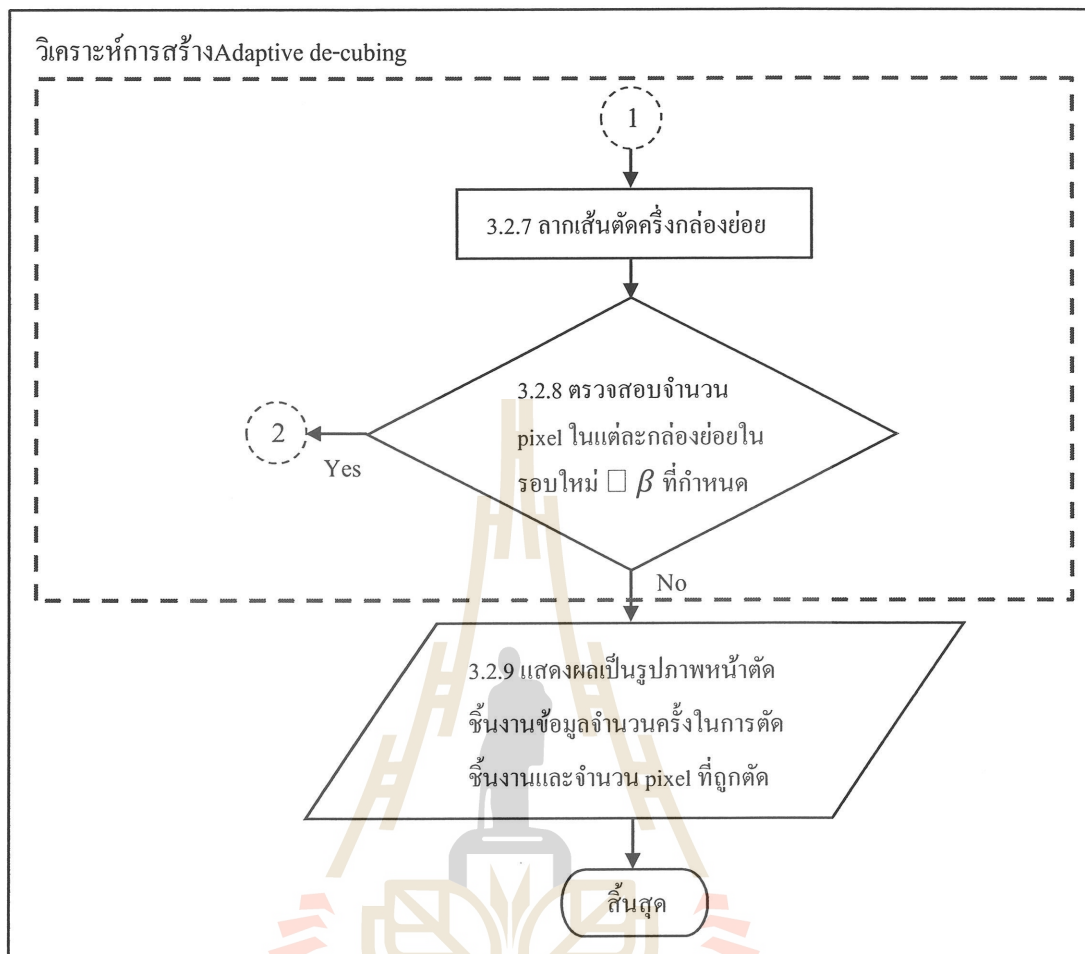


รูปที่ 3.2 แสดงวิธีการดำเนินงานทั้งหมด

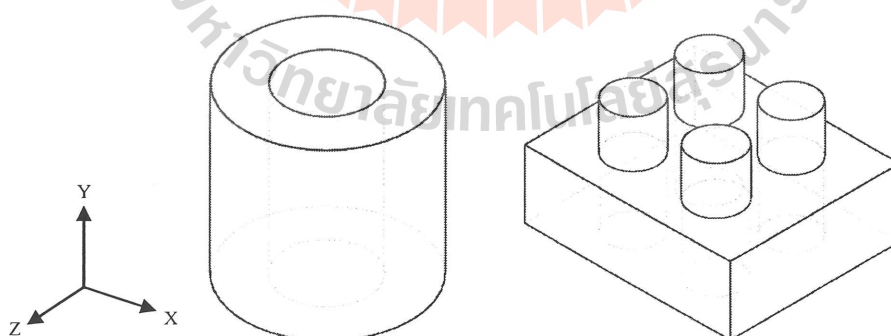
3.2 โครงสร้างโปรแกรมการวิเคราะห์ Adaptive de-cubing



รูปที่ 3.3 โครงสร้างโปรแกรมการวิเคราะห์ Adaptive de-cubing



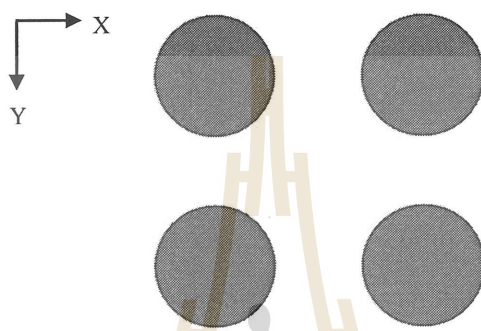
รูปที่ 3.4 โครงสร้างโปรแกรมการวิเคราะห์ Adaptive de-cubing (ต่อ)



รูปที่ 3.5 ตัวอย่างชิ้นงาน 3 มิติ

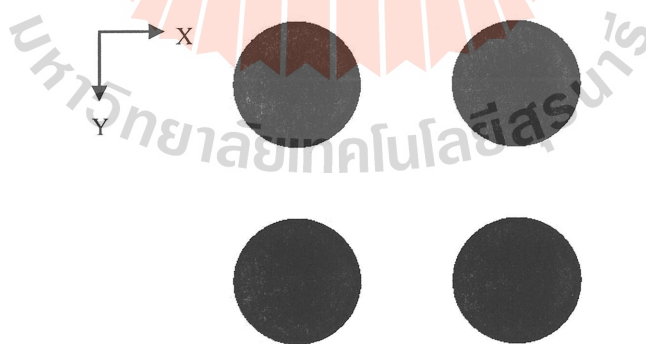
การวิเคราะห์ Adaptive de-cubing จะเป็นกระบวนการหลังจากที่สร้างชิ้นงานต้นแบบ 3 มิติ ดังแสดงในรูปที่ 3.5 และทำการแบ่งชิ้นงานเรียบร้อยแล้ว โดยภาพเส้นโครงร่างจะถูกนำเข้า ที่ละชั้นเป็นภาพ 2 มิติ รายละเอียดโครงสร้างโปรแกรมการวิเคราะห์ Adaptive de-cubing สามารถ แสดงได้ดังต่อไปนี้

3.2.1 เริ่มต้นจากใส่ชื่อรูปภาพ และประเภทของไฟล์รูปภาพที่ต้องการนำเข้า (.jpg, .png) ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 แสดงรูปภาพนำเข้า

3.2.2 เมื่อนำเข้ารูปภาพแล้วโปรแกรมจะทำการแปลงไฟล์รูปภาพให้เป็นภาพสีขาว-ดำ เพื่อให้โปรแกรมสามารถแยกส่วนที่เป็นเนื้อชิ้นงานออกจากส่วนที่ไม่ใช่เนื้อชิ้นงาน โดยบริเวณที่เป็นเนื้อชิ้นงานจะเป็นสีดำ และบริเวณที่ไม่ใช่เนื้อชิ้นงานจะเป็นสีขาว โดยการแปลงรูปภาพเป็น ขาว-ดำ ดังแสดงในรูป 3.7



รูปที่ 3.7 แสดงรูปภาพนำเข้าหลังจากถูกแปลงเป็นภาพขาว-ดำ

3.2.3 โปรแกรมจะทำการกำหนดขอบเขตที่จะสร้างกล่องขอบเขต (Bounding box) โดยพิจารณาจากการตรวจสอบหาตำแหน่งพิกเซลที่เป็นสีดำพิกเซลแรก และพิกเซลสุดท้ายของรูปภาพด้านแกน X โดยขนาดของกล่องขอบเขตด้านแกน X สามารถแสดงได้ดังสมการที่ (3.1)

$$SB_X = Ex_0 - Bx_0 \quad (3.1)$$

เมื่อ SB_X = ขนาดของกล่องขอบเขตด้านแกน X

Bx_0 = พิกเซลที่เป็นสีดำพิกเซลแรกของรูปภาพด้านแกน X

Ex_0 = พิกเซลที่เป็นสีดำพิกเซลสุดท้ายของรูปภาพด้านแกน X

ทำการตรวจสอบหาตำแหน่งพิกเซลที่เป็นสีดำพิกเซลแรก และพิกเซลสุดท้ายของรูปภาพด้านแกน Y โดยขนาดของกล่องขอบเขตด้านแกน Y สามารถแสดงได้ดังสมการที่ (3.2)

$$SB_Y = Ey_0 - By_0 \quad (3.2)$$

เมื่อ SB_Y = ขนาดของกล่องขอบเขตด้านแกน Y

By_0 = พิกเซลที่เป็นสีดำพิกเซลแรกของรูปภาพด้านแกน Y

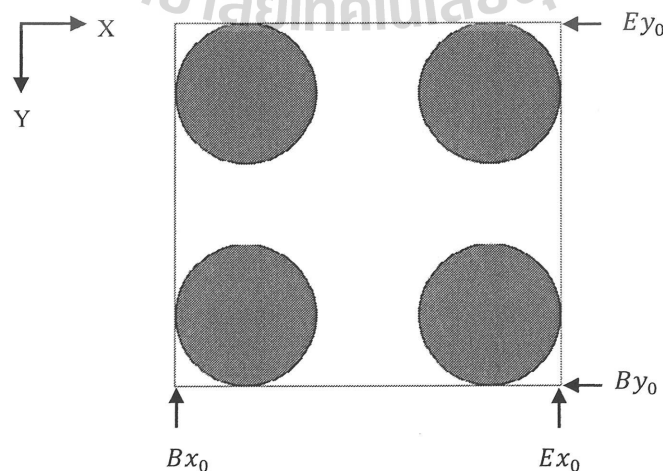
Ey_0 = พิกเซลที่เป็นสีดำพิกเซลสุดท้ายของรูปภาพด้านแกน Y

โปรแกรมทำการแสดงเส้นกล่องขอบเขตของรูปภาพเส้นโครงร่างที่นำเข้ามาจะมีขนาดดังแสดงในสมการที่ (3.3)

$$SB = (Ex_0 - Bx_0) \times (Ey_0 - By_0) \quad (3.3)$$

เมื่อ SB = ขนาดของกล่องขอบเขตของรูปภาพเส้นโครงร่างที่นำเข้ามา

กล่องขอบเขตของรูปภาพเส้นโครงร่างที่นำเข้ามาสามารถแสดงได้ดังรูปที่ 3.8 เมื่อกำหนดกล่องขอบเขตสำหรับการพิจารณาได้แล้ว จึงใช้เป็นขอบเขตเริ่มต้นการวิเคราะห์ต่อไป



รูปที่ 3.8 แสดงการสร้างกล่องขอบเขต

3.2.4 โปรแกรมจะทำการแบ่งครึ่งกล่องขอบเขตทั้งด้านแกน X และแกน Y โดยใช้ขอบเขตที่มากที่สุดลบด้วยขอบเขตที่น้อยที่สุดแล้วหารด้วย 2 ทั้งด้านแกน X และแกน Y เนื่องจากกล่องขอบเขตที่สร้างขึ้นอาจจะไม่ได้เริ่มต้นจากตำแหน่งพิกเซลที่ 1 เสมอไป ดังนั้นโปรแกรมจะต้องบวกค่า Bx_0 หรือ By_0 เข้าไปด้วยเพื่อจะทำให้ได้ตำแหน่งกึ่งกลางของกล่องขอบเขตดังนี้

ตำแหน่งพิกเซลที่แบ่งครึ่งกล่องขอบเขตด้านแกน X หาได้ดังสมการที่ (3.4)

$$C_x = Bx_0 + [(Ex_0 - Bx_0)/2] \quad (3.4)$$

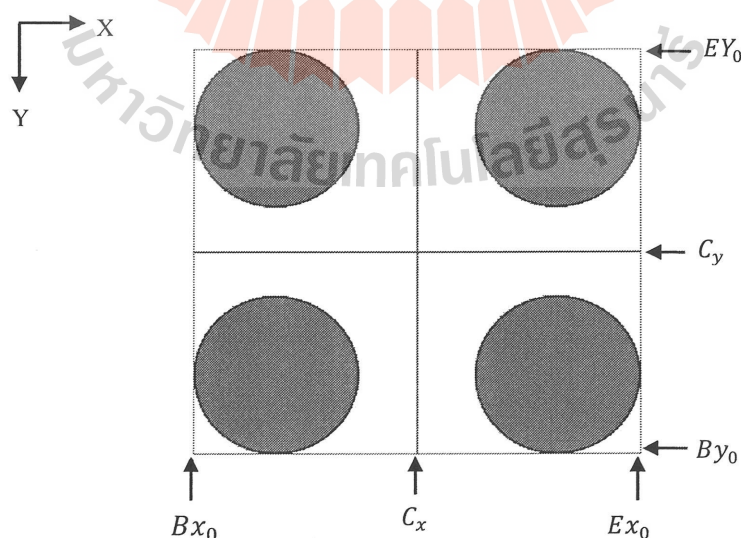
เมื่อ C_x = ตำแหน่งพิกเซลที่แบ่งครึ่งกล่องขอบเขตด้านแกน X

ตำแหน่งพิกเซลที่แบ่งครึ่งกล่องขอบเขตด้านแกน Y หาได้ดังสมการที่ (3.5)

$$C_y = By_0 + [(Ey_0 - By_0)/2] \quad (3.5)$$

เมื่อ C_y = ตำแหน่งพิกเซลที่แบ่งครึ่งกล่องขอบเขตด้านแกน Y

ในขั้นตอนนี้จะทำให้ได้กล่องขอบเขตที่จะพิจารณาใหม่เป็นกล่องย่อย 4 กล่องดังแสดงในรูปที่ 3.9 ในกรณีที่มีเศษทศนิยมที่เกิดจากการหาร โปรแกรมจะทำการปัดเศษโดยหากเศษทศนิยมมีค่าต่ำกว่า 5 โปรแกรมจะปัดเศษลง หากเศษทศนิยมตั้งแต่ 5 ขึ้นไป โปรแกรมจะทำการปัดเศษขึ้น โคนคำสั่งที่ใช้ในการปัดเศษทศนิยม โดยจากการคำนวณด้วยการปัดเศษทศนิยมดังกล่าวอาจส่งผลให้เกิดเส้นที่ติดกันหรือระยะห่างไม่เท่ากันได้



รูปที่ 3.9 แสดงการแบ่งกล่องขอบเขตออกเป็น 4 กล่องย่อย

3.2.5 โปรแกรมจะเลือกฟังก์ชันการแบ่งกล่องขอบเขตเพื่อเก็บค่าพิกัดใหม่ของแต่ละกล่องย่อยสำหรับการใช้ในการพิจารณาการแบ่งในรอบถัดไป โดยตรวจสอบจำนวนพิกเซลของกล่องขอบเขตย่อย ตามเงื่อนไขที่กำหนด

จำนวนพิกเซลที่น้อยที่สุดของกล่องขอบเขตย่อยที่จะพิจารณาต่อตามเงื่อนไขที่กำหนดที่ใช้กำหนดในโปรแกรมนั้นสามารถกำหนดได้ตามความต้องการโดยต้องคำนึงถึงขนาดของไฟล์ภาพที่นำเข้าและความเรียบผิวของชิ้นงานที่เราต้องการโดยในโปรแกรมนี้ได้กำหนดให้เป็น β พิกเซล เพื่อให้สามารถแสดงผลได้ละเอียดและเห็นได้ชัดเจน หากมีการกำหนดค่า β ให้มีค่าไม่เท่ากันจะทำให้รอบ (R) ในการวนตรวจสอบแตกต่างกันส่งผลให้เส้นทาง Adaptive de-cubing ที่ได้จะมีความละเอียดหรือความหยาบของผิว (Surface roughness ; Ra) ของชิ้นงานต้นแบบที่แตกต่างกันอีกด้วย

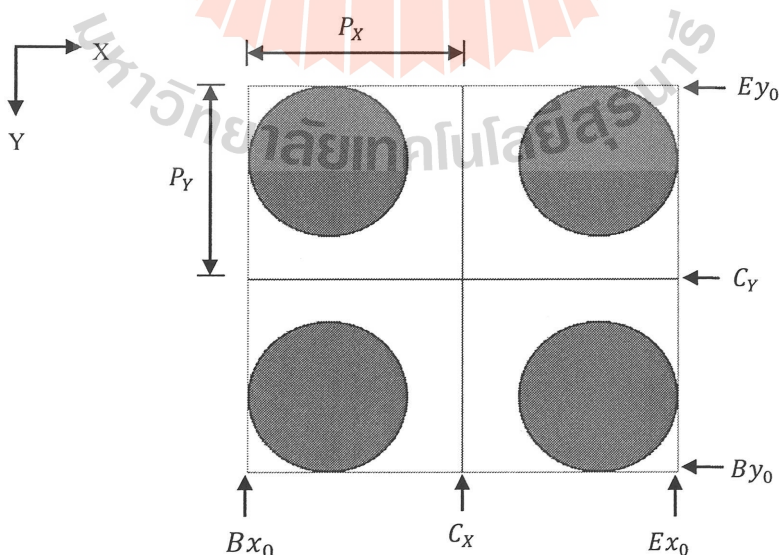
เมื่อกำหนดให้ β = จำนวนพิกเซลที่น้อยที่สุดของกล่องขอบเขตย่อย ที่จะใช้ตรวจสอบว่าต้องแบ่งกล่องขอบเขตย่อยต่อตามเงื่อนไขที่กำหนด หรือสิ้นสุดการแบ่งแล้ว

- เงื่อนไขการตรวจสอบค่า β ในแกน X แสดงได้ดังสมการที่ (3.6)

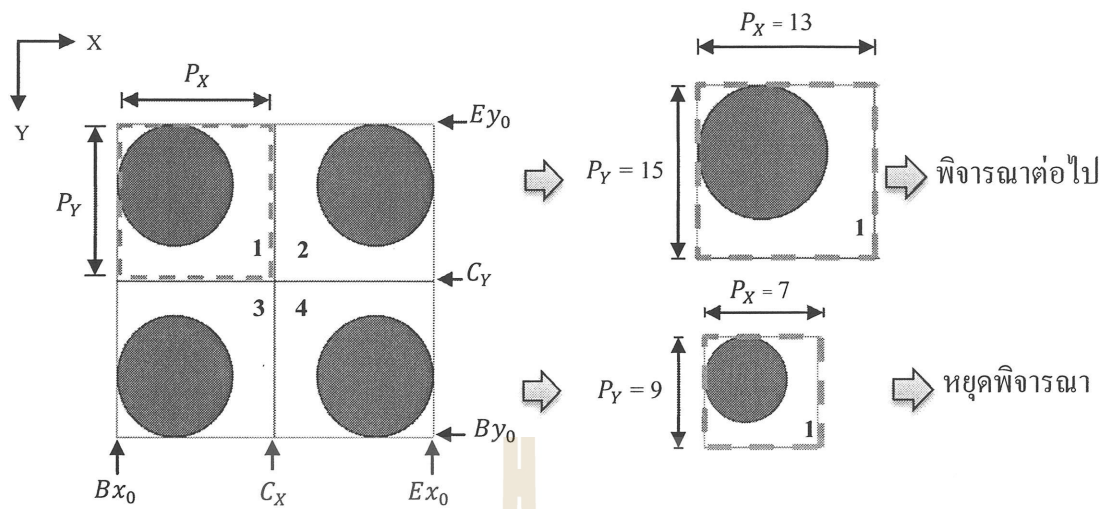
$$P_x \geq \beta \quad (3.6)$$

- เงื่อนไขการตรวจสอบค่า β ในแกน Y แสดงได้ดังสมการที่ (3.7)

$$P_y \geq \beta \quad (3.7)$$



รูปที่ 3.10 แสดงการตรวจสอบค่า β ในแกน X และแกน Y



รูปที่ 3.11 แสดงตัวอย่างการตรวจสอบค่า β

หากกำหนดให้ β ทั้งแกน X และแกน Y เท่ากับ 10 จะได้เงื่อนไขในการพิจารณาดังนี้

$$P_X \geq 10 \text{ และ } P_Y \geq 10$$

$$\text{กรณีที่ } P_X = 13, P_Y = 15$$

$$\text{ตรวจสอบแกน X ; } P_X \geq 10$$

$$13 \geq 10$$

$$\text{ตรวจสอบแกน Y ; } P_Y \geq 10$$

$$15 \geq 10$$

พบว่าเงื่อนไขเป็นจริงดังนั้นทำการพิจารณาการแบ่งกล่องขอบเขตย่อยนั้นต่อไป

$$\text{กรณีที่ } P_X = 7, P_Y = 9$$

$$\text{ตรวจสอบแกน X ; } P_X \geq 10$$

$$7 \geq 10$$

$$\text{ตรวจสอบแกน Y ; } P_Y \geq 10$$

$$9 \geq 10$$

พบว่าเงื่อนไขไม่เป็นจริงดังนั้นหยุดพิจารณาการแบ่งกล่องขอบเขตย่อยนั้น

จากเงื่อนไข $P_X \geq \beta$ หากเป็นจริง ทำการพิจารณาการแบ่งกล่องขอบเขตย่อยในแกน X ตามเงื่อนไขต่อไป ถ้าไม่เป็นจริงหยุดพิจารณาการแบ่งกล่องขอบเขตย่อยในแกน X และจากเงื่อนไข $P_Y \geq \beta$ หากเป็นจริง ทำการพิจารณาการแบ่งกล่องขอบเขตย่อยในแกน Y ตามเงื่อนไขต่อไป ถ้าไม่เป็นจริงหยุดพิจารณาการแบ่งกล่องขอบเขตย่อยในแกน Y ดังแสดงในรูปที่ 3.11 โดยจะทำการตรวจสอบกล่องขอบเขตย่อยที่ละกล่องตามลำดับ $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ ตามลำดับ

