

เครื่องสร้างแบบจำลอง 3 มิติสำหรับส่วนต่อขาของคนไข้ขาพิการ
3D SCANNER FOR LIMB PROSTHETIC PATIENT

แทนชนก สุขเกษม
THANCHANOK SUKKASEM

เศรษฐพัทธ์ คำพลอยเขียว
SRETTTHAPHAT KAPLOYKEO

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมชีวการแพทย์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2558

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องสร้างแบบจำลอง 3 มิติสำหรับส่วนต่อขาของคนไข้ขาพิการ

3D SCANNER FOR LIMB PROSTHETIC PATIENT

โดย



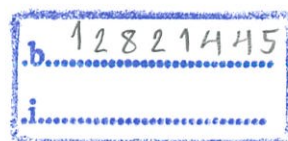
T144662

นาย แทนชนก สุขเกษม
นาย เศรษฐพัทธ์ คำพลอยเขียว

ส.พ.
ท.ร.ค
๒๐๕๘

อาจารย์ที่ปรึกษา

เลขหมู่.....144662 รศ.ดร.ชูชาติ ปิณฑวิรุจน์
เลขทะเบียน.....29 ๗๑. 2559
วัน,เดือน,ปี.....



ปริญญานิพนธ์เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตบัณฑิต

สาขาวิชาวิศวกรรมชีวการแพทย์

คณะวิศวกรรมศาสตร์

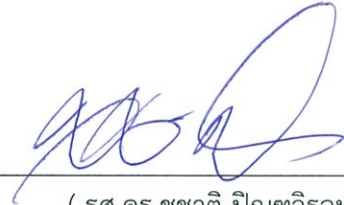
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2558

ปริญญาานิพนธ์ ปีการศึกษา 2558
สาขาวิชา วิศวกรรมชีวการแพทย์
คณะ วิศวกรรมศาสตร์
เรื่อง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เครื่องสร้างแบบจำลอง 3 มิติสำหรับส่วนต่อขาของคนไข้ขาพิการ
3D SCANNER FOR LIMB PROSTHETIC PATIENT

ผู้จัดทำ นายแทนชนก สุขเกษม รหัสประจำตัว 55010467
นายเศรษฐ์พัทธ์ คำพลอยเขียว รหัสประจำตัว 55011256

ปริญญาานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว



(รศ.ดร.ชูชาติ ปิณฑวิรุจน์)
อาจารย์ที่ปรึกษา

หัวข้อโครงการ	เครื่องสร้างแบบจำลอง 3 มิติสำหรับส่วนต่อขาของคนไข้ขาพิการ		
นักศึกษา	นายแทนชนก	สุขเกษม	รหัสนักศึกษา 55010467
	นายเศรษฐพัทธ์	คำพลอยเขียว	รหัสนักศึกษา 55011256
ปริญญา	วิศวกรรมศาสตร์บัณฑิต		
สาขาวิชา	วิศวกรรมชีวการแพทย์		
ปีการศึกษา	2558		
อาจารย์ที่ปรึกษาปริญญาโท	รศ.ดร.ชูชาติ ปิณฑวิรุจน์		

บทคัดย่อ

งานวิจัยนี้จัดทำขึ้นเพื่อการศึกษาทางด้านประมวลผลภาพ, งานทางด้านกลศาสตร์ และการใช้งานระหว่างไมโครคอนโทรลเลอร์ เพื่อใช้สำหรับการออกแบบอุปกรณ์กายเทียมที่มีความเหมาะสมสำหรับผู้ป่วย ผู้ที่ซึ่งมีปัญหาในการดำรงชีวิต เช่น การเดิน การขึ้นบันได และอื่นๆ ทำให้การใช้งานอุปกรณ์กายเทียมที่มีความเหมาะสมนั้นมีความสำคัญ ช่วยให้การใช้ชีวิตสะดวกขึ้น ดังนั้นทางกลุ่มจึงได้เริ่มงานวิจัยและพัฒนา งานวิจัยนี้มีแนวคิดพื้นฐานมาจากการทำงานของเครื่องเอ็กซเรย์โทโมกราฟี(CT scanner) เป็นการใช้การโปรเจกชันของภาพที่ได้มาจาก การถ่ายภาพมาจากวัตถุที่สนใจ(หรือส่วนของตอขาที่ต้องการนำมาทำแบบจำลอง) กล้องที่ใช้เป็นกล้องโมดูลราซเบอร์รี่พาย ใช้จำนวน 5 ตัว ติดตั้งอยู่ในแนวแกนหมุนของเครื่องสแกนเนอร์ ทำการถ่ายภาพแต่ละจากองศาเป็นจำนวนที่ได้กำหนดไว้ โดยทำงานร่วมกับ อาดูยโน(Arduino Uno) ควบคุมการทำงานของระบบถ่ายภาพและหมุนเป็นจังหวะให้เป็นระบบ จากนั้นนำภาพถ่ายทั้งหมดไปยังส่วนประมวลผลภาพทำการปรับค่าให้มีความเหมาะสม ขั้นสุดท้ายนำภาพถ่ายทั้งหมดเข้าสู่กระบวนการสร้างภาพสามมิติจะได้แบบจำลองสามมิติออกมา สามารถนำไปขึ้นรูปเข้าขา หรือวิเคราะห์ความผิดปกติต่อไปได้

กระบวนการทั้งหมดเป็นการทำงานทั้งในด้านกลศาสตร์, การประมวลผลภาพ และการทำงานระหว่างไมโครคอนโทรลเลอร์ เพื่อที่จะได้แบบจำลองที่มีคุณภาพสูง ใช้เวลาน้อยที่สุด และสามารถนำไปใช้ได้จริง

Project Title	3D Scanner for limb prosthetic patient	
Student	Mr. Thanchanok Sukkasem	Student ID 55010467
	Mr. Sretthaphat Kaploykeo	Student ID 55011256
Degree	Bachelor of Engineering	
Program	Biomedical Engineering	
Year	2015	
Thesis Advisor	Assoc.Prof.Dr.Chuchart Pintavirooj	

Abstract

The purposes of this research is to study and apply image processing, mechanism and micro-controller to design suitable limb-prosthetic for disability who might have been in trouble in living such as walking, climbing and so on. They should have used the perfect limb-prosthetic for individual and to do that they need the machine which can assist them. Thus we have started and developed this project. The research's method based on CT scanner procedure started with photography of all projection of objects (or remaining limbs after limb-amputation) by multi-Raspberry Pi cameras, which were attached on rotation plane, on three-dimension scanner machine. They were capturing images from all cameras every degree was controlled by Arduino-R3 Uno. Then, all of images were accumulated and wirelessly transmitted to imaging process unit, imaging program that on the computer. After that adjust the images to correct aberration, improve resolution and then project all data to three-dimension raw data for reconstruction. The last step was three-dimension reconstruction, all data were reconstruct to surface model and exported as stl for forming limb-prosthetic.

These process is an association of mechanical movement, image processing and micro-controller to achieve our goal; maximum accuracy, highest resolution and minimal time to model limb-prosthetic. The results of the project were as follows : 1) The 3D model is higher accuracy and higher resolution than traditional forming, according to original objects scales by scales. 2) It is able to practice with disability to forming the limb-prosthetic, more suitability and comfortable. 3) Using less time, just few seconds to capturing all the projection, so it also able to process quickly and reduce an error at least.

กิตติกรรมประกาศ

การทำวิทยานิพนธ์นี้สามารถสำเร็จลุล่วงไปได้ด้วยดีเนื่องจากได้รับความอนุเคราะห์และคำปรึกษาจากบุคคลหลายท่าน ตลอดทั้งให้ความรู้ในการทำโครงงานแก่คณะผู้จัดทำมาโดยตลอด คณะผู้จัดทำขอขอบพระคุณหลักสูตรวิศวกรรมชีวการแพทย์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่สนับสนุนเครื่องมือ สถานที่และงบประมาณในการศึกษา ออกแบบ สร้างและทดสอบต้นแบบเครื่องในวิทยานิพนธ์นี้

คณะผู้จัดทำขอขอบพระคุณ รศ.ดร. ชูชาติ ปิณฑวิรุจน์ อาจารย์ที่ปรึกษาโครงงาน ที่ให้ความ สนับสนุน แนะนำแนวทาง ให้ความกรุณาตรวจผลงานและแก้ไขข้อบกพร่องในเอกสารและชิ้นงานให้มีความสมบูรณ์มากยิ่งขึ้น รวมถึงให้ความช่วยเหลือในเรื่องต่างๆอย่างเต็มที่ ขอขอบพระคุณอาจารย์ประจำหลักสูตรที่ถ่ายทอดวิชา ความรู้ต่างๆแก่คณะผู้จัดทำ