3.2.6 เลือกฟังก์ชันการแบ่งกล่องย่อยเพื่อตรวจสอบจำนวนพิกเซลในแกน X และแกน Y ชื่อฟังก์ชัน “compare X” และ “compare Y” ในการวนรอบตรวจสอบจำนวนพิกเซลในกล่องขอบเขตย่อยที่ละด้านทั้งแกน X และแกน Y ตามขอบเขตพิกเซลที่ละกล่องย่อย หากพบว่าจำนวนพิกเซลสีดำ $\geq \tau$ * จำนวนพิกเซลทั้งหมดในแกน X ให้ดำเนินการในขั้นตอนที่ 3.2.7 หากไม่เป็นไปตามเงื่อนไขให้วนไปตรวจสอบกล่องย่อยถัดไปในขั้นตอนที่ 3.2.5 การกำหนดค่า threshold ในกรณีนี้คือการกำหนดอัตราส่วนของจำนวนพิกเซลสีดำต่อพิกเซลสีขาวที่เป็นเกณฑ์ในการพิจารณาว่ากล่องย่อยในแกน X หรือแกน Y นั้นจะต้องทำการแบ่งครึ่งหรือไม่ การกำหนดอัตราส่วนค่า threshold จะอยู่ในช่วงระหว่าง 0.01 - 0.99 โดยการกำหนดค่าอัตราส่วนค่า threshold ที่ต่างกันจะส่งผลให้ได้ตำแหน่งการทำงานของเครื่องมือตัดในการทำ Adaptive de-cubing ที่แตกต่างกันนั่นเอง

เงื่อนไขการตรวจสอบจำนวนพิกเซลสีดำในแกน X สามารถแสดงได้ในสมการที่ (3.11)

$$BP_X \geq \tau \times P_X \quad (3.11)$$

เมื่อ BP_X จำนวนพิกเซลสีดำในแกน X

τ = ค่าอัตราส่วนพิกเซลสีดำต่อพิกเซลสีขาว

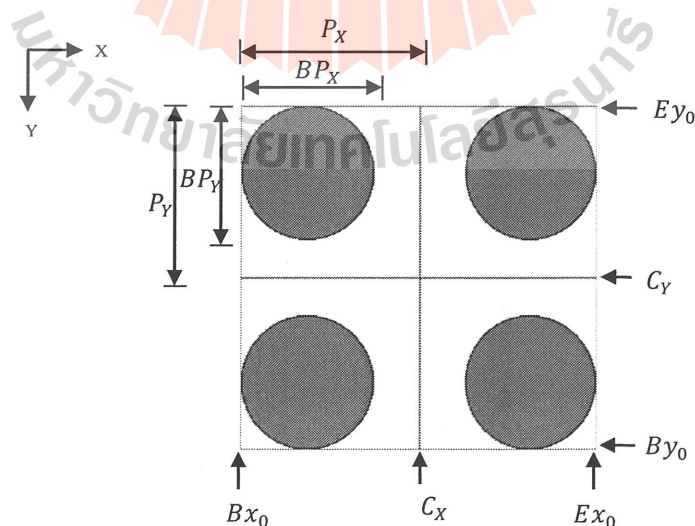
P_X = จำนวนพิกเซลทั้งหมดในแกน X

เงื่อนไขการตรวจสอบจำนวนพิกเซลสีดำในแกน Y สามารถแสดงได้ในสมการที่ (3.12)

$$BP_Y \geq \tau \times P_Y \quad (3.12)$$

เมื่อ BP_Y จำนวนพิกเซลสีดำในแกน Y

P_Y = จำนวนพิกเซลทั้งหมดในแกน Y



รูปที่ 3.12 แสดงค่าต่างๆ ที่ต้องใช้การตรวจสอบค่า τ ในแกน X และแกน Y

จากเงื่อนไข $BP_X \geq \tau \times P_X$ หากเป็นจริงทำการแบ่งกล่องขอบเขตย่อยในแกน X ตามเงื่อนไข ถ้าไม่เป็นจริงหยุดพิจารณาการแบ่งกล่องขอบเขตย่อยในแกน X และทำการพิจารณาการแบ่งกล่องขอบเขตย่อยนั้นในด้านแกน Y และ $BP_Y \geq \tau \times P_Y$ หากเป็นจริงทำการแบ่งกล่องขอบเขตย่อยในแกน Y ตามเงื่อนไขต่อไป ถ้าไม่เป็นจริงหยุดพิจารณาการแบ่งกล่องขอบเขตย่อยในแกน Y

โดยจะทำการตรวจสอบกล่องขอบเขตย่อยที่ละกล่องตามลำดับ $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ ตามลำดับ หากกำหนดให้ τ ทั้งแกน X และแกน Y เท่ากับ 0.25 และ 0.80 จะได้เงื่อนไขในการพิจารณาดังนี้

กรณีกำหนดให้ τ ทั้งแกน X และแกน Y เท่ากับ 0.25 เงื่อนไขในการตรวจสอบจะเป็นดังนี้ $BP_X \geq (0.25 \times P_X)$ และ $BP_Y \geq (0.25 \times P_Y)$

ตรวจสอบแกน X ; $BP_X \geq (0.25 \times P_X)$

$$11 \geq (0.25 \times 15)$$

$$11 \geq 3.75$$

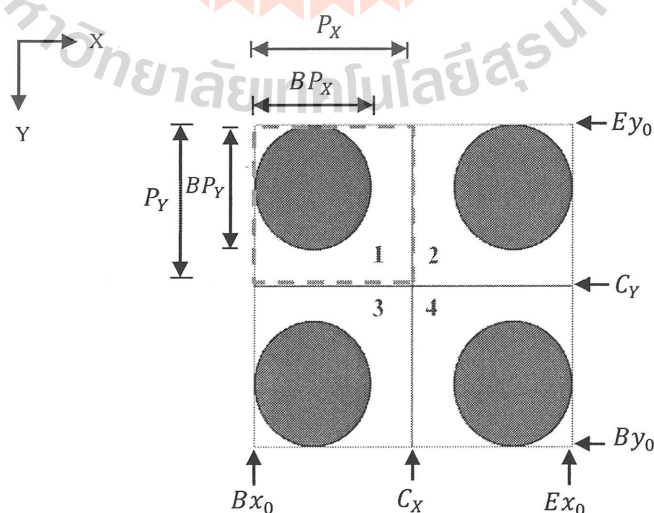
พบว่าเงื่อนไขเป็นจริงดังนั้นทำการแบ่งกล่องขอบเขตย่อยด้านแกน X

ตรวจสอบแกน Y ; $BP_Y \geq (0.25 \times P_Y)$

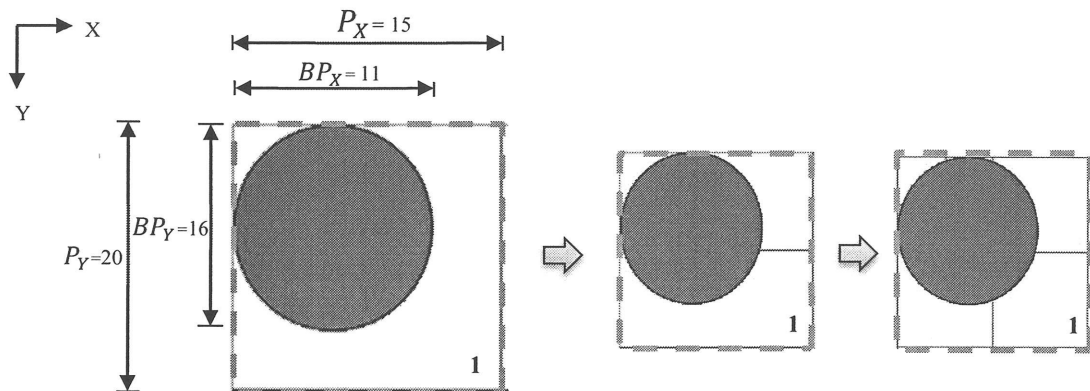
$$16 \geq (0.25 \times 20)$$

$$16 \geq 5$$

พบว่าเงื่อนไขเป็นจริงดังนั้นทำการแบ่งกล่องขอบเขตย่อยด้านแกน Y



รูปที่ 3.13 แสดงตัวอย่างลำดับการตรวจสอบค่า τ



รูปที่ 3.14 แสดงการการตรวจสอบค่า τ ในแกน X และแกน Y ของกล่องขอบเขตย่อย

ความสัมพันธ์ของจำนวนพิกเซลที่น้อยที่สุดของกล่องขอบเขตย่อยที่จะพิจารณาต่อตามเงื่อนไขที่กำหนดและจำนวนรอบในการตรวจสอบ สามารถแสดงได้ดังสมการที่ (3.8)

$$\beta \propto \frac{1}{R} \quad (3.8)$$

เมื่อ β = จำนวนพิกเซลที่น้อยที่สุดของกล่องขอบเขตย่อยที่จะพิจารณาต่อตามเงื่อนไขที่กำหนด

R = จำนวนรอบในการตรวจสอบ

จากสมการ (3.8) เมื่อกำหนดจำนวนพิกเซลที่น้อยที่สุดของกล่องขอบเขตย่อยที่จะพิจารณาต่อตามเงื่อนไขที่กำหนดให้มีความมากจะทำให้จำนวนรอบในการตรวจสอบนั้นลดลง

ความสัมพันธ์ของอัตราส่วนของพิกเซลสีดำต่อพิกเซลสีขาวและความหนาแน่นของชิ้นงาน สามารถแสดงได้ดังสมการที่ (3.9)

$$\tau \propto R_a \quad (3.9)$$

เมื่อ R_a = ความหนาแน่นของชิ้นงาน

จากสมการ (3.9) หากมีการกำหนดค่าอัตราส่วนของจำนวนพิกเซลสีดำต่อพิกเซลสีขาวให้มีค่าน้อยจะทำให้ Adaptive de-cubing มีความละเอียดมากขึ้นหรือจะทำให้ความหนาแน่นของชิ้นงานจะลดลงตามไปด้วย

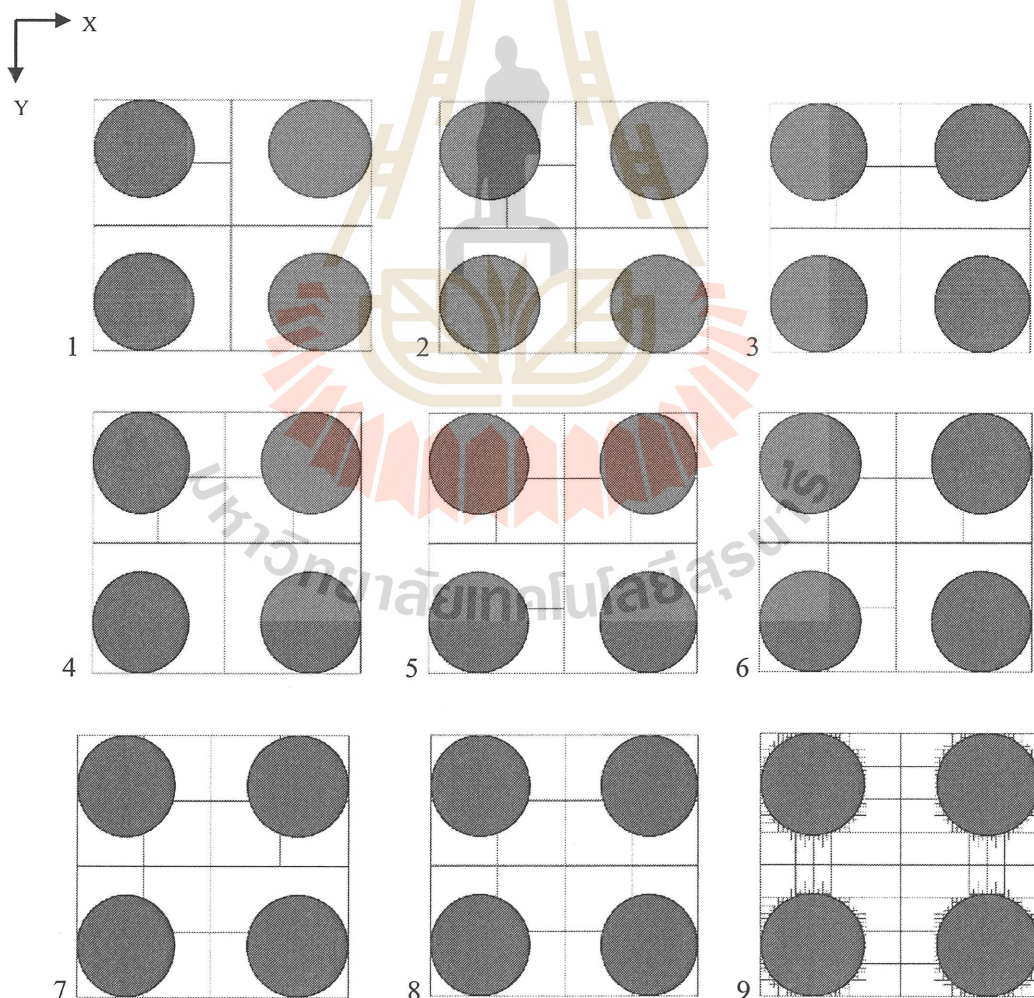
ความสัมพันธ์ระหว่างอัตราส่วนพิกเซลสีดำต่อพิกเซลสีขาว, ระยะทางการเดินทางของเครื่องมือตัด และระยะเวลาในการสร้างชิ้นงานต้นแบบ สามารถแสดงได้ดังสมการที่ (3.10)

$$\frac{1}{\tau} \propto S \propto T \quad (3.10)$$

เมื่อ S = ระยะทางการเดินทางของเครื่องมือตัด
 T = ระยะเวลาในการสร้างชิ้นงานต้นแบบ

จากสมการ (3.10) เมื่อค่าอัตราส่วนพิกเซลสีดำต่อพิกเซลสีขาวให้มีค่าน้อยจะทำให้ระยะทางการเดินทางของเครื่องมือตัดและระยะเวลาในการสร้างชิ้นงานต้นแบบเพิ่มมากขึ้น

3.2.7 ทำการแบ่งครึ่งกล่องย่อยตามเงื่อนไขในข้อ 3.2.6 โดยจะแสดงเส้นสีแดงในพิกเซลที่ต้องทำการแบ่ง ดังแสดงตัวอย่างในรูปที่ 3.10



รูปที่ 3.15 แสดงลำดับการวิเคราะห์การแบ่งกล่องย่อยเมื่อกำหนด $\beta = 10$, $\tau = 0.50$

3.2.8 ตรวจสอบจำนวน pixel ในแต่ละกล่องย่อยในการวนรอบใหม่หลังการแบ่งครึ่งกล่องย่อยในรอบที่แล้ว หากพบว่ามีกล่องย่อยใดที่มีขอบเขตในด้านแกน X หรือ Y มีจำนวนพิกเซล $\geq T$ ให้ทำการวนไปในขั้นตอนที่ 3.2.6 หากไม่มีกล่องย่อยใดที่มีจำนวนพิกเซล $\geq T$ ให้ดำเนินการในขั้นตอนที่ 3.2.9

3.2.9 กรณีที่วนรอบตรวจสอบแล้วพบว่าไม่มีกล่องย่อยใดที่ตรงตามเงื่อนไข โปรแกรมจะทำการแสดงผลลัพธ์ดังต่อไปนี้

1. แสดงรูปภาพหน้าตัดชิ้นงานที่แสดงตำแหน่งพิกเซลสำหรับวางแผนเส้นทางการเคลื่อนที่ของเครื่องมือตัดในขั้นตอนทำ De-cubing

2. แสดงข้อมูลจำนวนพิกเซลที่เครื่องมือตัดต้องเดินทางทั้งหมดสำหรับกระบวนการ De-cubing (P_C), ข้อมูลจำนวนพิกเซลที่เครื่องมือตัดต้องเดินทางทั้งหมดสำหรับกระบวนการ De-cubing ในแนวแกน X (P_{CX}) และข้อมูลจำนวนพิกเซลที่เครื่องมือตัดต้องเดินทางทั้งหมดสำหรับกระบวนการ De-cubing ในแนวแกน Y (P_{CY}) มีหน่วยเป็นพิกเซล โดยข้อมูลจำนวนพิกเซลที่เครื่องมือตัดต้องทำงานจะส่งผลต่อเวลาในการสร้างชิ้นงานต้นแบบ หากจำนวนพิกเซลมีค่ามาก จะต้องใช้เวลาในการตัดมากตามไปด้วย ทำให้เวลาในการสร้างชิ้นงานต้นแบบมากขึ้น หากจำนวนพิกเซลน้อยกว่าจะทำให้เวลาในการสร้างต้นแบบลดลงได้

3. แสดงข้อมูลจำนวนครั้งที่เครื่องมือต้องตัดชิ้นงานสำหรับกระบวนการ De-cubing ทั้งการตัดในแนวแกน X (N_{CX}) และแนวแกน Y (N_{CY}) มีหน่วยเป็นครั้ง โดยหากข้อมูลจำนวนครั้งที่มีการตัดทำงานมีค่ามากจะส่งผลให้เวลาในการสร้างต้นแบบมีค่ามากขึ้น หากจำนวนครั้งที่มีการตัดทำงานมีค่าน้อยลงจะส่งผลให้เวลาในการสร้างต้นแบบมีค่าลดลงเช่นกัน

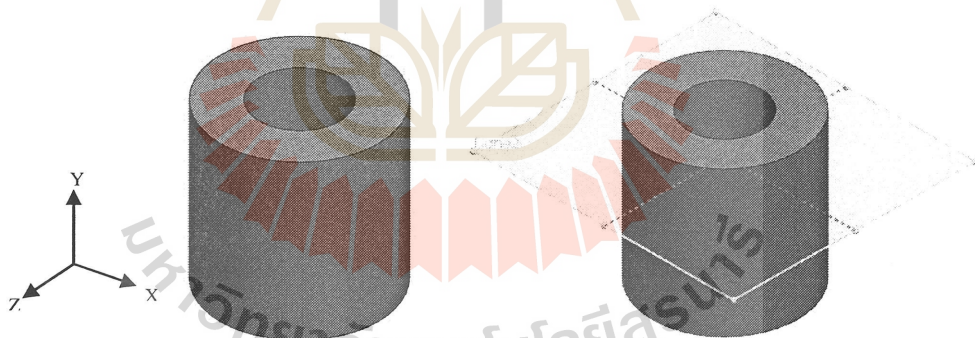
เมื่อทำการแสดงผลลัพธ์ทั้งหมดแล้ว จะสิ้นสุดโปรแกรมสำหรับการวิเคราะห์ตำแหน่งพิกเซลในขั้นตอน De-cubing ของข้อมูลเส้นโครงร่างในชั้นของชิ้นงานต้นแบบที่นำเข้าโดยจะเป็นแบบ Adaptive de-cubing

บทที่ 4

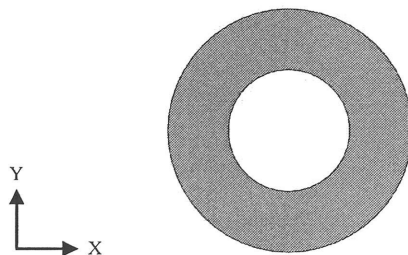
ผลการทดสอบ

บทนี้นำเสนอการดำเนินการทดสอบการวิจัย โดยทำการประมวลผลการวิจัยโดยใช้โปรแกรม MATLAB แสดงผลลัพธ์เป็นรูปภาพทั้ง Uniform de-cubing และ Adaptive de-cubing แสดงจำนวนเส้นทางการเคลื่อนที่ของเครื่องมือตัดและจำนวนพิกเซลที่เครื่องมือตัดต้องทำงาน โดยชิ้นงานสามมิติที่ใช้ในกรณีศึกษาถูกจำลองขึ้นด้วยโปรแกรม SOLIDWORK มีการนำเข้าข้อมูลจะเป็นภาพหน้าตัดของชิ้นงานซึ่งเป็นภาพ 2 มิติ ในการทดสอบได้ทำการกำหนดค่าอัตราส่วนของจำนวนพิกเซลสีดำต่อพิกเซลสีขาว (τ) ในการพิจารณาทั้งหมด 3 ค่า ได้แก่ 0.25, 0.50 และ 0.75 ซึ่งผลการทดสอบสามารถแสดงได้จากกรณีศึกษาดังนี้