สุดท้ายนี้ขอขอบพระคุณ บิดา มารดาที่เลี้ยงดู อบรมสั่งสอนและให้การศึกษาย่างดี ทำให้คณะผู้จัดทำมีวันนี้ได้ และยังให้คำปรึกษาในทุกๆด้าน รวมถึงเป็นแรงผลักดันให้คณะผู้จัดทำในการทำสิ่งต่างๆให้ประสบความสำเร็จ

แทนชนก
เศรษฐพัฒน์

สุขเกษม
คำพลอยเขียว

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป.....	VIII
บทที่ 1.....	1
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 สมมติฐานของการศึกษาวิจัย.....	1
1.4 ขอบเขตการวิจัย	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2.....	3
ทฤษฎีและความรู้พื้นฐานที่เกี่ยวข้อง.....	3
2.1 พยาธิสภาพของคนไข้แขนขาพิการ	3
2.1.1 คนไข้ขาพิการ.....	3
2.1.2 คนไข้แขนพิการ	4
2.1.3 คนไข้ที่มีมือขาดบางส่วน ไม่กระชับกับมือเทียม	4
2.1.4 สำหรับการตัดบริเวณข้อศอก.....	4
2.2 เอนโคเดอ์ (Encoder)	5
2.3 การสร้างภาพ (Image Reconstruction).....	6
2.4 แบบจำลองสามมิติ (3D Modeling)	6
2.4.1 การสร้างภาพ 3 มิติเชิงพื้นผิวด้วยอัลกอริทึมมาร์ชชิงคิวบ์ (Marching Cube).....	8
2.4.2 แบบจำลองการให้แสง (Illumination Model).....	11
2.4.3 การประมาณเวกเตอร์นอร์มอล (Normal Vector Approximation).....	13

2.4.4 การแรเงา (Shading Model).....	14
2.4.5 การสร้างภาพ 3 มิติเชิงปริมาตร	14
2.5 การประมวลผลข้อมูลบนระบบเครือข่าย.....	20
2.5.1 ชนิดของสัญญาณอิเล็กทรอนิกส์.....	20
2.5.2 รหัสที่ใช้แทนข้อมูลในการสื่อสาร.....	20
2.5.3 ระบบสื่อสารแอนะล็อก	21
2.5.4 การมอดูเลต.....	21
2.5.5 ระบบสื่อสารดิจิทัล	22
2.5.6 รูปแบบการมอดูเลตสัญญาณดิจิทัล	22
2.5.7 รูปแบบของการส่งข้อมูล.....	22
2.6 ระบบเครือข่ายแบบเบสแบนด์และบรอดแบนด์	28
2.6.1 สถาปัตยกรรมของระบบเครือข่าย	29
2.6.2 มาตรฐานการสื่อสารข้อมูล	32
2.7 การประยุกต์ใช้งานของระบบเครือข่ายคอมพิวเตอร์	32
2.7.1 รูปแบบการใช้งานของระบบเครือข่ายคอมพิวเตอร์.....	32
2.7.2 การแบ่งปันการใช้ทรัพยากรของระบบเครือข่าย	35
2.8 การประยุกต์ใช้งานเครือข่ายอินเทอร์เน็ต	35
2.8.1 การตั้งชื่อบนระบบเครือข่ายอินเทอร์เน็ต	35
2.8.2 จดหมายอิเล็กทรอนิกส์.....	37
2.8.3 การรับส่งแฟ้มข้อมูลบนอินเทอร์เน็ต	38
2.9 การได้มาของรูปภาพ (Image Acquisition)	38
2.9.1 รูปที่ได้มาจากความเข้มของแสง (Intensity Images).....	38
2.9.2 การจับรูปภาพแบบต่อเนื่อง (Real-time Capture)	39
2.9.3 รูปภาพแบบสี (Color Images)	39
2.9.4 รูปภาพแบบเป็นช่วง (Range Images).....	39
2.9.5 ระบบกล้องวิดีโอ.....	40
2.9.6 กล้องวิดีโอแบบขาวดำ.....	40

2.9.7 กล้องวิดีโอแบบสี	40
2.9.8 การจับภาพ (Capture).....	41
2.9.9 การแปลงสัญญาณจากอนาล็อกไปยังสัญญาณดิจิทัล	41
2.6.10 สแกนเนอร์ (Scanners)	41
2.10 การลบภาพพื้นหลัง (Background Subtraction).....	42
2.11 แรสบेอร์รี่พาย(Raspberry Pi).....	43
2.11.1. การกำเนิดแรสบेอร์รี่พาย.....	43
2.11 2.จุดเชื่อมต่ออุปกรณ์และความสามารถของแต่ละตำแหน่งบนบอร์ดแรสบेอร์รี่พาย	44
2.11.3 ข้อดีและข้อเสียของแรสบेอร์รี่พาย.....	45
2.11 4.พอร์ตจีพีไอโอของแรสบेอร์รี่พาย (Raspberry Pi GPIO).....	46
2.11 5.ติดตั้งระบบปฏิบัติการ	46
2.12 โปรแกรมคอมพิวเตอร์	50
2.12.1 การเขียนโปรแกรมคอมพิวเตอร์ (Computer programming).....	50
2.12.2 ภาษาที่ใช้เขียนโปรแกรม.....	50
2.12.3 ไลบรารีโอเพนซีวี (OpenCV Library).....	59
บทที่ 3.....	59
ระเบียบวิธีวิจัยขั้นตอนการดำเนินงานแผนการดำเนินงาน	60
3.1 การทำงาน	60
3.2 การออกแบบ	60
3.1.1 การออกแบบด้านฮาร์ดแวร์.....	60
3.1.2 การออกแบบด้านซอฟต์แวร์	61
3.3 เครื่องมือที่ใช้ในการทดลอง.....	64
3.4 ขั้นตอนการดำเนินงาน	64
บทที่ 4.....	65
การทดลองและผลการทดลอง	65
4.1 ผลการทดลองเก็บภาพลูกปิงปอง.....	65
4.2 ผลการทดลองภาพที่ถูกปรับเทรซโซ.....	66

4.3 ผลการทดลองภาพหลังจากทำการลบภาพพื้นหลัง	67
4.4 ผลการทดลองการสร้างภาพตัดขวาง	69
4.5 ผลการทดลองโมเดลสามมิติ	70
4.6 ผลเปรียบเทียบความคลาดเคลื่อนของวัตถุจริงกับวัตถุที่ได้จากการสร้างภาพสามมิติ	71
บทที่ 5.....	72
สรุปผลการวิจัยและข้อเสนอแนะ	72
เอกสารอ้างอิง.....	73
ภาคผนวก.....	74