4.1 กรณีศึกษาที่ 1



รูปที่ 4.1 ชิ้นงานสามมิติกรณีศึกษาที่ 1



รูปที่ 4.2 แสดงภาพหน้าตัดชิ้นงานสามมิติกรณีศึกษาที่ 1

Name	Value	Min	Max
Bx0	8	8	8
By0	7	7	7
Ck	1	1	1
CutSpaceX	569	569	569
CutSpaceY	671	671	671
Cx	5	5	5
Cy	5	5	5
DC	0	0	0
Ex0	105	105	105
Ey0	104	104	104
HafpiX	57	57	57
HafpiY	56	56	56
MetRes	<341x8 double>	0	105
MetX	<1x33 double>	8	105
MetY	<1x33 double>	7	104
NX	3.0313	3.0313	3.0313
NY	3.0313	3.0313	3.0313
NoRX	5	5	5
NoRY	5	5	5
NopiX	0	0	0
NopiY	0	0	0
PT_gray	<111x111 logical>		
PT_input	<111x111x3 uint8>	0	255

รูปที่ 4.3 แสดงข้อมูลของรูปภาพนำเข้ากรณีศึกษาที่ 1
 ภาพหน้าตัดชิ้นงานสามมิติกรณีศึกษาที่ 1 มีขนาด 111 x 111 พิกเซล
 จากรูปที่ 4.3 จะได้ข้อมูลของกรณีศึกษาที่ 1 ดังนี้

$$Bx_0 = 8$$

$$Ex_0 = 105$$

$$By_0 = 7$$

$$Ey_0 = 104$$

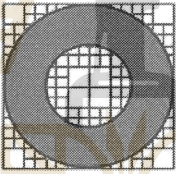
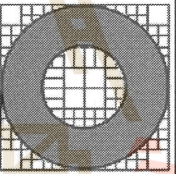
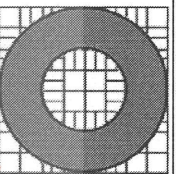
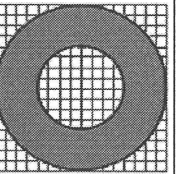
ขนาดของกล่องขอบเขตของรูปภาพเส้น โครงร่างที่นำเข้าจะเท่ากับ

$$SB = (Ex_0 - Bx_0) \times (Ey_0 - By_0)$$

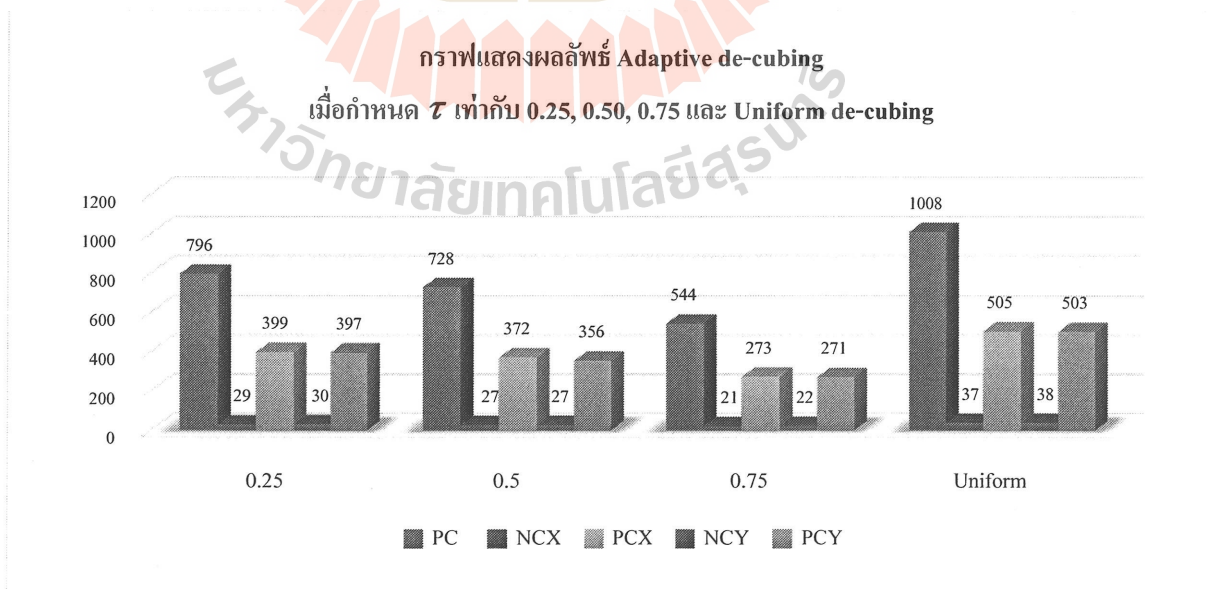
$$SB = (105 - 8) \times (104 - 7) = 97 \times 97 \text{ พิกเซล}$$

ในกรณีศึกษาที่ 1 ได้ทำการทดสอบโปรแกรมแบบ Adaptive de-cubing โดยกำหนดค่า $\tau = 0.25, 0.50$ และ 0.75 และกำหนดค่า $\beta = 10$ พิกเซล และทดสอบโปรแกรมแบบ Uniform de-cubing โดยกำหนดค่า $\beta = 10$ พิกเซล ได้ผลลัพธ์ดังต่อไปนี้

ตารางที่ 4.1 แสดงผลลัพธ์กรณีศึกษาที่ 1

ผลลัพธ์	Threshold (τ)			Uniform
	0.25	0.50	0.75	
จำนวนพิกเซลที่ตัดทั้งหมด (P_C ; พิกเซล)	796	728	544	1008
จำนวนครั้งที่ตัดชิ้นงานในแกน X (N_{CX} ; ครั้ง)	29	27	21	37
จำนวนพิกเซลที่ตัดด้านแกน X (P_{CX} ; พิกเซล)	399	372	273	505
จำนวนครั้งที่ตัดชิ้นงานในแกน Y (N_{CY} ; ครั้ง)	30	27	22	38
จำนวนพิกเซลที่ตัดด้านแกน Y (P_{CY} ; พิกเซล)	397	356	271	503
เส้นทางการเดินทางของ เครื่องมือตัดในกระบวนการ De-cubing				

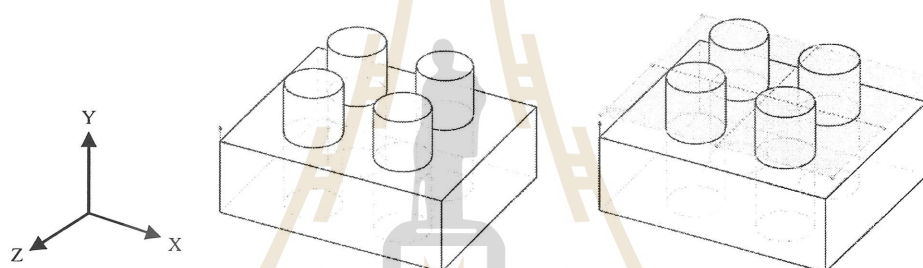
กราฟเปรียบเทียบผลลัพธ์กรณีศึกษาที่ 1



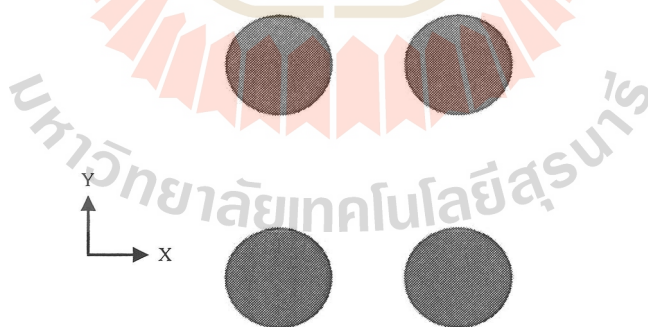
รูปที่ 4.4 แสดงผลลัพธ์เปรียบเทียบในกรณีศึกษาที่ 1

จากผลลัพธ์การศึกษาที่ 1 ที่แสดงในตารางที่ 4.1 และรูปที่ 4.4 จะเห็นได้ว่าการทำ Uniform de-cubing นั้นจะมีค่าจำนวนพิกเซลที่ตัดชิ้นงานทั้งหมด รวมถึงจำนวนครั้งที่เครื่องมือตัดตัดชิ้นงานทั้งหมด มากกว่าการทำ Adaptive de-cubing จากผลลัพธ์ของการทำ Adaptive de-cubing หากกำหนดค่า τ น้อยคือเท่ากับ 0.25 และ 0.50 จะทำให้ค่าจำนวนพิกเซลที่ตัดชิ้นงานทั้งหมด รวมถึงจำนวนครั้งที่เครื่องมือตัดตัดชิ้นงานทั้งหมดมีค่ามากขึ้นตามไปด้วย เมื่อกำหนดค่า τ เท่ากับ 0.75 จะส่งผลให้ค่าจำนวนพิกเซลที่ตัดชิ้นงานทั้งหมด รวมถึงจำนวนครั้งที่เครื่องมือตัดตัดชิ้นงานทั้งหมดมีค่าลดลงมากที่สุด ทำให้ได้ผลลัพธ์น้อยที่สุด ส่งผลให้ระยะเวลาในการสร้างชิ้นงานต้นแบบน้อยที่สุดด้วย

4.2 กรณีศึกษาที่ 2



รูปที่ 4.5 ชิ้นงานสามมิติกรณีศึกษาที่ 2



รูปที่ 4.6 แสดงภาพหน้าตัดชิ้นงานสามมิติกรณีศึกษาที่ 2

Name	Value	Min	Max
Bx0	19	19	19
By0	21	21	21
Ck	1	1	1
CutSpaceX	4151	4151	4151
CutSpaceY	4161	4161	4161
Cx	7	7	7
Cy	7	7	7
DC	0	0	0
Ex0	374	374	374
Ey0	397	397	397
HafpiX	197	197	197
HafpiY	209	209	209
MetRes	<5461x8 double>	0	397
MetX	<1x129 double>	19	374
MetY	<1x129 double>	21	397
NX	2.7734	2.7734	2.7734
NY	2.9375	2.9375	2.9375
NoRX	7	7	7
NoRY	7	7	7
NopiX	0	0	0
NopiY	0	0	0
PT_gray	<389x419 logical>		
PT_input	<389x419x3 uint8>	0	255

รูปที่ 4.7 แสดงข้อมูลของรูปภาพนำเข้ากรณีศึกษาที่ 2

ภาพหน้าตัดชั้นงานสามมิติกรณีศึกษาที่ 2 มีขนาด 389 x 419 พิกเซล

จากรูปที่ 4.7 จะได้ข้อมูลของกรณีศึกษาที่ 2 ดังนี้

$$Bx_0 = 19$$

$$Ex_0 = 374$$

$$By_0 = 21$$

$$Ey_0 = 397$$

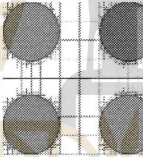
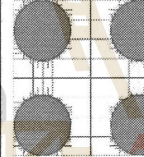
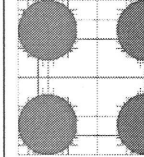
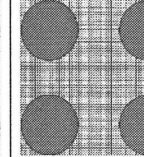
ขนาดของกล่องขอบเขตของรูปภาพเส้น โครงร่างที่นำเข้าจะเท่ากับ

$$SB = (Ex_0 - Bx_0) \times (Ey_0 - By_0)$$

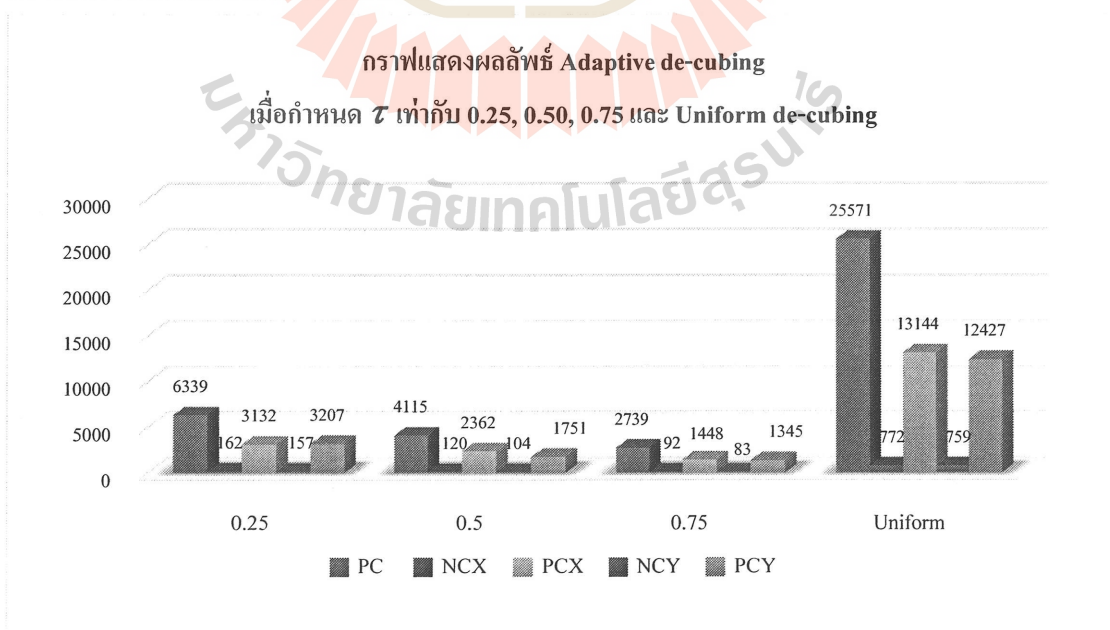
$$SB = (374-19) \times (397-21) = 355 \times 376 \text{ พิกเซล}$$

ในกรณีศึกษาที่ 2 ได้ทำการทดสอบโปรแกรมแบบ Adaptive de-cubing โดยกำหนดค่า $\tau = 0.25, 0.50$ และ 0.75 และกำหนดค่า $\beta = 10$ พิกเซล และทดสอบโปรแกรมแบบ Uniform de-cubing โดยกำหนดค่า $\beta = 10$ พิกเซล ได้ผลลัพธ์ดังต่อไปนี้

ตารางที่ 4.2 แสดงผลลัพธ์กรณีศึกษาที่ 2

ผลลัพธ์	Threshold (τ)			Uniform
	0.25	0.50	0.75	
จำนวนพิกเซลที่ตัดทั้งหมด (P_C ; พิกเซล)	6339	4113	2739	25571
จำนวนครั้งที่ตัดชิ้นงานในแกน X (N_{CX} ; ครั้ง)	162	120	92	772
จำนวนพิกเซลที่ตัดด้านแกน X (P_{CX} ; พิกเซล)	3132	2362	1448	13144
จำนวนครั้งที่ตัดชิ้นงานในแกน Y (N_{CY} ; ครั้ง)	157	104	83	759
จำนวนพิกเซลที่ตัดด้านแกน Y (P_{CY} ; พิกเซล)	3207	1751	1345	12427
เส้นทางการเดินทางของ เครื่องมือตัดในกระบวนการ De-cubing				

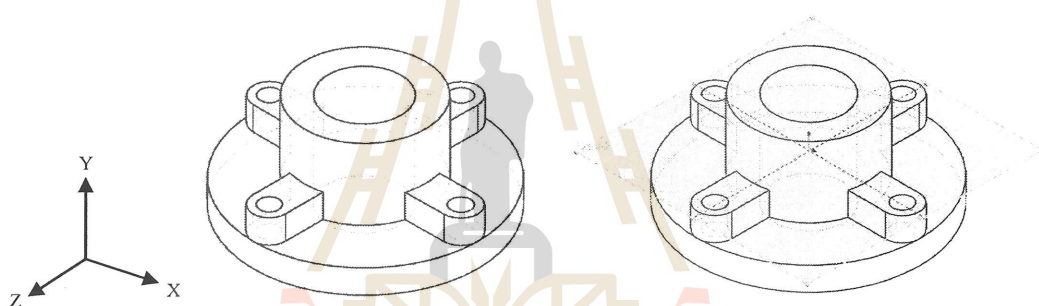
กราฟเปรียบเทียบผลลัพธ์กรณีศึกษาที่ 2



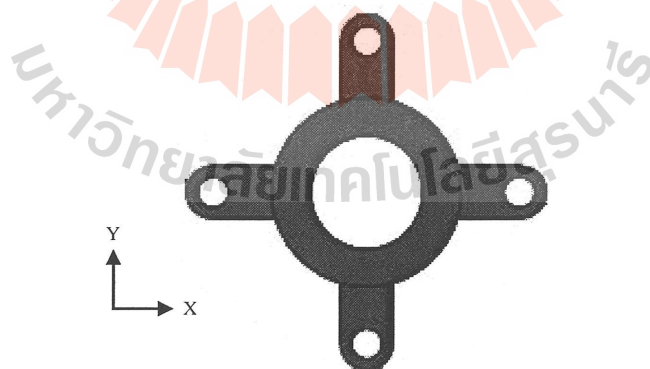
รูปที่ 4.8 แสดงผลลัพธ์เปรียบเทียบในกรณีศึกษาที่ 2

จากผลลัพธ์การศึกษาที่ 2 ที่แสดงในตารางที่ 4.2 และรูปที่ 4.8 จะเห็นได้ว่าการทำ Uniform de-cubing นั้นจะมีค่าจำนวนพิกเซลที่ตัดชิ้นงานทั้งหมด รวมถึงจำนวนครั้งที่เครื่องมือตัดตัดชิ้นงานทั้งหมด มากกว่าการทำ Adaptive de-cubing หลายเท่า จากผลลัพธ์ของการทำ Adaptive de-cubing หากกำหนดค่า τ น้อยคือเท่ากับ 0.25 และ 0.50 จะทำให้ค่าจำนวนพิกเซลที่ตัดชิ้นงานทั้งหมด รวมถึงจำนวนครั้งที่เครื่องมือตัดตัดชิ้นงานทั้งหมดมีค่ามากขึ้นตามไปด้วย เมื่อกำหนดค่า τ เท่ากับ 0.75 จะส่งผลให้ค่าจำนวนพิกเซลที่ตัดชิ้นงานทั้งหมด รวมถึงจำนวนครั้งที่เครื่องมือตัดตัดชิ้นงานทั้งหมดมีค่าลดลงมากที่สุด ทำให้ได้ผลลัพธ์น้อยที่สุด ส่งผลให้ระยะเวลาในการสร้างชิ้นงานต้นแบบน้อยที่สุดด้วย

4.3 กรณีศึกษาที่ 3



รูปที่ 4.9 ชิ้นงานสามมิติกรณีศึกษาที่ 3



รูปที่ 4.10 ภาพหน้าตัดชิ้นงานสามมิติกรณีศึกษาที่ 3

Name ▲	Value	Min	Max
Bx0	6	6	6
By0	12	12	12
Ck	1	1	1
CutSpaceX	2024	2024	2024
CutSpaceY	2020	2020	2020
Cx	6	6	6
Cy	6	6	6
DC	0	0	0
Ex0	169	169	169
Ey0	175	175	175
HafpiX	88	88	88
HafpiY	94	94	94
MetRes	<1365x8 double>	0	175
MetX	<1x65 double>	6	169
MetY	<1x65 double>	12	175
NX	2.5469	2.5469	2.5469
NY	2.5469	2.5469	2.5469
NoRX	6	6	6
NoRY	6	6	6
NopiX	0	0	0
NopiY	0	0	0
☒ PT_gray	<179x192 logical>		
PT_input	<179x192x3 uint8>	0	255

รูปที่ 4.11 แสดงข้อมูลของรูปภาพนำเข้ากรณีศึกษาที่ 3
 ภาพหน้าตัดชิ้นงานสามมิติกรณีศึกษาที่ 3 มีขนาด 179 x 192 พิกเซล
 จากรูปที่ 4.11 จะได้ข้อมูลของกรณีศึกษาที่ 3 ดังนี้

$$Bx_0 = 6$$

$$Ex_0 = 169$$

$$By_0 = 12$$

$$Ey_0 = 175$$

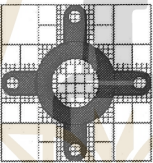
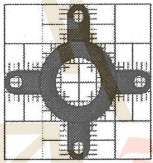
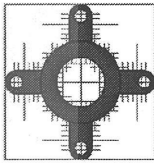
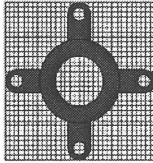
ขนาดของกล่องขอบเขตของรูปภาพเส้นโครงร่างที่นำเข้าจะเท่ากับ

$$SB = (Ex_0 - Bx_0) \times (Ey_0 - By_0)$$

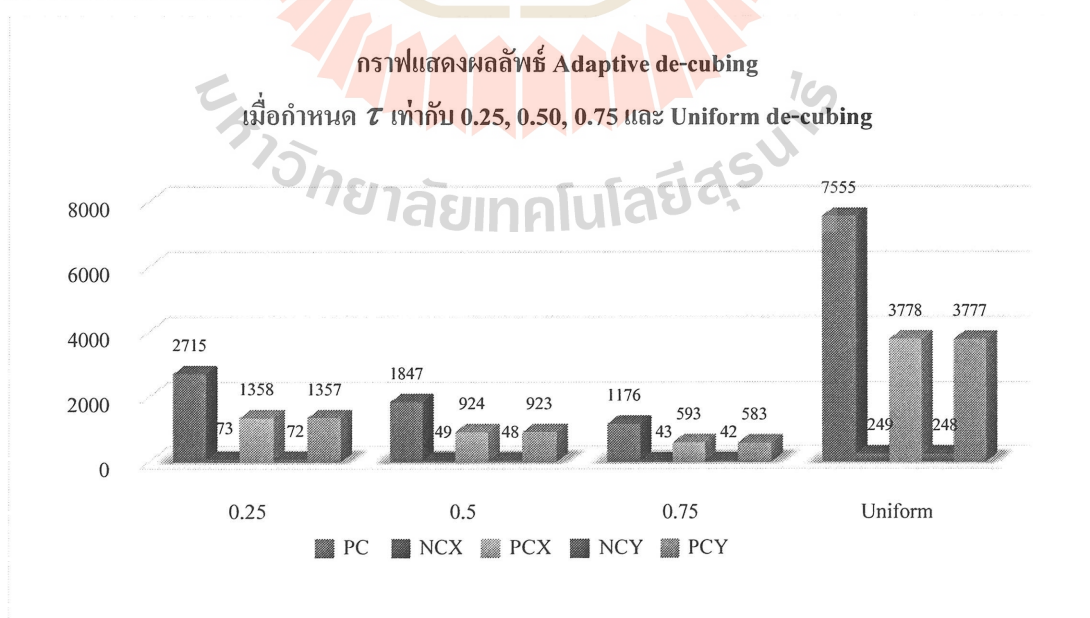
$$SB = (169-6) \times (175-12) = 163 \times 163 \text{ พิกเซล}$$

ในกรณีศึกษาที่ 3 ได้ทำการทดสอบโปรแกรมการแบบ Adaptive de-cubing โดยกำหนดค่า $\tau = 0.25, 0.50$ และ 0.75 และกำหนดค่า $\beta = 10$ พิกเซล และทดสอบโปรแกรมแบบ Uniform de-cubing โดยกำหนดค่า $\beta = 10$ พิกเซล ได้ผลลัพธ์ดังต่อไปนี้

ตารางที่ 4.3 แสดงผลลัพธ์กรณีศึกษาที่ 3

ผลลัพธ์	Threshold (τ)			Uniform
	0.25	0.50	0.75	
จำนวนพิกเซลที่ตัดทั้งหมด (P_C ; พิกเซล)	2715	1847	1176	7555
จำนวนครั้งที่ตัดชิ้นงานในแกน X (N_{CX} ; ครั้ง)	73	49	43	249
จำนวนพิกเซลที่ตัดด้านแกน X (P_{CX} ; พิกเซล)	1358	924	593	3778
จำนวนครั้งที่ตัดชิ้นงานในแกน Y (N_{CY} ; ครั้ง)	72	48	42	248
จำนวนพิกเซลที่ตัดด้านแกน Y (P_{CY} ; พิกเซล)	1357	923	583	3777
เส้นทางการเดินทางของ เครื่องมือตัดในกระบวนการ De-cubing				

กราฟเปรียบเทียบผลลัพธ์กรณีศึกษาที่ 3



รูปที่ 4.12 แสดงผลลัพธ์เปรียบเทียบในกรณีศึกษาที่ 3

จากผลลัพธ์กรณีศึกษาที่ 3 ที่แสดงในตารางที่ 4.3 และรูปที่ 4.12 จะเห็นได้ว่าการทำ Uniform de-cubing นั้นจะมีค่าจำนวนพิกเซลที่ตัดชิ้นงานทั้งหมด รวมถึงจำนวนครั้งที่เครื่องมือตัดตัดชิ้นงานทั้งหมด มากกว่าการทำ Adaptive de-cubing หลายเท่า จากผลลัพธ์ของการทำ Adaptive de-cubing หากกำหนดค่า τ น้อยคือเท่ากับ 0.25 และ 0.50 จะทำให้ค่าจำนวนพิกเซลที่ตัดชิ้นงานทั้งหมด รวมถึงจำนวนครั้งที่เครื่องมือตัดตัดชิ้นงานทั้งหมดมีค่ามากขึ้นตามไปด้วย เมื่อกำหนดค่า τ เท่ากับ 0.75 จะส่งผลให้ค่าจำนวนพิกเซลที่ตัดชิ้นงานทั้งหมด รวมถึงจำนวนครั้งที่เครื่องมือตัดตัดชิ้นงานทั้งหมดมีค่าลดลงมากที่สุด ทำให้ได้ผลลัพธ์น้อยที่สุด ส่งผลให้ระยะเวลาในการสร้างชิ้นงานต้นแบบน้อยที่สุดด้วย

4.4 กรณีศึกษาที่ 4



รูปที่ 4.13 แสดงภาพหน้าตัดชิ้นงานสามมิติกรณีศึกษาที่ 4

Name	Value	Min	Max
Bx0	14	14	14
By0	10	10	10
Ck	1	1	1
CutSpaceX	2443	2443	2443
CutSpaceY	2159	2159	2159
Cx	6	6	6
Cy	6	6	6
DC	0	0	0
Ex0	235	235	235
Ey0	252	252	252
HafpiX	125	125	125
HafpiY	131	131	131
MetRes	<1365x8 double>	0	252
MetX	<1x65 double>	14	235
MetY	<1x65 double>	10	252
NX	3.4531	3.4531	3.4531
NY	3.7813	3.7813	3.7813
NoRX	6	6	6
NoRY	6	6	6
NopiX	0	0	0
NopiY	0	0	0
PT_gray	<258x263 logical>		
PT_input	<258x263x3 uint8>	0	255

รูปที่ 4.14 แสดงข้อมูลของรูปภาพนำเข้ากรณีศึกษาที่ 4

ภาพหน้าตัดชิ้นงานสามมิติกรณีศึกษาที่ 4 มีขนาด 263 x 258 พิกเซล

จากรูปที่ 4.14 จะได้ข้อมูลของกรณีศึกษาที่ 4 ดังนี้

$$Bx_0 = 14$$

$$Ex_0 = 235$$

$$By_0 = 10$$

$$Ey_0 = 252$$

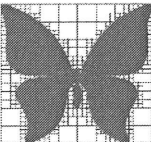
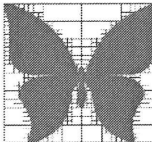
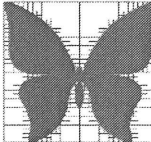
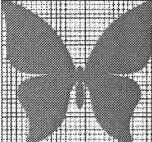
ขนาดของกล่องขอบเขตของรูปภาพเส้นโครงร่างที่นำเข้าจะเท่ากับ

$$SB = (Ex_0 - Bx_0) \times (Ey_0 - By_0)$$

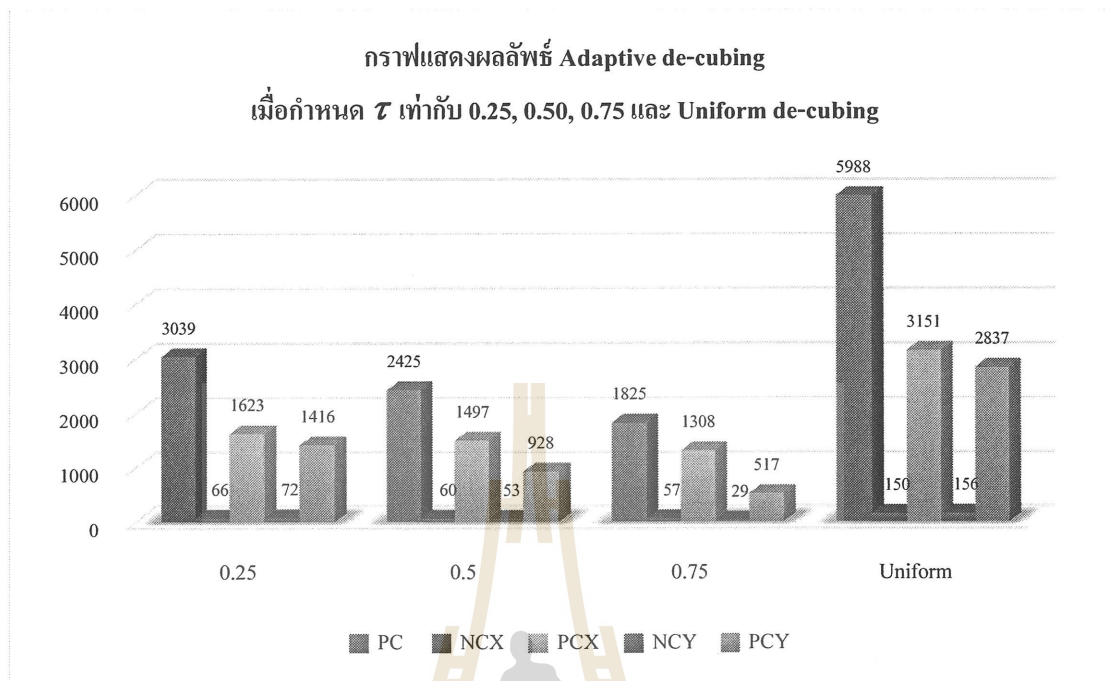
$$SB = (235-14) \times (252-10) = 221 \times 242 \text{ พิกเซล}$$

ในกรณีศึกษาที่ 4 ได้ทำการทดสอบโปรแกรมการแบบ Adaptive de-cubing โดยกำหนดค่า $\tau = 0.25, 0.50$ และ 0.75 และกำหนดค่า $\beta = 10$ พิกเซล และทดสอบโปรแกรมแบบ Uniform de-cubing โดยกำหนดค่า $\beta = 10$ พิกเซล ได้ผลลัพธ์ดังต่อไปนี้

ตารางที่ 4.4 แสดงผลลัพธ์กรณีศึกษาที่ 4

ผลลัพธ์	Threshold (τ)			Uniform
	0.25	0.50	0.75	
จำนวนพิกเซลที่ตัดทั้งหมด (P_C ; พิกเซล)	3039	2425	1825	5988
จำนวนครั้งที่ตัดชิ้นงานในแกน X (N_{CX} ; ครั้ง)	66	60	57	150
จำนวนพิกเซลที่ตัดด้านแกน X (P_{CX} ; พิกเซล)	1623	1497	1308	3151
จำนวนครั้งที่ตัดชิ้นงานในแกน Y (N_{CY} ; ครั้ง)	72	53	29	156
จำนวนพิกเซลที่ตัดด้านแกน Y (P_{CY} ; พิกเซล)	1416	928	517	2837
เส้นทางการเดินทางของ เครื่องมือตัดในกระบวนการ De-cubing				

กราฟเปรียบเทียบผลลัพธ์กรณีศึกษาที่ 4



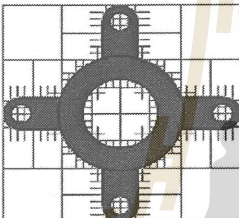
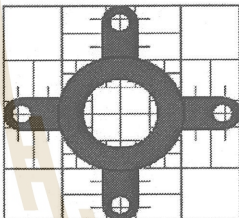
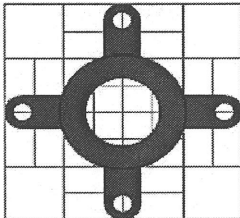
รูปที่ 4.15 แสดงผลลัพธ์เปรียบเทียบในกรณีศึกษาที่ 4

จากผลลัพธ์กรณีศึกษาที่ 4 ที่แสดงในตารางที่ 4.4 และรูปที่ 4.15 จะเห็นได้ว่าการทำ Uniform de-cubing นั้นจะมีค่าจำนวนพิกเซลที่ตัดชิ้นงานทั้งหมด รวมถึงจำนวนครั้งที่เครื่องมือตัดตัดชิ้นงานทั้งหมด มากกว่าการทำ Adaptive de-cubing ถึงหนึ่งเท่า จากผลลัพธ์ของการทำ Adaptive de-cubing หากกำหนดค่า τ น้อยคือเท่ากับ 0.25 และ 0.50 จะทำให้ค่าจำนวนพิกเซลที่ตัดชิ้นงานทั้งหมด รวมถึงจำนวนครั้งที่เครื่องมือตัดตัดชิ้นงานทั้งหมดมีค่ามากขึ้นตามไปด้วย เมื่อกำหนดค่า τ เท่ากับ 0.75 จะส่งผลให้ค่าจำนวนพิกเซลที่ตัดชิ้นงานทั้งหมด รวมถึงจำนวนครั้งที่เครื่องมือตัดตัดชิ้นงานทั้งหมดมีค่าลดลงมากที่สุด ทำให้ได้ผลลัพธ์น้อยที่สุด ส่งผลให้ระยะเวลาในการสร้างชิ้นงานต้นแบบน้อยที่สุดด้วย

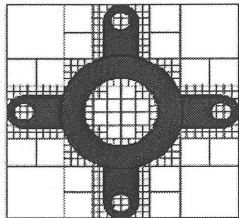
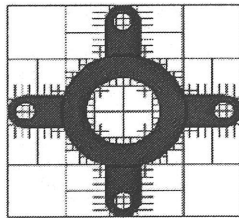
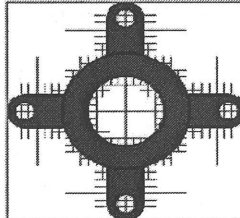
4.5 ผลลัพธ์เมื่อเปลี่ยนแปลงค่า β และ τ

จากตารางที่ 4.5 และ 4.6 จะเห็นได้ว่าเมื่อเราเปลี่ยนแปลงค่า β และ τ โดยถ้ากำหนดค่า β และ τ มากขึ้น จะส่งผลให้ชิ้นงานต้นแบบที่ได้มีความหนาของผิวชิ้นงานมากขึ้น ดังนั้นหากต้องการชิ้นงานที่มีความละเอียดมาก จึงต้องทำการกำหนดค่า β และ τ ให้น้อยลงนั่นเอง

ตารางที่ 4.5 แสดงผลลัพธ์เมื่อกำหนดค่า $\tau = 0.50$, $\beta = 10, 20$ และ 30 ตามลำดับ

ผลลัพธ์	β		
	10	20	30
จำนวนพิกเซลที่ตัดทั้งหมด (P_c ; พิกเซล)	1847	1198	888
จำนวนครั้งที่ตัดชิ้นงานทั้งหมด (N_c ; ครั้ง)	97	48	24
ตำแหน่งการเดินทางของเครื่องมือตัดในขั้นตอน De-cubing			

ตารางที่ 4.6 แสดงผลลัพธ์เมื่อกำหนดค่า $\beta = 10$, $\tau = 0.25, 0.50$ และ 0.75 ตามลำดับ

ผลลัพธ์	τ		
	0.25	0.50	0.75
จำนวนพิกเซลที่ตัดทั้งหมด (P_c ; พิกเซล)	2715	1847	1176
จำนวนครั้งที่ตัดชิ้นงานทั้งหมด (N_c ; ครั้ง)	145	97	85
ตำแหน่งการเดินทางของเครื่องมือตัดในขั้นตอน De-cubing			

บทที่ 5

สรุปและข้อเสนอแนะ

บทนี้เสนอการสรุปผลของงานวิจัย ผลจากการศึกษาที่ได้จากโปรแกรมคือภาพแสดงเส้นทางการทำ Uniform de-cubing และ Adaptive de-cubing รวมถึงความสัมพันธ์ของจำนวนจำนวนเส้นทางการเคลื่อนที่ของเครื่องมือตัดและจำนวนพิกเซลที่เครื่องมือตัดต้องทำงาน เมื่อทำการกำหนดค่าอัตราส่วนของจำนวนพิกเซลสีดำต่อพิกเซลสีขาว (τ) ที่แตกต่างกัน สำหรับใช้เป็นข้อมูลในการวางแผนการกระบวนการทำ De-cubing ในการสร้างชิ้นงานต้นแบบรวดเร็วด้วยเทคนิค Laminate Object Manufacturing (LOM) และข้อเสนอแนะสำหรับงานวิจัยต่อไป

5.1 สรุปผลงานวิจัย

5.1.1 การวิเคราะห์ผลการทดสอบระหว่าง Uniform de-cubing กับ Adaptive de-cubing

จากผลการทดสอบพบว่าการทำ Adaptive de-cubing จะให้ค่าจำนวนพิกเซลที่ตัดชิ้นงานทั้งหมด รวมถึงจำนวนครั้งที่เครื่องมือตัดตัดชิ้นงานทั้งหมดน้อยกว่า Uniform de-cubing โดยมีการกำหนดค่าจำนวนพิกเซลที่เล็กที่สุดของกล่องขอบเขตย่อยที่จะพิจารณาต่อตามเงื่อนไขที่กำหนด(β) ให้เท่ากัน พบว่าหากค่า β มีค่ามาก จะส่งผลให้จำนวนพิกเซลที่ตัดชิ้นงานทั้งหมด รวมถึงจำนวนครั้งที่เครื่องมือตัดตัดชิ้นงานทั้งหมดน้อยลง ชิ้นงานต้นแบบที่ได้จะมีความละเอียดลดน้อยลงด้วยความสัมพันธ์ระหว่าง β , จำนวนพิกเซลที่ตัดชิ้นงานทั้งหมด และจำนวนครั้งที่ตัดชิ้นงานสามารถแสดงได้ดังสมการที่ (5.1)

$$\beta \propto \frac{1}{P_c} \propto \frac{1}{N_c} \quad (5.1)$$

เมื่อ P_c = จำนวนพิกเซลที่เครื่องมือตัดต้องตัดชิ้นงานทั้งหมด
 N_c = จำนวนครั้งที่เครื่องมือตัดต้องตัดชิ้นงานทั้งหมด

5.1.2 การวิเคราะห์ผลการทดสอบ Adaptive de-cubing เมื่อกำหนดค่า τ เท่ากับ 0.25, 0.50 และ 0.75

จากผลการทดสอบพบว่าหากกำหนดค่าอัตราส่วนของจำนวนพิกเซลสีดำต่อพิกเซลสีขาว (τ) เท่ากับ 0.25 จะส่งผลให้ค่าจำนวนพิกเซลที่ตัดชิ้นงานทั้งหมดและจำนวนครั้งที่ตัดชิ้นงานมีค่ามาก ทำให้ต้องใช้เวลาในการสร้างชิ้นงานต้นแบบมากแต่ชิ้นงานต้นแบบที่ได้จะมีความละเอียดสูง หากทำการกำหนดค่า τ ให้เพิ่มมากขึ้นจะทำให้ค่าจำนวนพิกเซลที่ตัดชิ้นงานทั้งหมดและจำนวนครั้งที่ตัดชิ้นงานลดลง ทำให้สามารถลดระยะเวลาในการสร้างชิ้นงานต้นแบบให้น้อยลง แต่ชิ้นงานต้นแบบที่ได้จะมีความละเอียดลดลงไปอีกด้วย ความสัมพันธ์ของค่า τ , จำนวนพิกเซลที่ตัดชิ้นงานทั้งหมดและจำนวนครั้งที่ตัดชิ้นงานสามารถแสดงได้ดังสมการที่ (5.2)

$$\tau \propto \frac{1}{P_c} \propto \frac{1}{N_c} \quad (5.2)$$

ในการกำหนดค่า β และค่า τ สามารถเลือกปรับค่าได้เพื่อให้ตรงตามวัตถุประสงค์ของการสร้างชิ้นงานต้นแบบ หากต้องการต้นแบบที่มีความละเอียดสูงต้องทำการปรับค่า β และค่า τ ให้น้อยลง แต่การสร้างต้นแบบก็ต้องใช้เวลาเพิ่มขึ้นด้วย หากต้องการต้นแบบที่ไม่ต้องละเอียดมากแต่ต้องการความรวดเร็วในการสร้างต้นแบบ ต้องทำการปรับค่า β และค่า τ ให้สูงขึ้น โปรแกรมการวิเคราะห์การสร้างเส้นทางการทำ De-cubing นี้สามารถนำไปประยุกต์ใช้ให้ตรงกับความต้องการได้อย่างหลากหลายขึ้นอยู่กับวัตถุประสงค์และการใช้งาน

5.2 ข้อเสนอแนะในการทำงานวิจัยต่อไป

1. นำข้อมูลตำแหน่งเส้นทางการทำ De-cubing ไปใช้กับการสร้างชิ้นงานต้นแบบจริง
2. สร้างโปรแกรมที่สามารถวิเคราะห์ได้เป็นอัตโนมัติ
3. พัฒนาโปรแกรมที่มีความแม่นยำมากยิ่งขึ้น
4. พัฒนาโปรแกรมให้สะดวกต่อการใช้งานมากขึ้น

รายการอ้างอิง

- Charles L. Thomas. (1996). Automating Sheet-Based Fabrication: **The Conveyed-Adherent™ Process**. Ennex Corp.
- Chris Solomon and Toby Breckon. (2011). **Fundamentals of Digital Image Processing (A practical approach with example in MATLAB)**, 1st ed. West Sussex: John Wiley & Sons, Ltd.,
- Cho, I., Lee, K., Choi, W., & Song, Y.-A. (2000). Development of a new sheet deposition type rapid prototyping system. **International Journal of Machine Tools and Manufacture**, 40(12), 1813-1829.
- C K Chua, K F Leong and C S Lim. (2009). **Rapid prototyping: principles and applications**. Singapore: World Scientific Publishing.
- Jim Flowers and Matt Moniz. (2002). Rapid Prototyping in Technology Education. **The Technology Teacher**, November.
- J. Hur, K. Lee. (1996). Efficient algorithm for automatic support structure generation in layered manufacturing, in: **Proceedings of the 1996 ASME Design Engineering Technical Conference and Computers in Engineering Conference**, Irvine, USA.
- Ludmila Novakova-Marcincinova and Ivan Kuric. (2012). Basic and Advanced Materials for Fused Deposition Modeling Rapid Prototyping Technology. **Manuf. And Ind. Eng.**, 11(1).
- Majd Al Mardini, DDS,^a Carlo Ercoli, DDS,^b and Gerald N. Graser, DDS, MS^c. (2005). **A technique to produce a mirror-image wax pattern of an ear using rapid prototyping technology**. August.
- Narendra B. Dahotre and Sandip P. Harimkar. (2008). **Laser-Based Rapid Prototyping Processes**, 3rd ed. Springer US: Springer, pp 353-411.
- Ovidiu MOLDOVAN. (2011). Rapid Prototyping for Robotics Applications. **Nonconventional Technology Review-no. 1/2011**.