สารบัญรูป

รูปที่	หน้า
2.1 เฟสของเอนโคดเดอร์ที่เชื่อมกัน 90 องศา.....	5
2.2 ส่วนประกอบของออปติคอลเอนโคดเดอร์.....	6
2.3 ภาพตัดขวางที่ได้จากเครื่องเอ็กซเรย์ โทโมกราฟีหรือเครื่องตรวจวินิจฉัยโรคด้วยคลื่นแม่เหล็กไฟฟ้า.....	7
2.4 เชนพื้นผิวและเชิงปริมาตรแสดงในรูป.....	7
2.5 หน่วยข้อมูลเล็กที่สุดของภาพและหน่วยข้อมูลเล็กที่สุดของข้อมูลเชิงปริมาตร.....	7
2.6 ตัวอย่างข้อมูลมาร์ชชิงเซลล์.....	8
2.7 มาร์ชชิงคิวบ์.....	9
2.8 รูปแบบของมาร์ชชิงคิวบ์.....	9
2.9 การอ้างถึงลูกบาศก์.....	9
2.10 การคำนวณตำแหน่งที่ไอโซเซอร์เฟสตัดลูกบาศก์.....	10
2.11 ตัวอย่างการตัดไอโซเซอร์เฟสรูปแบบที่ 2.....	11
2.12 การสะท้อนแสงดิฟฟิวส์.....	12
2.13 การสะท้อนแสงสเป็คคิวลาร์.....	12
2.14 ผลของแบบจำลองการให้แสงแบบต่างๆ.....	13
2.15 แบบจำลองการให้เงา.....	15
2.16 การโปรเจกชันทางเดินแสง.....	16
2.17 การโปรเจกชันทางเดินแสงแบบขนานและการสุ่มข้อมูลความเข้ม(มองในระนาบเดียว.....	16
2.18 การอินเตอร์โพลเลชันแบบเชิงเส้นในสามมิติ.....	16
2.19 การแบ่งส่วนภาพ(Segmentation).....	5
2.20 การกำหนดค่าที่บ่งแสง.....	5
2.21 การอินเตอร์โพลเลชันสามมิติของค่าสีตามแนวลำแสง สามารถทำได้สองวิธีคือ.....	19
2.22 การประกอบภาพ.....	20
2.23 ตัวอย่างการส่งรหัสแอสกีจากคอมพิวเตอร์ไปยังเครื่องพิมพ์.....	21
2.24 รูปแบบของการสื่อสารในการรับส่งสัญญาณ.....	21
2.25 การส่งข้อมูลแบบขนาน.....	23
2.26 การส่งข้อมูลแบบอนุกรม.....	23
2.27 การสื่อสารแบบอะซิงโครนัสที่ไม่ได้ใช้พาร์ตีบิต.....	24
2.28 การสื่อสารแบบอะซิงโครนัสที่ใช้พาร์ตีบิต.....	5
2.29 อักขระซิง.....	25
2.30 ข้อผิดพลาดในการใช้อักขระซิง.....	25
2.31 ตัวอย่างการใช้อักขระซิง 2 ตัวในการสื่อสารแบบซิงโครนัส.....	25
2.32 แสดงการตัดแฉวของบิตออกเป็นกลุ่มๆ ละ 8 บิต.....	26
2.33 การส่งผ่านข้อมูลแบบซิงโครนัส.....	26
2.34 การส่งผ่านข้อมูลแบบอะซิงโครนัส.....	26
2.35 การส่งข้อมูลแบบทิศทางเดียว.....	28
2.36 การส่งข้อมูลแบบทิศทางสลับกัน.....	28

2.37 การส่งข้อมูลสองทิศทางส่งได้พร้อมกัน.....	28
2.38 โทโปโลยีแบบบัส.....	29
2.39 โทโปโลยีรูปวงแหวน.....	30
2.40 โทโปโลยีแบบดาว.....	31
2.41 โทโปโลยีแบบผสม.....	32
2.42 ระบบเครือข่ายแบบรวมศูนย์กลาง.....	33
2.43 ระบบเครือข่ายแบบเพียร์ทูเพียร์.....	34
2.44 ระบบเครือข่ายแบบโคลเอนท์/เซิร์ฟเวอร์.....	34
2.45 การทำงานของโดเมนเนม.....	36
2.46 แสดงการใช้กระบวนการจับภาพของวิดีโอและการแสดงผล.....	39
2.47 โครงสร้างของกล่องวิดีโอ.....	40
2.48 ภาพตัดของกล่องวิดีโอ.....	40
2.49 แสดงอุปกรณ์ที่ใช้ในการแปลงสัญญาณ.....	41
2.50 การทำลบบภาพพื้นหลังและการปรับเทรซโซ.....	42
2.51 ส่วนประกอบของแรสเบอร์รี่พาย.....	44
2.52 พอร์ตจีพีไอโอของแรสเบอร์รี่พาย.....	46
2.53 แอลอีดีแสดงสถานะของแรสเบอร์รี่พาย.....	46
2.54 ตัวอย่างเม็มโมรีการ์ดที่ใช้ติดตั้งระบบปฏิบัติการ.....	47
2.55 โปรแกรม SD Formatter V4.0.....	47
2.56 โปรแกรมWin32 Disk Imager.....	48
2.57 การแตกไฟล์ชิพของไฟล์ระบบปฏิบัติการแรซเบียน.....	48
2.58 การแสดงผลลักษณะการเชื่อมต่อเครือข่าย.....	49
2.59 แสดงการตั้งค่าไอพีแอตเดรสบนแรสเบอร์รี่พาย.....	49
3.1 ขั้นตอนการทำงานของเครื่องสร้างแบบจำลอง 3 มิติสำหรับส่วนต่อขาของคนไข้ขาพิการ.....	60
3.2 โฟลวชาร์ตไดอะแกรมของการออกแบบซอฟต์แวร์.....	62
4.1 การเก็บภาพจากกล่องแรสเบอร์รี่พาย.....	66
4.2 ภาพที่ทำการปรับเทรซโซ.....	67
4.3 ภาพที่ทำการลบพื้นหลัง.....	68
4.4 ภาพที่ได้หลังจากทำการตัดขวาง.....	70
4.5 ภาพสามมิติจากลูกบิ๊งโปง.....	70