- Sarinya Putthawonga, Pisut Koomsap and Natthavika Chansri. (2014). Image-based adaptive crosshatch toolpath generation for laminated object manufacturing, **Virtual and Physical Prototyping**, Vol. 9, No. 4, 233–249.
- Tinku Acharya and Ajoy K. Ray. (2005). **Image Processing Principles and Applications**, United States of America: John Wiley & Sons, Ltd.,
- Y.S. Liao, L.C. Chiu, Y.Y. Chiu. (2003). A new approach of online waste removal process for laminated object manufacturing (LOM). **Journal of Materials Processing Technology**, 136–140.
- Y.Y. Chiu, Y.S. Liao and C.C. Hou. (2015). The Study of On-line Waste Material Removal Procedures for Bridge Laminated Object Manufacturing (LOM) Process. **Journal of Materials Processing Technology**.
- ZHANG Yu and LIU Hongwu. (2015). Application of Rapid Prototyping Technology in Die Making of Diesel Engine. Volume 14, Number S1, June.
- ธนาคาร เป้าทอง. (2555). การระบุความสัมพันธ์ของเส้นโครงร่างเพื่อวางแผนเส้นทางการเคลื่อนที่ของเครื่องมือในกรรมวิธีการสร้างต้นแบบรวดเร็ว.
- ปภากร พิทยชาวล, ธนาคาร เป้าทอง. (2556). การวางแผนเส้นทางการเคลื่อนที่ของเครื่องมือในเทคโนโลยีการสร้างต้นแบบรวดเร็วจากความสัมพันธ์ของเส้นโครงร่าง A Tool Path Planning of Rapid Prototyping Technology Base on a Contour Relationship.
- วิบูลย์ ตั้งวัชรณกุล. (2011). เทคโนโลยีการสร้างต้นแบบรวดเร็ว (Rapid prototyping technologies). April-May, Vol.38 No.216.
- ฉัตรชัย จันทร์เด่นดวง. (2546). เทคโนโลยีการสร้างต้นแบบรวดเร็ว Rapid Prototyping Technology. ปทุมธานี: ศูนย์เทคโนโลยีโลหะและวัสดุแห่งชาติ (เอ็มเทค).
- โสธรญา แจ่มการ, กนต์ธร ชำนิประศาสน์. (2559). การใช้ MATLAB สำหรับงานทางวิศวกรรม (ฉบับปรับปรุงครั้งที่ 2)
- สุธรรม ศรีเกษมและคณะ, MATLAB เพื่อการแก้ปัญหาทางวิศวกรรม. ปทุมธานี: มหาวิทยาลัยรังสิต.



1.1 โมดูลหลักของการทำงาน (Main Module)

ชื่อตัวแปร	ชนิดของตัวแปร	หน้าที่การทำงาน
PT_input()	อาร์เรย์ 3 มิติ ของ Integer	เป็นตัวแปรเก็บรหัสสีในระบบ RGB ของรูปภาพที่ต้องการวิเคราะห์
PT_gray()	อาร์เรย์ 2 มิติ ของ Integer	เป็นตัวแปรเก็บรหัสสีของรูปภาพที่โดนเปลี่ยนเป็นภาพขาว-ดำ สีขาว เก็บค่าเป็น 1 และสีดำ เก็บค่าเป็น 0
ResultX	Integer	เป็นตัวแปรสำหรับเก็บค่าผลรวมการตัดชิ้นงานในแนวตั้ง
ResultY	Integer	เป็นตัวแปรสำหรับเก็บค่าผลรวมการตัดชิ้นงานในแนวนอน
CutSpaceX	Integer	เป็นตัวแปรสำหรับเก็บค่าผลรวมจำนวนพิกเซลที่ถูกตัดเมื่อตัดแนวตั้ง
CutSpaceY	Integer	เป็นตัวแปรสำหรับเก็บค่าผลรวมจำนวนพิกเซลที่ถูกตัดเมื่อตัดแนวนอน
max_rows	Integer	เป็นตัวแปรสำหรับเก็บค่าจำนวนพิกเซลมากที่สุด
max_columns	Integer	เป็นตัวแปรสำหรับเก็บค่าจำนวนพิกเซลมากที่สุด
Bx0, By0	Integer	เป็นตัวแปรสำหรับเก็บค่าพิกัดของพิกเซลเริ่มต้นของแนวแกน x และ y
Ex0, Ey0	Integer	เป็นตัวแปรสำหรับเก็บค่าพิกัดของพิกเซลสุดท้ายของแนวแกน x และ y
X0, Y0	Integer	เป็นตัวแปรสำหรับเก็บค่าพิกัดของพิกเซลเริ่มต้นของแนวแกน x และ y ซึ่งเปลี่ยนแปลงตามรอบการตัด
X1, Y1	Integer	เป็นตัวแปรสำหรับเก็บค่าพิกัดของพิกเซลสุดท้ายของแนวแกน x และ y ซึ่งเปลี่ยนแปลงตามรอบการตัด
N	Integer	เป็นตัวแปรสำหรับเก็บค่าจำนวนช่องที่โดนแบ่งในแต่ละครั้ง

1.2 ฟังก์ชันการทำงานในหาการหาเส้นตัดแนวแกนนอน (X) และแกนตั้ง (Y)

ตัวแปรสำหรับบันทึกค่าที่ถูกส่งจากโมดูลหลัก

ชื่อตัวแปร	ชนิดของตัวแปร	หน้าที่การทำงาน
PT	อาร์เรย์ 2 มิติ ของ Integer	เป็นตัวแปรสำหรับเก็บค่ารหัสสีของรูปภาพ
Bx, By	Integer	เป็นตัวแปรสำหรับเก็บค่าพิกเซลเริ่มต้นของแนวแกน x และ y
Ex, Ey	Integer	เป็นตัวแปรสำหรับเก็บค่าพิกเซลสุดท้ายของแนวแกน x และ y

ตัวแปรที่ถูกส่งกลับไปยังโมดูลหลักและหน้าต่างแสดงผล (Command window)

ชื่อตัวแปร	ชนิดของตัวแปร	หน้าที่การทำงาน
R	Integer	เป็นตัวแปรสำหรับเก็บค่าผลรวมของจำนวนครั้งตัดชิ้นงาน
CutSpace	Integer	เป็นตัวแปรสำหรับเก็บค่าผลรวมจำนวนพิกเซลที่โดนตัดเมื่อตัดแนวตั้ง
pixelCutX, pixelCutY	Integer	เป็นตัวแปรสำหรับเก็บค่าพิกเซลในแกน x และ y ที่โดนตัด
cntX,cntY	Integer	เป็นตัวแปรสำหรับเก็บค่าจำนวนครั้งในการตัดส่วนที่เป็นของเสียทั้งแนวตั้งและแนวนอน (De-cubing process)



ภาคผนวก ข

โปรแกรม MATLAB

ข.1 โปรแกรม MATLAB วิเคราะห์การสร้าง Uniform de-cubing

```

PT_input = imread('Ex1.jpg');
PT_gray = im2bw(PT_input,0.7);
[max_rows,max_columns] = size(PT_gray);

for i=1:max_rows
    Ck=0;

    for j=1:max_columns
        if PT_gray(i,j) == 0;
            Bx0 = i;
            x1=i;
            Ck = 1;
            break
        end
    end
    if Ck ==1;
        break
    end
end
for i=max_rows:-1:1
    Ck=0;
    for j=1:max_columns
        if PT_gray(i,j) == 0;
            Ex0 = i;
            x2=i;
            Ck=1;
            break
        end
    end
end

```

```

if Ck ==1;
    break
end
end
end
for i=1:max_columns
    Ck=0;
    for j=1:max_rows
        if PT_gray(j,i) == 0;
            By0 = i;
            y1=i;
            Ck=1;
            break
        end
    end
    if Ck ==1;
        break
    end
    end
    for i=max_columns:-1:1
        Ck=0;
        for j=1:max_rows
            if PT_gray(j,i) == 0;
                Ey0 = i;
                y2=i;
                Ck=1;
                Break
            end
        end
    end
end

```

```

if Ck ==1;
    break
end
end
end
PT_input(Bx0,By0:Ey0,1) = 255;
PT_input(Bx0,By0:Ey0,2) = 0;
PT_input(Bx0,By0:Ey0,3) = 0;

PT_input(Ex0,By0:Ey0,1) = 255;
PT_input(Ex0,By0:Ey0,2) = 0;
PT_input(Ex0,By0:Ey0,3) = 0;

PT_input(Bx0:Ex0,By0,1) = 255;
PT_input(Bx0:Ex0,By0,2) = 0;
PT_input(Bx0:Ex0,By0,3) = 0;

PT_input(Bx0:Ex0,Ey0,1) = 255;
PT_input(Bx0:Ex0,Ey0,2) = 0;
PT_input(Bx0:Ex0,Ey0,3) = 0;
image(PT_input)
cntX = 4;
cntY = 4;
NoRX = 0;
NoRY =0;
NX = Ex0-Bx0;
NY = Ey0-By0;
ResultX = 1; ResultY = 1; CutSpaceX=0; CutSpaceY=0;

```

```
HafpiX = round(Bx0 + (Ex0-Bx0)/2);
```

```
MetRes(1,1) = HafpiX;
```

```
MetRes(1,3) = 1;
```

```
HafpiY = round(By0 + (Ey0-By0)/2);
```

```
MetRes(1,2) = HafpiY;
```

```
MetRes(1,4) = 1;
```

```
MetRes(1,5) = Bx0;
```

```
MetRes(1,6) = Ex0;
```

```
MetRes(1,7) = By0;
```

```
MetRes(1,8) = Ey0;
```

```
for y = By0 : Ey0
```

```
    f = 0;
```

```
    if PT_gray(HafpiX,y)~= 0
```

```
        PT_input(HafpiX,y,1) = 255;
```

```
        PT_input(HafpiX,y,2) = 0;
```

```
        PT_input(HafpiX,y,3) = 0;
```

```
        f = 1;
```

```
    end
```

```
end
```

```
image(PT_input)
```

```
if f == 1
```

```
    cntX = cntX + 1;
```

```
end
```

```
for x =Bx0 : Ex0
```

```
    f = 0;
```



```

    if PT_gray(x,HafpiY)~= 0
        PT_input(x,HafpiY,1) = 255;
        PT_input(x,HafpiY,2) = 0;
        PT_input(x,HafpiY,3) = 0;
        f = 1;
    end
end
if f == 1
    cntY = cntY + 1;
end
MetX = [Bx0,Ex0];
MetY = [By0,Ey0];
Cx =0; Cy=0;Px=1; Py=1;
depX = Ex0-Bx0;
depY = Ey0-By0;
count=1;
while and((depX>=4),(depY>=4))
    Cx = Cx+1;
    Cy = Cy+1;
    [MetX,MetY]= Uniformdivide(PT_gray,Bx0,Ex0,By0,Ey0,Cx,Cy);
    depX = MetX(2)- MetX(1);
    depY = MetY(2)- MetY(1);
    if and((depX>=4),(depY>=4))
        sx = size(MetX) ; sy =size(MetY);
        for Px=1 : sx(1,2)-1
            for Py=1 : sy(1,2)-1
                count = count+1;
            end
        end
    end
end

```

```

X0 = MetX(Px);
X1 = MetX(Px+1);
Y0 = MetY(Py);
Y1 = MetY(Py+1);
[ResultX,CutSpaceX,NopiX,DC]=UniformX1(PT_gray,X0,X1,Y0,Y1,ResultX,CutSpaceX);
if NopiX ~= 0;
[ResultY,CutSpaceY,NopiY,DC]=UniformY1(PT_gray,X0,X1,Y0,Y1,ResultY,CutSpaceY);
if NopiY ~= 0
MetRes(count,1) = NopiX;MetRes(count,2) = NopiY;
MetRes(count,3) = DC;MetRes(count,4) = DC;
for y = Y0 : Y1
f=0;
if PT_gray(NopiX,y)~= 0
PT_input(NopiX,y,1) = 255;
PT_input(NopiX,y,2) = 0;
PT_input(NopiX,y,3) = 0;
f=1;
end
end
if f==1
cntX = cntX + 1;
end
for x = X0 : X1
f=0;
if PT_gray(x,NopiY)~= 0
PT_input(x,NopiY,1) = 255;
PT_input(x,NopiY,2) = 0;
PT_input(x,NopiY,3) = 0;

```

```

        f=1;
    end
end
if f==1
    cntY = cntY + 1;
end
    image(PT_input)
end
if NopiY == 0
    MetRes(count,4) = 0;
end
if NopiX == 0
    MetRes(count,3) = 0;
end
    MetRes(count,5) = X0;
    MetRes(count,6) = X1;
    MetRes(count,7) = Y0;
    MetRes(count,8) = Y1;
end
end
end
end
    TotalCutSpace = CutSpaceX + CutSpaceY;
fprintf('จำนวนพิกเซลที่ตัดทั้งหมด (De-cubing process) : %d พิกเซล\n',TotalCutSpace)
fprintf('จำนวนรอบการเดินทางของเครื่องมือตัดแนวตั้ง(ตัดแกนX) : %d ครั้ง\nจำนวน
พิกเซลที่ถูกตัดแนวตั้ง(De-cubing)ตัดแกนX : %d พิกเซล\n',cntX,CutSpaceX)
fprintf('จำนวนรอบการเดินทางของเครื่องมือตัดแนวนอน(ตัดแกนY) : %d ครั้ง\nจำนวน
พิกเซลที่ถูกตัดแนวนอน(De-cubing)ตัดแกนY : %d พิกเซล\n',cntY,CutSpaceY)

```

ข.1.1 ฟังก์ชัน Compare X วิเคราะห์การสร้าง Uniform de-cubing ในแกน X

```
function [R,CutSpace,pixelCutX,dc] = UniformX(PT,Bx,Ex,By,Ey,countX,CutSpace)

    dc = 0;
    for j=By:Ey
        for i=Bx:Ex
            if PT(i,j)==0
                end
            end
        end
        s = (Ex-Bx)+1;
        if (Ex-Bx) >= 10
            dc = 1;
            R = countX+ 1;
            pixelCutX = Bx + floor(s/2);
            for j=By:Ey
                if PT(pixelCutX,j)== 1
                    CutSpace = CutSpace + 1;
                end
            end
            break
        else
            CutSpace = CutSpace;
            pixelCutX = 0;
            R = countX;
        end
    end
end
```


ข.1.2 ฟังก์ชัน Compare Y วิเคราะห์การสร้าง Uniform de-cubing ในแกน Y

```

function [R,CutSpace,pixelCutY,dc] = UniformY(PT,Bx,Ex,By,Ey,countY,CutSpace)

dc = 0;
for i=Bx:Ex
    for j=By:Ey
        if PT(i,j)==0
            end
        end
    end
    s = (Ey-By)+1;
    if (Ey-By)>= 10
        dc = 1;
        R = countY + 1;
        pixelCutY = By + floor(s/2);
        for i=Bx:Ex
            if PT(i,pixelCutY) == 1
                CutSpace = CutSpace + 1;
            end
        end
        break
    else
        CutSpace = CutSpace;
        pixelCutY = 0;
        R = countY;
    end
end
end

```

ข.1.3 ฟังก์ชัน divide วิเคราะห์และเก็บค่าการแบ่งกล่องย่อยในการวนแต่ละรอบ

```
function [MetX,MetY] = Uniformdivide(PT,Bx,Ex,By,Ey,X,Y)
r=2; n=2;
HafX = (Ex-Bx)/2^X;
HafY = (Ey-By)/2^Y;
MetX(1) = Bx;
MetY(1) = By;
for r=2 : 1 + 2^X
    MetX(r) = round(Bx + (HafX*(r-1)));
for n=2 : 1 + 2^Y
    MetY(n) = round(By + (HafY*(n-1)));
end
end
end
```

ข.2 โปรแกรม MATLAB วิเคราะห์การสร้าง Adaptive de-cubing

```

PT_input = imread ('Ex1.jpg');
PT_gray = im2bw (PT_input,0.7);
[max_rows, max_columns] = size (PT_gray);
for i=1:max_rows
    Ck=0;
    for j=1:max_columns
        if PT_gray(i,j) == 0;
            Bx0 = i;
            x1=i;
            Ck = 1;
            break
        end
    end
    if Ck ==1;
        break
    end
end
for i=max_rows:-1:1
    Ck=0;
    for j=1:max_columns
        if PT_gray(i,j) == 0;
            Ex0 = i;
            x2=i;
            Ck=1;
            break
        end
    end
end
end

```

```

        if Ck ==1;
            break
        end
    end
end

for i=1:max_columns
    Ck=0;
    for j=1:max_rows
        if PT_gray(j,i) == 0;
            By0 = i;
            y1=i;
            Ck=1;
            break
        end
    end
    if Ck ==1;
        break
    end
end

for i=max_columns:-1:1
    Ck=0;
    for j=1:max_rows
        if PT_gray(j,i) == 0;
            Ey0 = i;
            y2=i;
            Ck=1;
            break
        end
    end
end
end

```



```

        if Ck == 1;
            break
        end
    end

    PT_input(Bx0,By0:Ey0,1) = 255;
    PT_input(Bx0,By0:Ey0,2) = 0;
    PT_input(Bx0,By0:Ey0,3) = 0;
    PT_input(Ex0,By0:Ey0,1) = 255;
    PT_input(Ex0,By0:Ey0,2) = 0;
    PT_input(Ex0,By0:Ey0,3) = 0;
    PT_input(Bx0:Ex0,By0,1) = 255;
    PT_input(Bx0:Ex0,By0,2) = 0;
    PT_input(Bx0:Ex0,By0,3) = 0;
    PT_input(Bx0:Ex0,Ey0,1) = 255;
    PT_input(Bx0:Ex0,Ey0,2) = 0;
    PT_input(Bx0:Ex0,Ey0,3) = 0;
    cntX = 4;
    cntY = 4;
    NoRX = 0;
    NoRY = 0;
    NX = Ex0-Bx0;
    NY = Ey0-By0;
    while or((NX>=10),(NY>=10))
        NoRX = NoRX + 1;
        NX = (Ex0-Bx0)/2^(NoRX);
        X0 = Bx0 ;
        X1 = round(Bx0 + (NX));
        NoRY = NoRY + 1;
        NY = (Ey0-By0)/2^(NoRY);

```

```

Y0 = By0 ;
Y1 = round(By0 + (NY));
end

ResultX = 1; ResultY = 1; CutSpaceX=0; CutSpaceY=0;

HafpiX = round(Bx0 + (Ex0-Bx0)/2);

MetRes(1,1) = HafpiX;

MetRes(1,3)=1;

HafpiY = round(By0 + (Ey0-By0)/2);

MetRes(1,2) = HafpiY;

MetRes(1,4)=1;

MetRes(1,5) = Bx0;

MetRes(1,6) = Ex0;

MetRes(1,7) = By0;

MetRes(1,8) = Ey0;

for y = By0 : Ey0
    f = 0;
    if PT_gray(HafpiX,y)~= 0
        PT_input(HafpiX,y,1) = 255;
        PT_input(HafpiX,y,2) = 0;
        PT_input(HafpiX,y,3) = 0;
        f = 1;
    end
end

if f == 1
    cntX = cntX + 1;
end

for x =Bx0 : Ex0
    f = 0;
    if PT_gray(x,HafpiY)~= 0

```

```

PT_input(x,HafpiY,1) = 255;
PT_input(x,HafpiY,2) = 0;
PT_input(x,HafpiY,3) = 0;
f = 1;
end
end

    if f == 1
        cntY = cntY + 1;
    end

MetX = [Bx0,Ex0];
MetY = [By0,Ey0];
Cx = 0; Cy=0;Px=1; Py=1;
depX = Ex0-Bx0;
depY = Ey0-By0;
count=1;
while and((depX>=10),(depY>=10))
    Cx = Cx+1;
    Cy = Cy+1;
[MetX,MetY]= divide(PT_gray,Bx0,Ex0,By0,Ey0,Cx,Cy);
    depX = MetX(2)- MetX(1);
    depY = MetY(2)- MetY(1);
    if and((depX>=10),(depY>=10))
        sx = size(MetX) ; sy =size(MetY);
        for Px=1 : sx(1,2)-1
            for Py=1 : sy(1,2)-1
                count = count+1;
                X0 = MetX(Px);
                X1 = MetX(Px+1);
                Y0 = MetY(Py);

```

```

        Y1 = MetY(Py+1);
[ResultX,CutSpaceX,NopiX,DC]= compareX(PT_gray,X0,X1,Y0,Y1,ResultX,CutSpaceX);
    if NopiX ~= 0
        MetRes(count,1) = NopiX;
        MetRes(count,3) = DC;
        for y = Y0 : Y1
            f=0;
            if PT_gray(NopiX,y)~= 0
                PT_input(NopiX,y,1) = 255;
                PT_input(NopiX,y,2) = 0;
                PT_input(NopiX,y,3) = 0;
                f=1;
            end
        end
        if f==1
            cntX = cntX + 1;
        end
    end
[ResultY,CutSpaceY,NopiY,DC]= compareY(PT_gray,X0,X1,Y0,Y1,ResultY,CutSpaceY);
    if NopiY ~= 0
        MetRes(count,2) = NopiY;
        MetRes(count,4) = DC;
        for x = X0 : X1
            f=0;
            if PT_gray(x,NopiY)~= 0
                PT_input(x,NopiY,1) = 255;
                PT_input(x,NopiY,2) = 0;
                PT_input(x,NopiY,3) = 0;

```



```

        f=1;
        end
    end

    if f==1
        cntY = cntY + 1;
        end
end

if NopiY == 0
    MetRes(count,4) = 0;
end

if NopiX == 0
    MetRes(count,3) = 0;
end

MetRes(count,5) = X0;
MetRes(count,6) = X1;
MetRes(count,7) = Y0;
MetRes(count,8) = Y1;
    end
end
end

TotalCutSpace = CutSpaceX + CutSpaceY;
fprintf('จำนวนพิกเซลที่ตัดทั้งหมด (De-cubing process) : %d พิกเซล\n',TotalCutSpace)
fprintf('จำนวนรอบการเดินทางของเครื่องมือตัดแนวตั้ง(ตัดแกนX) : %d ครั้ง\nจำนวนพิกเซลที่ถูกตัดแนวตั้ง(De-cubing)ตัดแกนX : %d พิกเซล\n',cntX,CutSpaceX)
fprintf('จำนวนรอบการเดินทางของเครื่องมือตัดแนวนอน(ตัดแกนY) : %d ครั้ง\nจำนวนพิกเซลที่ถูกตัดแนวนอน(De-cubing)ตัดแกนY : %d พิกเซล\n', cntY, CutSpaceY)

```

ข.2.1 ฟังก์ชัน Compare X วิเคราะห์การสร้าง Adaptive de-cubing ในแกน X

```
function [R,CutSpace,pixelCutX,dc] = compareX(PT,Bx,Ex,By,Ey,countX,CutSpace)

    dc = 0;
    for j=By:Ey
        back = 1;
        for i=Bx:Ex
            if PT(i,j)==0
                back = back + 1;
            end
        end
        s = (Ex-Bx)+1;
        if back >= round(0.25*s)
            dc = 1;
            R = countX + 1;
            pixelCutX = Bx + floor(s/2);
            for j=By:Ey
                if PT(pixelCutX,j)==1
                    CutSpace = CutSpace + 1;
                end
            end
            break
        else
            CutSpace = CutSpace;
            pixelCutX = 0;
            R = countX;
        end
    end
end
```

ข.2.2 ฟังก์ชัน Compare Y วิเคราะห์การสร้าง Adaptive de-cubing ในแกน Y

```

function [R,CutSpace,pixelCutY,dc] = compareY(PT,Bx,Ex,By,Ey,countY,CutSpace)

    dc = 0;
    for i=Bx:Ex
        back = 1;
        for j=By:Ey
            if PT(i,j)==0
                back = back + 1;
            end
        end
        s = (Ey-By)+1;
        if back >= round(0.25*s)
            dc = 1;
            R = countY + 1;
            pixelCutY = By + floor(s/2);
            for i=Bx:Ex
                if PT(i,pixelCutY) == 1
                    CutSpace = CutSpace + 1;
                end
            end
            break
        else
            CutSpace = CutSpace;
            pixelCutY = 0;
            R = countY;
        end
    end
end

```

ข.2.3 ฟังก์ชัน divide วิเคราะห์และเก็บค่าการแบ่งลงย่อยในการวนแต่ละรอบ

```
function [MetX,MetY] = divide(PT,Bx,Ex,By,Ey,X,Y)
r=2; n=2;
HafX = (Ex-Bx)/2^X;
HafY = (Ey-By)/2^Y;
MetX(1) = Bx;
MetY(1) = By;
for r=2 : 1 + 2^X
    MetX(r) = round(Bx + (HafX*(r-1)));
end
for n=2 : 1 + 2^Y
    MetY(n) = round(By + (HafY*(n-1)));
end
end
```



ค.1 ขั้นตอนการใช้งานโปรแกรมวิเคราะห์การสร้าง Uniform de-cubing

1. ใส่ชื่อภาพและประเภทของไฟล์ภาพดังแสดงในรูปที่ ค.1.1

```
PT_input = imread('Ex1.jpg');
PT_gray = im2bw(PT_input,0.7);
```

รูปที่ ค.1.1 แสดงการนำเข้าไฟล์รูปภาพ

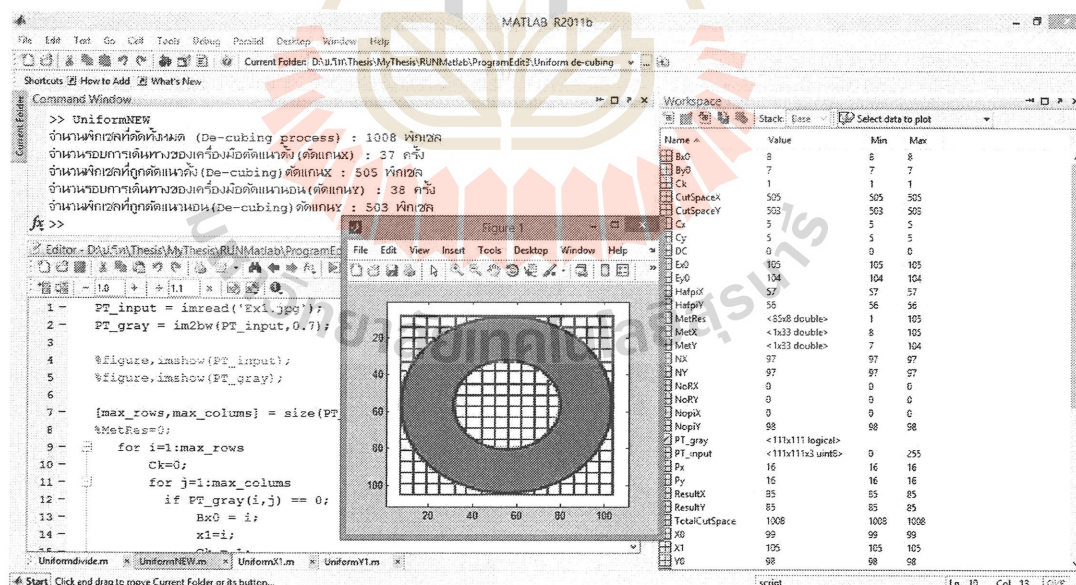
2. กำหนดค่า จำนวนพิกเซลที่เล็กที่สุดของกล่องขอบเขตย่อยที่จะพิจารณาต่อตามเงื่อนไขที่กำหนด β ในฟังก์ชันชื่อ “UniformX” และ “UniformY” ดังแสดงในรูปที่ ค.1.2

```
s = (Ex-Bx)+1;
if (Ex-Bx) >= 10

s = (Ey-By)+1;
if (Ey-By) >= 10
```

รูปที่ ค.1.2 แสดงการกำหนดค่า β ในโปรแกรม

3. รันโปรแกรมเพื่อแสดงผลดังแสดงในรูปที่ ค.1.3



รูปที่ ค.1.3 หน้าการแสดงผลการรันโปรแกรมวิเคราะห์ Uniform de-cubing

ค.2 ขั้นตอนการใช้งานโปรแกรมวิเคราะห์การสร้าง Adaptive de-cubing

1. ใส่ชื่อภาพและประเภทของไฟล์ภาพดังแสดงในรูปที่ ค.1.4

```
PT_input = imread('Ex1.jpg');
PT_gray = im2bw(PT_input,0.7);
```

รูปที่ ค.1.4 แสดงการนำเข้าไฟล์รูปภาพ

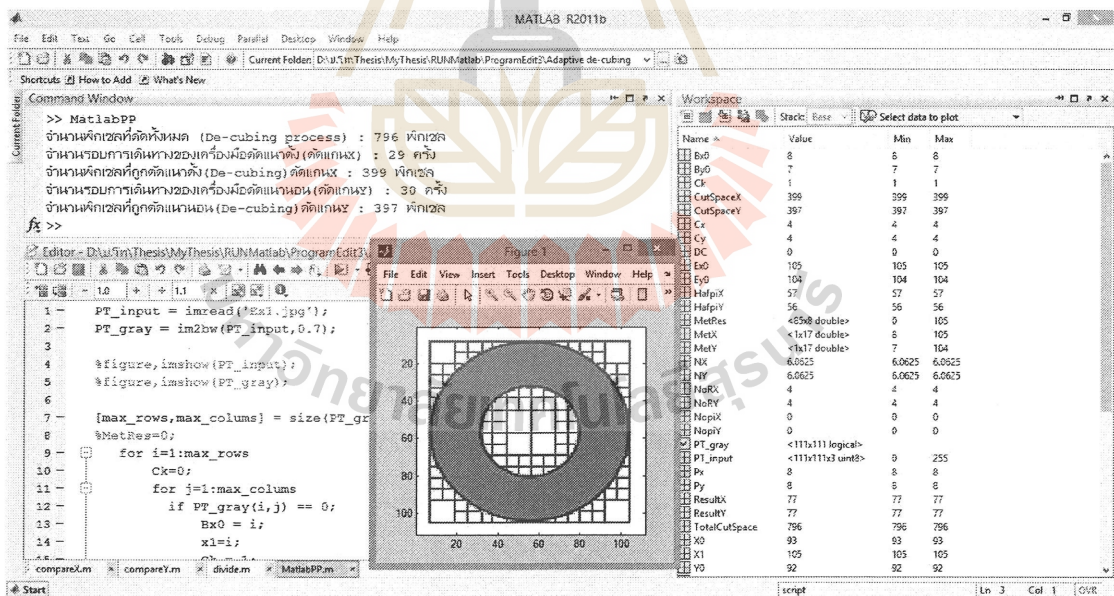
2. กำหนดค่าอัตราส่วนของจำนวนพิกเซลสีดำต่อพิกเซลสีขาว (τ) ในฟังก์ชันชื่อ “CompareX” และ “CompareY” ดังแสดงในรูปที่ ค.1.5

```
s = (Ex-Bx)+1;
if back >= round(0.25*s)
```

```
s = (Ey-By)+1;
if back >= round(0.25*s)
```

รูปที่ ค.1.5 แสดงการกำหนดค่า τ ในโปรแกรม

3. รันโปรแกรมเพื่อแสดงผลดังแสดงในรูปที่ ค.1.6



รูปที่ ค.1.6 หน้าการแสดงผลการรัน โปรแกรมวิเคราะห์ Adaptive de-cubing

ภาคผนวก ง
บทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างศึกษา



รายชื่อบทความที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างศึกษา

Paphakorn Pitayachaval, Thanakharn Baotthong and Niramol Chaphol. **A Tool Path Planning For 2d Contour Cutting In Rapid Prototyping Technology.**

การประชุมวิชาการวิทยาศาสตร์และเทคโนโลยีแห่งประเทศไทย ครั้งที่ 41 (วทท 41),
The 41st Congress on science and Technology of Thailand (STT41), November 6-8, 2015.

Paphakorn Pitayachaval, Paramate Horkaew and Niramol Chaphol. **An Adaptive De-Cubing Processing For Laminated Object Manufacturing (Lom) Based On Contour Proportion in Each Square Cube.**

10th SOUTH EAST ASIAN TECHNICAL UNIVERSITY CONSORTIUM
SYMPOSIUM, February 22-24, 2016, Shibaura Institute of Technology.

Paphakorn Pitayachaval, Muhammatsoifu Sato and Niramol Chaphol. **Simplify GO-NO Gage For Quality Inspection In Pe Bottle Production.**

11th SOUTH EAST ASIAN TECHNICAL UNIVERSITY CONSORTIUM
SYMPOSIUM, March 13-14, 2017, Ho Chi Minh University of Technology, Viet Nam.

Paphakorn Pitayachaval, Paramate Horkaew and Niramol Chaphol. **Generate an Adaptive De-cubing Automatic Processing for Laminated Object Manufacturing (LOM).**

The 4th International Conference on Industrial Engineering and Applications (ICIEA
2017), April 21-23, 2017, Nagoya Institute of Technology, Nagoya, Japan.

A TOOL PATH PLANNING FOR 2D CONTOUR CUTTING IN RAPID PROTOTYPING TECHNOLOGY

Paphakorn Pitayachaval,¹ Thanakharn Baotthong,² Niramol Chaphol^{1,*}

School of Industrial Engineering, Institute of Engineering, Suranaree University of Technology, Amphur Muang, Nakhorn Ratchasima 30000

*e-mail: ¹email: Paphakorn@g.sut.ac.th, ²email: Bthanakharn@gmail.com, ³email: Niramol1990@gmail.com

Abstract: A tool path planning for one layer of RP can be classified into two types: tracing on contours (operating time) and moving between contours (non-operating time), to move from the end of contour to the start of the next contour in same layer. In order to establish an optimize tool path planning, paths of tool moving between contours have be shortened, since these paths consume time and energy. This paper presents two algorithms for tool path planning that minimize distance tool travelling between contours: the searched point based on home position and the searched point based on previous contour. The searched points are utilized as connecting points between contours. Then, a concept of travelling salesman problem (TSP) and heuristic method have been applied to generate possible tool paths. The algorithms had been implemented on a case study, and the distance of total tool path from those two algorithms were also presented.

Introduction: Rapid Prototyping technology (RP), also called Layered Manufacturing (LM) or Solid Freeform Fabrication (SFF), is a fabrication process of prototype created layer-by-layer directly from Computer Aided Design (CAD) model. Since there may be several contours on one layer, the cutting tool tracing that traces the considering contours consist two types: tracing on contours (operating time) and moving between contours (non-operating time), to move from the end of contour to the start of the next contour in same layer. Thus, the total fabrication time for one layer consists of two components: the cutting tool operating time and cutting tool non-operating time. Since there are many contours on one layer that may be nested contours or independent contours, the inner contours and outer contours have to identify in order to assign part and non-part areas. Moreover, each contour has multiple vertices that can be a possible starting point and the ending point for tool tracing. These present several possible solutions for cutting tool path planning. Additionally, contour relationship identification is also required before assign tool path planning in order to avoid miss-cutting on part area. This paper presents contour identification for tool path planning of a rapid prototyping construction. In order to specify the starting and ending points on each contour, there are two proposed algorithms after applying parent-and-child relationship concept. A concept of the travelling salesman problem has been utilized to determine tool path sequences for minimizing distance between contours. In addition, proposed algorithms are also implemented on a case study.

Methodology: Tool Path Planning

To determine tool path, a process starts with a slice contours image as an input as shown in Fig. 1. Then, object contours are extracted from this image. These contours are defined their relationships. Before the possible tool paths are established, a point on each contour is defined. This point will be a starting point and an ending point for a tracing contour. In order to determine these points, two algorithms are presented: the searched point based on home position and the searched point based on previous contour. The first algorithm defined a point on each contour by searching the nearest point with the home position while the second algorithm assigned a point on existing contour by comparing with the point on previous contour. A concept of the first algorithm is shown in Fig. 2(a). The algorithm assigns a home position at the bottom left corner of the slice contour image and specifies a point on particular contour by searching the nearest point on contour with home position. This point on contour can be explained by using Eq. 1.

Min:

$$\text{Distance} = \sqrt{(X_i - X_h)^2} + \sqrt{(Y_i - Y_h)^2} \quad (1)$$

Where

$P = (X_i, Y_i) : P \in C_x$

(X_h, Y_h) is the coordinate of home position

(X_i, Y_i) is the coordinate of point on particular contour

C_x is x^{th} contour

A concept of the second algorithm is shown in Fig. 2(b), the algorithm specifies a home position at the bottom left corner of the slice contour image as same as the first algorithm. The algorithm specifies a point on particular contour of the first contour by searching the nearest contour with the home position, and then assigns a point on the next contour by discovering the nearest point on contour that compare with point on existing contour. Subsequently, after searching points on all contours, a concept of the travelling salesman problem has been applied to determine cutting tool path in when tool is assigned as a salesman and considering contours are defined as cities. The condition is that the salesman must travel to all cities only one time then come back to a home position. Moreover, the nearest neighbor heuristic technique is utilized to identify the next point on contour as shown in Fig. 1. To obtain all possible tool paths, the mathematical model is as follows:

$$\text{Min } Z(T) = \sum_{i,j \in T} D_{ij} X_{ij} \quad (2)$$

$$\sum_{j=1}^N X_{ij} = 1 \quad \forall i \in n \quad (3)$$

$$\sum_{i=1}^N X_{ij} = 1 \quad \forall j \in n \quad (4)$$

$$\sum_{i,j \in S} X_{ij} \leq |S| - 1, S \subset V, |S| > 1 \quad (5)$$

$$X_{ij} \in \{0,1\} \quad (6)$$

The objective function aims to minimize distance of laser path travelling between contours, as shown in Eq. 2. When T is a set of the path sequence, D_{ij} is denoted as travelling distance from point i on existing contour to point j on the next contour, N is denoted as the number of points on contours. Eq. 3, Eq. 4, Eq. 6 are constraints in which if $X_{ij} = 1$ then there is path from i to j , and $X_{ij} = 0$ then there is no path from i to j in order to guarantee that particular contour is visited exactly once time. Additionally, Eq. 5 is a sub-tour elimination constraint.

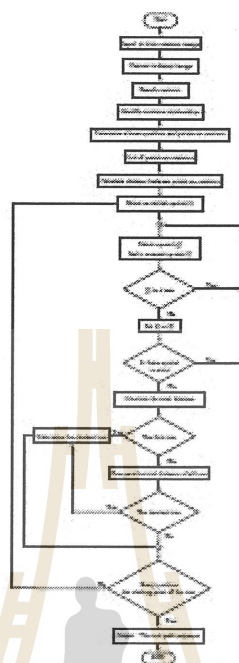


Figure 1. Flow chart



Figure 2. The two algorithms

Results and Discussion: Case Study

MATLAB program was utilized as a tool to implement contour searching algorithms and establish tool path for a specified layer. The contour searching algorithms were performed to identify contour relationship and generate tool path planning for a specified layer of machine part. A slicing plane was assigned on a CAD model, as shown in Fig. 3 (a). An image of investigated layer was extracted from CAD model and was conducted as input of algorithms, as shown in Fig. 3 (b) and 3 (c), respectively. For this layer, there were eight contours that were identified their relationship based on parent-and-child hierarchy, as shown in Fig. 3 (d) and 4 (a), respectively. Contours C1, C2, C3, and C4 had no parent, so they were in level 0 and their child were in the next level. Then, a point on each contour was presented including a home position. This point was a starting point and an ending point of a considering contour. The points of contours, defined by using those two algorithms, were established, as shown in Table 1. The tool path planning result was starting from home position $\rightarrow C1 \rightarrow C5 \rightarrow C2 \rightarrow C6 \rightarrow C3 \rightarrow C7 \rightarrow C4 \rightarrow C8 \rightarrow$ home position. The tool paths were presented that contour would be cut base on point coordinates that were defined by those

two algorithms. The distances of tool path travel using those two algorithms were 1335.3644 and 1316.7035 length units, as shown in Fig. 4 (b) and 4 (c), respectively. This result showed that the searched points base on home position assigned optimum distance for tool path planning

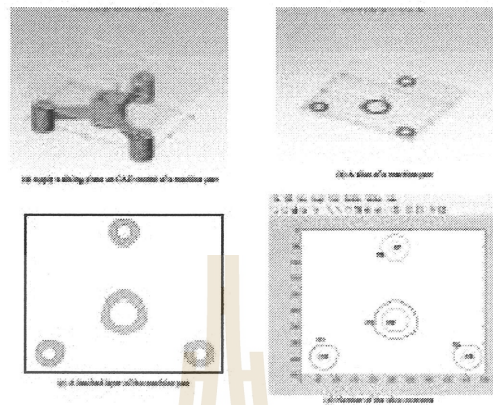


Figure 3. Section view of machine part

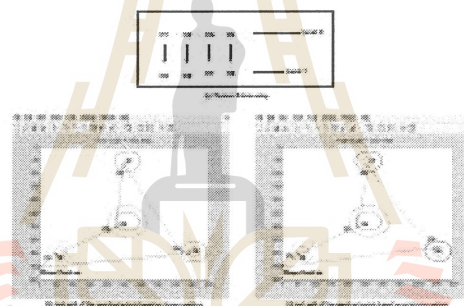


Figure 4. Contour relationship and tool path

Table 1. The result of points on contours between the searched points based on home position and the searched points based on previous contour algorithms

Contours	the searched points based on home position	the searched points based on previous contour
Home Position	(0,452)	(0,452)
C1	(34,418)	(34,418)
C2	(208,314)	(49,404)
C3	(246,95)	(208,314)
C4	(414,368)	(228,299)
C5	(49,404)	(254,97)
C6	(228,299)	(256,72)
C7	(249,69)	(428,362)
C8	(439,400)	(444,381)

Conclusion: The two algorithms for searching a starting point on each contour have been presented: the searching points based on home position and the searching points based on previous contour. The concept of TSP and heuristic Nearest Neighbor have been applied to determine tool path planning in order to minimize distance of tool path for constructing layer on rapid prototyping. The result was shown that starting point, assigned by the searched points based on previous contour algorithm, provided shortest tool path for layer contour construction, as shown in Table 2. For a future work, the searched points based on previous contour algorithm will be implemented on rapid prototyping machine. Moreover, this algorithm will be also applied to create prototype based on the multi-material layered manufacturing.

Table 2. The result comparison between the searched points based on home position and the searched points based on previous contour algorithms

non-operating path	the searched points based on home position	the searched points based on previous contour
1) travel from home position to first contour (Unit)	48.0833	48.0833
2) travel between contours and finally (Unit)	845.2121	818.9792
3) travel from last contour to the home position (Unit)	442.0690	449.6410
4) total distance (Unit)	1335.3644	1316.7035

References:

1. Choi, S.H. and Kwok, K.T., in: *Hierarchical slice contours for layered-manufacturing*, Computer Industrial, 48:99-114. (2002).
2. Choi, S.H. and Kwok, K.T., in: *A topological hierarchy-sorting algorithm for layered manufacturing*, Rapid Prototyping Journal, 10(2), 98-113. (2004)
3. Soonanon, P., Koomsap, P., and Chansri, N., in: *Identifying Contour(s) for Layer-Based Geometrical Reconstruction*, the 8th Asia Pacific Industrial Engineering and Management Systems Conference (APIEMS), Kaohsiung, Taiwan (2007).
4. Kolahen, F., and Liang, M., in: *Optimization of hole-making operations: a tabu-search approach*, International Journal of Machine Tools & Manufacturing, 40, pp. 1735-1753. (2002).
5. Castellino, K., D'Souza, R. and Wright, P. K., in: *Toolpath optimization for minimizing airtime during machining*, Journal of Manufacturing Systems, Vol.22, No.3, pp.173-180. (2002)
6. Joneja, A., Pang, K.W., Murty, K.G., Lam, D. and Yuen, M., in: *A genetic algorithm for path planning in rapid prototyping*, Proceeding of DETC'98 1998 ASME Design Engineering Technical Conference, September 13-16, Atlanta (1998).
7. Tang, K., and Pang, A., in: *Optimal connection of loops in laminated object manufacturing*, Computer-Aided Design, 35(11): 1011-1022. (2003)
8. C.C. Chou, Y.K. Chen and S.Y. Chou, in: *Shortest traversal path of n circles in layered manufacturing applications*, IEEE Computer Graphics, Imaging and Visualization: New Advances, Bangkok, Thailand (2007)

AN ADAPTIVE DE-CUBING PROCESSING FOR LAMINATED OBJECT MANUFACTURING (LOM) BASED ON CONTOUR PROPORTION IN EACH SQUARE CUBE

Paphakorn Pitayachaval^{1*}, Paramate Horkaew², Niramol Chaphol¹

¹School of Industrial Engineering, Institute of Engineering, Suranaree University of Technology

²School of Computer Engineering, Institute of Engineering, Suranaree University of Technology

*E-mail Address: Paphakorn@g.sut.ac.th

ABSTRACT

A Laminated Object Manufacturing (LOM) technique, which is one of a rapid prototyping technique, creates a physical model directly from 3D CAD model without mold and dies by using laminated material. To complete a model, a tool is used to cut laminated material each layer according to layer contours in which the waste material has to remove as de-cubing process. The de-cubing process is to assign shape of waste material into generally small square shape that can be easily remove in order to reduce time and avoid prototype damages. This paper presents a process to assign the size of waste material in the de-cubing process. The adaptive de-cubing process has been established by using proportion of a length of contour section per a length side of a considering square shape. If the proportion is more than a threshold, then the considering side will be divided. On the other hand, if the proportion is less than the threshold, then the considering side will be skipped. The result showed the variety size of waste material.

Keywords: Rapid prototype, Laminated Object Manufacturing

1. INTRODUCTION

Rapid prototyping process is a manufacturing process that manufactures products directly from CAD model without mold and die. It was introduced to shorten prototype construction time. This process converts a 3D CAD model to be a stack of 2D contours which are used to generate machine commands to build an object layer by layer. In this process, material is added to produce a part layer by layer. Several techniques have been developed for rapid prototyping process and they can be classified into 3 categories based on the initial state of materials [1] liquid-based, solid-based and powder-based. Stereolithography

apparatus (SLA), a popular technique in a liquid-based system, is the first commercial rapid prototyping system. In this technique, UV laser is used to solidify photo-curable liquid polymer to form layers. Selective Laser Sintering (SLS) is a famous technique in a powder-based rapid prototyping system. This technique is similar to SLA but instead of curing liquid polymer with UV laser, CO₂ laser is used in SLS to sinter powder to form a layer. Similarly, 3D printing is also a popular technique in a powder-based rapid prototyping system. Besides using UV laser, glue had been applied. Last group of rapid prototyping system is solid-based. In this group, there are several types of solid material used including wire, roll of laminated sheet and pellets. Popular techniques of this group are Laminated Object Manufacturing (LOM) and Fused Deposition Modelling (FDM) [2].

To complete a model created by LOM, a tool is used to cut laminated material each layer according to layer contours in which the waste material has to remove as de-cubing process. The de-cubing process is to assign shape of waste material into generally small square shape that can be easily remove in order to reduce time and avoid prototype damages.

This paper presents a process to assign the size of waste material in the de-cubing process. The adaptive de-cubing process has been established by using proportion of a length of contour section per a length side of a considering square shape.

2. LITERATURE REVIEW

Based on the process of cutting laminate material in LOM, a defective model is generally occurred during contour cutting step. A method of "cut then bond" had been introduced [6]. The outer area of model contour is cut into equal squares, called uniform de-cube, simultaneously cutting outer contour. For this method,

the edging of prototype is quite rough and difficult to separate between model and scrap. Moreover, Hur and Lee [7] proposed the problems occurred during de-cubing process: the cumulative error, which is introduced into the serious problem and the rest of scrap material that sticks with the model especially, a complexity model and a hollow model. For the hollow model, Cho and Lee [8] were proposed an approach to create by using LOM. There are two steps: cut outer contour of model for a concurrent layer then remove the scrap material before bound with a pervious layer. This approach presented more accumulative error. In order to reduce the accumulative error, Chiu, et al. [9] proposed the bridge generation to connect the inner model area and the outer model area. Even though this approach reduce error in X and Y axis but error in Z axis is increase in which the model is bended and twisted.

3. AN ADAPTIVE DE-CUBING ALGORITHM

An adaptive de-cubing is to assign shape of waste material into generally small non-uniform square shape that can be easily remove in order to reduce time and avoid prototype damages. The algorithm is proposed, as shown in figure 1. An input is an image of 2D layer contour. Since the contour image presents the redundancies areas that are not related with contour information, a contour bounding box is crated to identify a working area. This working area presents both outer and inner areas of particular layer contour. By using a scan line algorithm, the outer and inner areas are identified and the image pixels of inner area are assigned by black color otherwise white color is applied. This working area is divided into four main equal cubes based on the half points along Y axis and Y axis. Then, an adaptive de-cube calculating is applied on each cube, starting from four main equal.

Each main cube is investigated to calculate adaptive de-cube. The adaptive de-cubing process applies proportion of a length of contour section per a length side of a considering square shape. If the proportion is more than a threshold, then the considering side will be divided. On the other hand, if the proportion is less than the threshold, then the considering side will be skipped. A condition for dividing cube using threshold is established in equation (1).

$$\frac{\sum_{i=1}^N x_i}{N} \times 100 \geq t_{th} \quad (1)$$

when x is number of pixels that represent inner areas, N is number of pixels in particular cube and t_{th} is the threshold.

This condition is applied until there is no sub-cube contains pixel more than threshold. When the particular cube is assigned to divide, the cutting points on X axis and Y axis are recorded. Finally, the output of this process are cutting points that can be use to generate tool path planning for adaptive de-cubing.

4. CASE STUDY

To implement the adaptive de-cube algorithm, a cylinder model was studied, as shown in figure 2, by obtaining layer contours image. Then, the contour bounding box concept and the scan line algorithm were applied, the working area and both inner area (black pixels) and outer (white pixels) area were established, respectively, as shown in figure 3.

After working area was define, the working area was divided into four main uniform cubes, as shown in figure 4(a). Then, the adaptive de-cube was applied for all main cubes, as shown in figure 4(b).

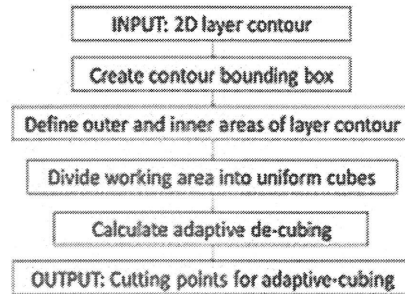


Figure 1: Adaptive de-cubing algorithm



Figure 2: Layer contours of sliced cylinder model

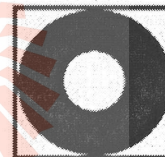
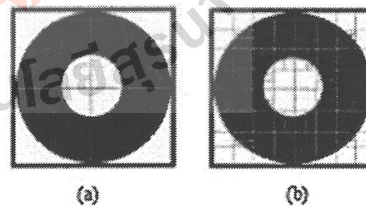


Figure 3: The working area of particular layer contours



(a)

(b)

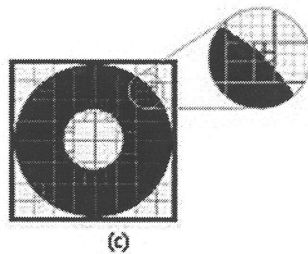


Figure 4: The adaptive de-cubes of layer contour

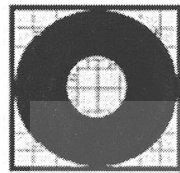


Figure 5: Tool path for adaptive de-cubing

The result showed the variety size of waste material, as shown in figure 4(c). All cutting points were recorded for tool path planning that were assigned on the waste material, as show in figure 5.

5. CONCLUSION

The adaptive de-cubing process has been established by using proportion of a length of contour section per a length side of a considering square shape. The variety sizes of waste material were established in order to easy remove waste from model edge. By using this algorithm, the automatically identify threshold comparing with the model error will be studied as a future work.

6. ACKNOWLEDGE

This research has been supported by the research and development supporting fund, Suranaree University of Technology.

REFERENCES

- [1] D. Pham, and R. Gault, A comparison of rapid prototyping technologies. *International Journal of Machine Tools and Manufacture* 38 (10-11), 1998, pp. 1257-1287.
- [2] V. Kumar, and D. Dutta, An assessment of data formats for layered manufacturing. *Advances in Engineering Software* 28 (3), 1997, pp. 151-164.
- [3] P. Soonanon, and P. Koomsap, Towards direct transformation of orthographic-view drawings into a prototype. *Virtual and Physical Prototyping*, 4 (2), 2009, pp. 75-90.
- [4] W. Cao, and Y. Miyamoto, Direct slicing from AutoCAD solid models for rapid prototyping. *The*

International Journal of Advanced Manufacturing Technology 21 (10-11), 2003, pp. 739-742.

[5] P. Pitayachaval and T. Baotthong, A tool path planning of rapid prototype technology base on a contour relationship. *Journal of engineering Ubolratchathani university* 1 (6), 2556, pp. 61-69. (in Thai)

[6] Charles L. Thomas, *Automating Sheet-Based Fabrication: The Conveyed-Adherent™ Process*, 1996.

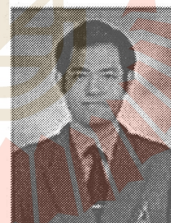
[7] J. Hux, K. Lee, Efficient algorithm for automatic support structure generation in layered manufacturing, in: *Proceedings of the 1996 ASME Design Engineering Technical Conference and Computers in Engineering Conference*, Irvine, USA, 1996.

[8] Cho, I., Lee, K., Choi, W., & Song, Y.-A. (2000). Development of a new sheet deposition type rapid prototyping system. *International Journal of Machine Tools and Manufacture*, 40 (12), 1813-1829.

[9] Y.Y. Chiu, Y.S. Liao and C.C. Hou, The Study of On-line Waste Material Removal Procedures for Bridge Laminated Object Manufacturing (LOM) Process. *Journal of Materials Processing Technology*, 2015.



Paphakorn Pitayachaval received the B.Eng.(1993) from Suranaree University of Technology. M.Eng. Industrial and Manufacturing Systems Engineering (2000) Ph.D. (Design and Manufacturing Engineering), Asian Institute of Technology (AIT) (2009) She is Assistance Professor, Department of Industrial Engineering, Suranaree University of Technology



Paramate Horkaew received the B.Eng. Program in Telecommunications Engineering from King Mongkut's Institute of Technology Ladkrabang Ph.D. (Computer Science), Imperial College London He is Assistance Professor, Department of Computer Engineering, Suranaree University of Technology.



Niramol Chaphol received the B.Eng. (2014) from Suranaree University of Technology. Currently Studying for Master Degree in Industrial Engineering, Suranaree University of Technology

SIMPLIFY GO-NO GAGE FOR QUALITY INSPECTION IN PE BOTTLE PRODUCTION

Paphakorn Pitayachaval*, Muhammadsoifu Sato, Niramol Chaphol
School of Industrial Engineering, Institute of Engineering, Suranaree University of
Technology, NakhonRatchasima, 30000, Thailand

*E-mail Address: paphakorn@g.sut.ac.th

ABSTRACT

In PE bottle production, the bottle neck is an important component in order assembly with cap. However, operator have to more concentrate during quality checking by using vernier caliper. There are several problems occurred due to that operator. This paper presents a process to simplify GO-NO gage by apply fish bone diagram, SWIH analysis and why-why analysis in order to reduce inspection time, reduce scrap while operators are pleased. The result of simplify GO-NO gage were showed as rapid prototype model in order to trial with operators in production line.

KEYWORDS: GO-NO gage, Rapid prototype, Quality inspection

1. INTRODUCTION

In present, a plastic forming process is rapid process to form part, specially to produce bottle. Since plastic forming process contains many feature details and complex geometry, the inspection is time consuming. Beside checking feature details, there are many bottle sizes that have to simultaneously inspection according customer requirement. Moreover, a batch of bottle production is a mass production. The operator might be confused for dimension checking of the small lot size. For a considered company, a vernier caliper is employed to inspect PE bottle after a preform model is injection. A preform model inspection is a consuming process in which presents a bottle neck for bottle production. In order to solve this problem, a rapid prototype technology[1] has been applied to establish inspection gage for process improvement. However, the root cause for process improvement has be determined by before the gage is designed[4].

This paper presents a process to simplify GO-NO gage by apply why-why analysis and fish bone diagrams in order to reduce inspection time, reduce scrap while operators are pleased.

2. PROBLEM IDENTIFICATION

PE bottle production process is start with fusion grain plastic[2], then a polyethylene fusion forms PE to a preform tube. The preform tube can be used to form any bottle shape which is depend upon blow mold. Finally, the bottle is packed in a batch and is prepared to deliver, as shown in figure 1. However, during preform tube is constructed, there might be some flaws in which the inspector need to monitor before forming PE process to contain perfect PE bottle [3]. If inspector defines that the preform tube does not perform assigned dimension, then the following process, blow molding, should be stop. Simultaneously, the process of constructing preform tube should be identified problem.

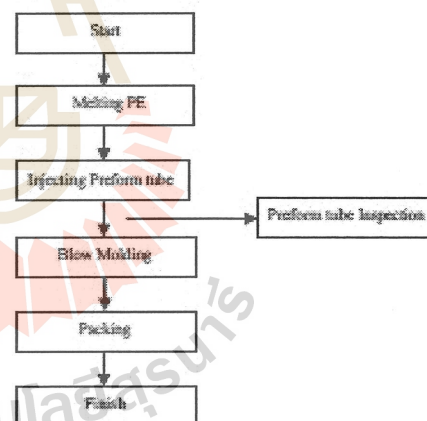


Figure 1 :A manufacturing process of PE bottle

A defective perform tube is defined as several cases such as un-expectation dimension, flaw on tube and miss-dimension on wall thickness. So, the inspector need to check at less five dimension: outer diameter, length, wall thickness, neck thickness and twist length, as shown in figure 2.

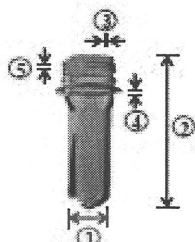


Figure 2 :The dimension of preform tube inspection

In the considering company, the inspector uses Vernier caliper to measure those dimensions that cause time consuming for the inspection process in which the bottle neck is presented. Moreover, there are operators that do not familiar with Vernier caliper so they ignore to concern measuring. To reduce inspection time and motivate operator to beware defective part, a fish bone diagram and a why-why analysis have been used to identify root cause for inspection problem. To determine cause and effect of the bottle neck of inspection process, the fish bone diagram was performed, as shown in figure 3. Those causes were analyzed via 5WHY to identify a solution, as shown in Table 1.

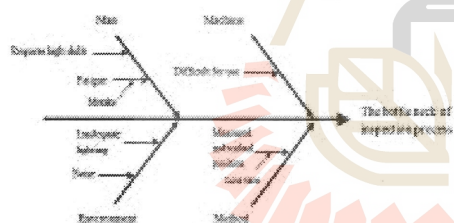


Figure 3 :The fish bone diagram analysis

Table 1: 5WHY analysis

WHAT	WHY	WHERE	WHO	WHEN	HOW
Crabbed scale vernier caliper	vernier is analog tool	Inspection zone	Inspector	15/11	change to digital

Measure a lot of checking point	a lot of major point	Inspection zone	Inspector	15/11	change to method or tool
No skill	never use tool and Lack of training	Inspection zone	Inspector	15/11	training how to use tool
Inadequate lighting and noise	No have inspection room	Inspection zone	Inspector	15/11	Build inspection room

Why 1: Why does a bottle neck occur?

Answer: Because operators does not familiar with Vernier caliper.

Why 2: Why do they use Vernier caliper?

Answer: They use to measure concerned dimension.

Why 3: Why do we have to concern those dimensions?

Answer: Those dimensions present a good product quality.

Figure 4: Why-why analysis

The why-why analysis presents that bottle neck occurs due to several dimension have to inspection while the operator does not familiar with tool. Thus, in order to reduce time consuming. The GO-NO gage has been established to measure several dimension as the same time. Moreover, the GO-NO gage has to friendly use while do not require skill of inspector.

3. GO-NO GAGE CREATED

To construct GO-NO gage, a rapid prototype technology is applied, as shown in figure 5. Starting with design gage with accurate dimension of preform tube was presented in a CAD file via Rhinoceros program, as shown in figure 6. Then, a 3D model of GO-NO gage was build, as shown in figure 7. Finally, the GO-NO gage was tested to inspection preform tube, as shown in figure 8.

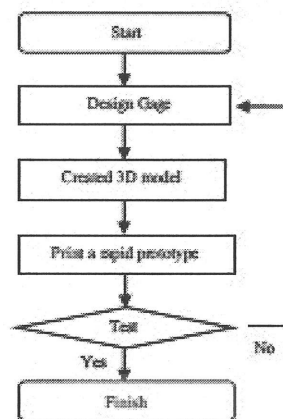


Figure 5 :Constructing GO-NO gage process

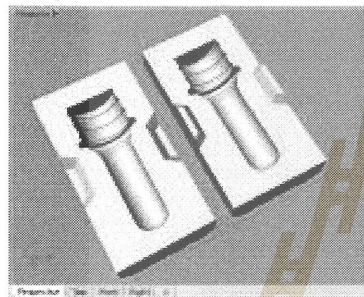


Figure 6 :3D CAD model of GO-NO gage



Figure 7 :Rapid prototype model of GO-NO gage

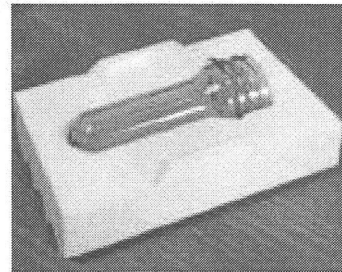


Figure 8 :Inspection preform tube by GO-NO gage

4 RESULT

Table 2: Comparing inspection time between Vernier caliper and GO-NO gage

Amount of example (piece/30min)	Time to inspection by vernier caliper (min/piece)	Time to inspection by GO-NO gage (min/piece)	Reduce Time
Model 1 :16	1.875	0.45	76%
Model 2 :24	1.25	0.32	74.4%

The inspection consumed time were recorded, as shown in Table 2. The result was shown that inspection time by GO-NO gage is less than inspection by Vernier caliper more than 70%. This approach be able to reduce the over processing waste, high accuracy and can be improve productivity.

5 .CONCLUSION

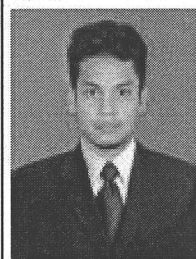
To product PE bottle, several dimension of preform tube have to be controlled even this operation consumes time and propose bottle neck. To reduce inspection time and avoid human error, the fish bone diagram, 5WH analysis and why-why analysis was performed to initiate exploit idea. The rapid prototype technology was used to present GO-NO gage rapidly without mold and die construction. The simplify GO-NO gage was established to quality control in the inspection process.

6. ACKNOWLEDGE

This research has been supported by the research and development supporting fund, Suranaree University of Technology.

REFERENCES

- [1] Waterman, N.A., Dickens, P., 1994. Rapid product development in the USA, Europe and Japan, World class Design to manufacture 1 (3):27-36.
- [2] Payaksiri, N., 2005. Extrusion blow molding process of modified CaCO₃-filled HDPE bottles, King Mongkut's Institute of Technology Ladkrabang. Science Program in Rubber Technology. Publisher King Mongkut's Institute of Technology Ladkrabang. Central Library.
- [3] Boonsriudomruk, S., 2000. Recipe Improvements in the Production of Polyethylene Based-Foam Sheets ISBN: 9746282743. King Mongkut's University of Technology Thonburi.
- [4] Dey, A.K., Abowd, G.D., 2000. "Towards a better understanding of context and contextawareness." In the Workshop on the What, Who, Where, When and How of Context-Awareness, affiliated with the 2000 ACM Conference on Human Factors in Computer Systems.
- [5] Chaichiana, M., Prešnajdera, P., Sheikh-Jabbaria, M.M., Tureama, A., 2001. Noncommutative gauge field theories: a no-go theorem, Received 3 December 2001, Accepted 14 December 2001, Available online 4 January 2002.
- [6] Cao, W., Miyamoto, Y., 2003. Direct slicing from AutoCAD solid models for rapid prototyping. The International Journal of Advanced Manufacturing Technology 21 (10-11), 2003, pp. 739-742.



Muhammadsoifu Sato received the B.Eng. (2013) from Suranaree University of Technology. Currently Studying for Master Degree in Industrial Engineering, Suranaree University of Technology



Niramol Chaphol received the B.Eng. (2014) from Suranaree University of Technology. Currently Studying for Master Degree in Industrial Engineering, Suranaree University of Technology



Paphakorn Pitayachaval received the B.E. (1998) from Suranaree University of Technology, M.E. (2000) from King Mongkut's University of Technology Thonburi and Ph.D. (2009) from Asian Institute of Technology. She is a Asst. Prof. in a school of industrial engineering, Suranaree University of Technology.

Generate an Adaptive De-cubing Automatic Processing for Laminated Object Manufacturing (LOM)

Paphakorn Pitayachaval¹ and Niramol Chaphol²

School of Industrial Engineering, Institute of Engineering
Suranaree University of Technology, SUT
Nakhon Ratchasima 30000, Thailand
e-mail: paphakorn@eng.sut.ac.th¹, niramol1990@gmail.com²

Paramate Horkaew

School of Computer Engineering, Institute of Engineering
Suranaree University of Technology, SUT
Nakhon Ratchasima 30000, Thailand
e-mail: phorkaew@sut.ac.th

Abstract— A de-cubing process are important in the Laminated Object Manufacturing (LOM) technique, creates a physical model directly from 3D CAD model without mold and dies by using laminated material. The de-cubing process is to assign shape of waste material into generally small square shape that can be easily remove in order to reduce time and avoid prototype damages. The adaptive de-cubing process applies proportion of number of black pixels on considering area per number of total pixels on considering area. If the proportion is more than a threshold, then the considering axis is divided. On the other hand, if the proportion is less than the threshold, then the considering axis is skipped. The adaptive process algorithm has been developed base on MATLAB platform. The result shown that the biggest threshold exploded the rough de-cubing and stair layer contour, while the smaller threshold produced the fine de-cubing and smooth layer contour.

Keywords—component; Rapid prototype; Laminate Object Manufacturing; Image processing

I. INTRODUCTION

Rapid prototyping process is a manufacturing process that manufactures products directly from CAD model without mold and die. It was introduced to shorten prototype construction time. This process converts a 3D CAD model to be a stack of 2D contours which are used to generate machine commands to build an object layer by layer. In this process, material is added to produce a part layer by layer. Several techniques have been developed for rapid prototyping process and they can be classified into 3 categories based on the initial state of materials [1] liquid-based, solid-based and powder-based. Stereolithography apparatus (SLA), a popular technique in a liquid-based system, is the first commercial rapid prototyping system. In this technique, UV laser is used to solidify photo-curable liquid polymer to form layers. Selective Laser Sintering (SLS) is a famous technique in a powder-based rapid prototyping system. This technique is similar to SLA but instead of curing liquid polymer with UV laser, CO₂ laser is used in SLS to sinter powder to form a layer. Similarly, 3D printing is also a popular technique in a powder-based rapid prototyping system. Besides using UV laser, glue had been applied. Last group of rapid prototyping system is solid-based. In this group, there are several types of solid material used including wire, roll of laminated sheet and pellets. Popular techniques of this group are Laminated Object

Manufacturing (LOM) and Fused Deposition Modelling (FDM) [2].

To complete a model created by LOM, a tool is used to cut laminated material each layer according to layer contours in which the waste material has to remove as de-cubing process. The de-cubing process is to assign shape of waste material into generally small square shape that can be easily remove in order to reduce time and avoid prototype damages.

This paper presents a process to assign the size of wastes material in the de-cubing process. The adaptive de-cubing process has been established by using proportion of a length of contour section per considering cube boundaries.

II. LITERATURE REVIEW

Based on the process of cutting laminate material in LOM, a defective model is generally occurred during contour cutting step. A method of "cut then bond" had been introduced [6]. The outer area of model contour is cut into equal squares, called uniform de-cube, simultaneously cutting outer contour. For this method, the edging of prototype is quite rough and difficult to separate between model and scrap. Moreover, Hur and Lee [7] proposed the problems occurred during de-cubing process: the cumulative error, which is introduced into the serious problem and the rest of scrap material that sticks with the model especially, a complexity model and a hollow model. For the hollow model, Cho and Lee [8] were proposed an approach to create by using LOM. There are two steps: cut outer contour of model for a concurrent layer then remove the scrap material before bound with a previous layer. This approach presented more accumulative error. In order to reduce the accumulative error, Chiu, Liao and Hou [9] proposed the bridge generation to connect the inner model area and the outer model area. Even though this approach reduce error in X and Y axis but error in Z axis is increase in which the model is bended and twisted, Puthawong, Koomsap and Chansri [10] were proposed the image processing technique has been applied in the adaptive crosshatch toolpath generation. An adaptive crosshatch algorithm has been created three size of crosshatch pattern, small cube, main cube and large cube. The created main pattern is modified according to the layer profiles. If the cube that contain black pixels more than one quarter of its pixels four small cubes are inserted into the considered cube. If the number of black pixels in every four main cubes is

determined. The four cubes will be replaced by the large cube if they all are empty. This algorithm developed and generate by LabVIEW program.

III. AN ADAPTIVE DE-CUBING ALGORITHM

An adaptive de-cubing is to assign shape of waste material into generally small non-uniform square shape that can be easily remove in order to reduce time and avoid prototype damages. The algorithm is proposed, as shown in figure 1. An input is an image of 2D layer contour. Since the contour image presents the redundancies areas that are not related with contour information, a contour bounding box is created to identify a working area. This working area presents both outer and inner areas of particular layer contour. Then, RGB layer contours image will be convert to binary layer contours image, the outer and inner areas are identified and the image pixels of inner area are assigned by black color otherwise white color is applied. This working area is divided into four main uniform cubes based on the half points along X axis and Y axis. Then, an adaptive de-cube calculating is applied on each cube, starting from four main uniform cube.

Each main cube is investigated to calculate adaptive de-cube. The adaptive de-cubing process applies proportion of number of black pixels (BP) on considering area per number of total pixels on considering area. If the proportion is more than a threshold, then the considering axis is divided. On the other hand, if the proportion is less than the threshold, then the considering axis is skipped. A condition for dividing cube using threshold is established in following condition,

//Cutting point on X axis.

Set BP = 0;

FOR (considering each column in a cube)

IF (there is black pixel in considering column)

THEN BP++;

ELSE

BP = BP;

IF BP > T_{th} ;

The cutting point = $\frac{x_n}{2}$

//Cutting point on Y axis.

Set BP = 0;

FOR (considering each row in a cube)

IF (there is black pixel in considering row)

THEN BP++;

ELSE

BP = BP;

IF BP > T_{th} ;

The cutting point = $\frac{y_n}{2}$

When T_{th} is threshold, x_n = number of column in considering cube, y_n = number of row in considering cube.

This condition is applied until there is sub-cube contains pixel less than threshold. When the particular cube is assigned to divide, the cutting points on X axis and Y axis are recorded.

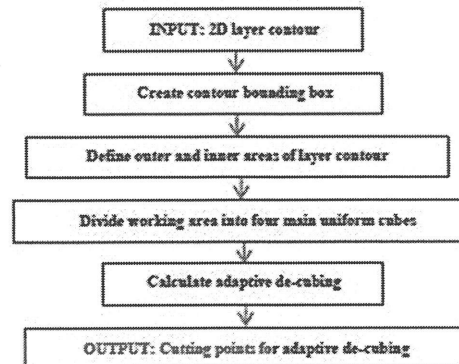


Figure 1. Adaptive de-cubing algorithm

IV. CASE STUDY

To implement the adaptive de-cube algorithm, generate by MATLAB program. A case study model have a diverse shapes, by obtaining layer contours image. Then, RGB layer contours image will be convert to binary layer contours image. In order to be able to separate the outer and inner areas are identified and the image pixels of inner area are assigned. Contour bounding box concept were applied, the working area and both inner area and outer area were established, respectively. After working area was define, the working area was divided into four main uniform cubes. Then, the adaptive de-cube was applied for all main cubes. Each main cube is investigated to calculate adaptive de-cube. From MATLAB program, the difference threshold are assigned as 25% 50% and 75%, respectively. The results were presented, as show in case study model I and II.

A variety of case study model was studies, as show in Figure 2, 6 and 10. The program was calculated until there was all sub-cubes contain pixel less than 4 pixels. The adaptive de-cubing was created, there are differently pattern following by difference threshold.

A. Case study model I



Figure 2. Layer contours of case study model I

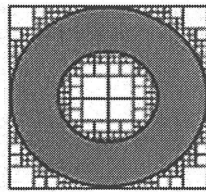


Figure 3. The adaptive ds-cubes of layer contours for case study I for threshold 0.25%

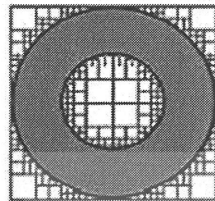


Figure 4. The adaptive ds-cubes of layer contours for case study I for threshold 0.50%

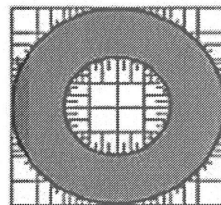


Figure 5. The adaptive ds-cubes of layer contours for case study I for threshold 0.75%

TABLE I. RESULTS COMPARING FOR CASE STUDY I

Result	Threshold		
	25%	50%	75%
Total cutting length (pixel)	1345	1212	838
Cutting length on X axis (pixel)	674	569	406
Cutting length on Y axis (pixel)	671	643	432
Total cutting points (time)	144	127	91
Cutting point on X axis (time)	71	58	43
Cutting point on Y axis (time)	73	69	48

B. Case study model II

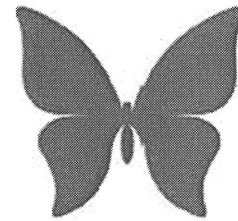


Figure 6. Layer contour of case study model II

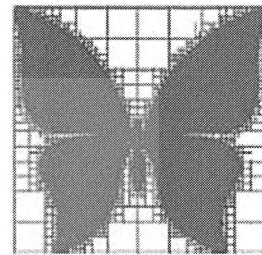


Figure 7. The adaptive ds-cubes of layer contour for case study II for threshold 0.25%

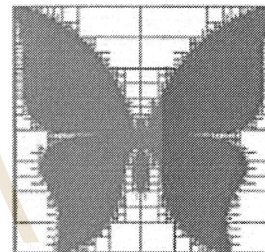


Figure 8. The adaptive ds-cubes of layer contour for case study II for threshold 0.50%

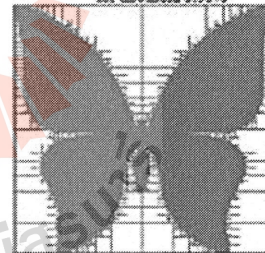


Figure 9. The adaptive ds-cubes of layer contour for case study II for threshold 0.75%

TABLE II. RESULTS COMPARING FOR CASE STUDY II

Result	Threshold		
	25%	50%	75%
Total cutting length (pixel)	4604	3588	2677
Cutting length on X axis (pixel)	2443	2149	1800
Cutting length on Y axis (pixel)	2161	1439	877
Total cutting points (time)	332	257	192
Cutting point on X axis (time)	165	139	119
Cutting point on Y axis (time)	167	118	73

The results showed the variety size of waste material, cutting length (pixel) and cutting points (time). For case study I, if the threshold was changed from 25% to 50%, then the total cutting length were decreased 9.89% and cutting points also were decreased 11.81%. If the threshold was changed 25% to 75%, then the total cutting length were decreased 37.69% and cutting points were decreased 36.81%. For case study II, if the threshold was changed from 25% to 50%, then the total cutting length were decreased 22.07% and cutting points also were decreased 22.59%. If the threshold was changed 25% to 75%, then the total cutting length were decreased 41.85% and cutting points were decreased 42.17%.

In addition, the decreased of cutting length and cutting points reduce tool path and time of de-cubing process. On the other hand, if threshold is increased, the rough surface of prototype and the stair layer contour were occurred.

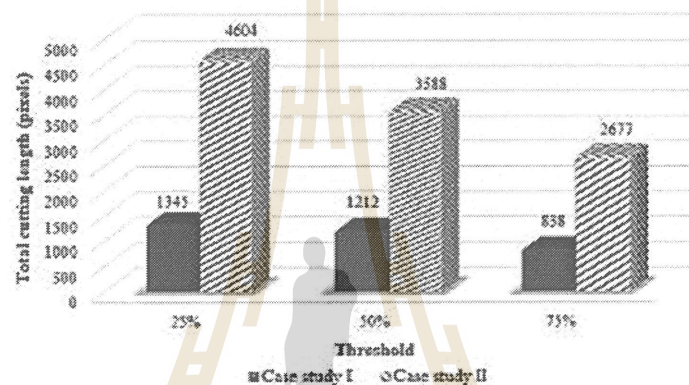


Figure 10. Comparing total cutting length for case study model I and II

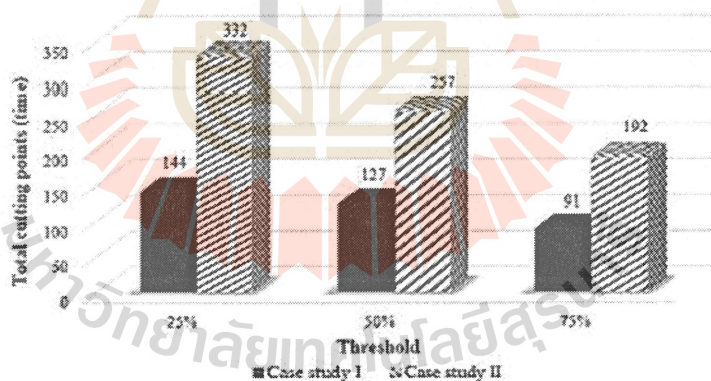


Figure 11. Comparing total cutting points for case study model I and II

V. CONCLUSIONS

The adaptive de-cubing process has been established by using proportion of number of black pixels (BP) on considering axis per number of total pixels on considering axis. The variety sizes of waste material were established in order to easy remove waste from model edge. By using this algorithm, the automatically identify threshold comparing with the model error was generated by MATLAB program. Finally, the output of this process are cutting points that can be used to generate tool path planning for adaptive de-cubing process.

ACKNOWLEDGMENT

This research has been supported by the research and development supporting fund, Suranaree University of Technology.

REFERENCES

- [1] D. T. Pham, and R. S. Gani, "A comparison of rapid prototyping technologies," *International Journal of machine tools and manufacture*, vol. 38, no. 10, pp. 1257-1267, 1998.
- [2] V. Kumar, and D. Dutta, "An assessment of data formats for layered manufacturing," *Advances in Engineering Software*, vol. 28, no. 3, pp. 131-164, 1997.
- [3] P. Soenawan, and P. Koomsap, "Towards: direct transformation of orthographic-view drawings into a prototype," *Virtual and Physical Prototyping*, vol. 4, no. 2, pp. 75-90, 2009.
- [4] W. Cao, and Y. Miyamoto, "Direct slicing from AutoCAD solid models for rapid prototyping," *The International Journal of Advanced Manufacturing Technology*, vol. 21, no. 10-11, pp. 739-742, 2003.
- [5] P. Pinyacharal and T. Baothong, "A Tool Path Planning of Rapid Prototyping Technology Base on a Contour Relationship," *UBU Engineering Journal*, vol. 6, no. 1, pp. 61-69, 2013.
- [6] C. L. Thomas, and K. J. Hayworth, "Automating sheet-based fabrication: the Conveyed-AdherentTM process," pp. 281-290.
- [7] J. Hur, and K. Lee, "Efficient algorithm for automatic support structure generation in layered manufacturing," pp. 18-22.
- [8] I. Cho, K. Lee, W. Choi, and Y.-A. Song, "Development of a new sheet deposition type rapid prototyping system," *International Journal of Machine Tools and Manufacture*, vol. 40, no. 12, pp. 1813-1829, 2000.
- [9] Y. Chiu, Y. Liao, and C. Hou, "The Study of On-line Waste Material Removal Procedures for Bridge Laminated Object Manufacturing (LOM) Process."
- [10] S. Pantharwong, P. Koomsap, and N. Chanai, "Image-based adaptive crosshatch toolpath generation for laminated object manufacturing: This paper propose an algorithm for preparation of mapped layer image, placement of small and large tiles, and avoidance of uncut area," *Virtual and Physical Prototyping*, vol. 9, no. 4, pp. 233-249, 2014.



ประวัติผู้เขียน

นางสาวนิรมล ชาพล เกิดเมื่อวันที่ 6 พฤษภาคม พ.ศ. 2533 เริ่มศึกษาชั้นประถมศึกษาปีที่ 1 โรงเรียนบ้านหนองปรึก อำเภอโชคชัย จังหวัดนครราชสีมา ศึกษาชั้นประถมศึกษาปีที่ 2-6 โรงเรียนบ้านหัวทะเล อำเภอเมือง จังหวัดนครราชสีมา ศึกษาชั้นมัธยมศึกษาปีที่ 1-6 โรงเรียนบุญวัฒนา อำเภอเมือง จังหวัดนครราชสีมา และสำเร็จการศึกษาระดับปริญญาวิศวกรรมศาสตรบัณฑิต (วิศวกรรมอุตสาหการ) สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา ในปี พ.ศ. 2556 ปี พ.ศ. 2557 เข้าศึกษาต่อในระดับปริญญาโททางวิศวกรรมศาสตร์ (วิศวกรรมอุตสาหการ) มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา โดยขณะศึกษาได้รับทุนผู้ช่วยวิจัยในโครงการวิจัย เรื่องการศึกษาการวางแผนการเคลื่อนที่ของเครื่องตัดในการสร้างชิ้นงานต้นแบบรวดเร็วด้วยกระบวนการ Laminated Object Manufacturing และในระหว่างการศึกษาได้ปฏิบัติงานเป็นผู้สอนปฏิบัติการรายวิชาปฏิบัติการกรรมวิธีการผลิตและปฏิบัติการวิศวกรรมอุตสาหการขั้นพื้นฐาน สาขาวิชาวิศวกรรมอุตสาหการ มหาวิทยาลัยเทคโนโลยีสุรนารี