สารบัญตาราง

ตารางที่	หน้า
4.1 แสดงผลความคลาดเคลื่อนของวัตถุตัวอย่างได้แก่ ลูกปิงปอง วัตถุทรงสี่เหลี่ยม และด้ามจับแรสเบอร์รี่พาย โดยจะเป็นการเปรียบเทียบโดยการวัดปริมาตรของวัตถุจริงกับวัตถุที่ได้จากการสร้างภาพจำลองสามมิติ.....	71

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในการเยียวยารักษาคนไข้แขนขาพิการ จะทำการวินิจฉัยและการวิเคราะห์แนวทางการรักษาแก่คนไข้แขนขาพิการเป็นรายบุคคล ซึ่งต้องใช้บุคคลากรที่เชี่ยวชาญ มีความซับซ้อนของขั้นตอนต่างๆ รวมทั้งสรีระของคนไข้แต่ละราย ทำให้การรักษาโดยการสวมใส่อวัยวะเทียมแก่คนไข้แขนขาพิการ ในบางกรณีอาจเกิดปัญหาขึ้นได้ เช่น ความไม่พอดีของอุปกรณ์ที่สวมใส่กับคนไข้, อุปกรณ์เสริมที่ต้องมีการออกแบบเป็นพิเศษ หรือแม้แต่ศักยภาพของสถานพยาบาลแห่งนั้นที่ไม่อาจจะไม่มีศักยภาพที่เพียงพอในการตอบสนองต่อความจำเพาะเจาะจงแก่คนไข้ ทำให้คนไข้ไม่ได้รับการรักษาหรือได้รับอวัยวะเทียมที่ไม่เหมาะสม อาจจะทำให้เกิดปัญหาต่างๆตามมา รวมทั้งคุณภาพชีวิตของคนไข้อีกด้วย

ดังนั้นจากปัญหาข้างต้นดังกล่าวที่อาจเกิดขึ้นได้ ทำให้เกิดการคิดค้นเครื่องมือสร้างแบบจำลองสามมิติที่สามารถทำการวิเคราะห์, การออกแบบ, การจำลองสภาพพื้นผิวรอยต่อ รวมถึงการจำลองแบบสามมิติของอุปกรณ์เสริม ซึ่งมีความจำเพาะเจาะจงแก่คนไข้แขนขาพิการแต่ละราย โดยเป็นการประยุกต์ใช้กระบวนการประมวลผลภาพ(Image Processing) โดยการนำภาพถ่ายแต่ละบริเวณมาผ่านการประมวลผลเป็นแบบจำลองสามมิติที่เสมือนจริง สามารถนำข้อมูลที่ได้ไปออกแบบอุปกรณ์เสริมที่มีความเหมาะสมแก่คนไข้แขนขาพิการในการสวมใส่แขนขาเทียม โดยหวังว่าเครื่องมือนี้จะสามารถช่วยคนไข้แขนขาพิการได้ไม่มากก็น้อย และเป็นการกระตุ้นให้มีการพัฒนาอุปกรณ์การแพทย์ในประเทศให้มากยิ่งขึ้น

1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อการสร้างแบบจำลองภาพสามมิติจากวัตถุหรืออวัยวะ
2. เพื่อการวิเคราะห์บริเวณพื้นผิวรอยต่อจากวัตถุหรืออวัยวะ
3. เพื่อนำความรู้ด้านไมโครคอนโทรลเลอร์และโมดูลกล้องมาประยุกต์ใช้กับเครื่องมือที่สร้างขึ้น
4. เพื่อนำความรู้ด้านสร้างภาพสามมิติ(3D Reconstruction) มาประยุกต์ใช้จากภาพถ่ายที่ได้
5. เพื่อผลักดันให้มีเครื่องมือที่สามารถช่วยคนไข้ได้จริง และคนไข้สามารถเข้าถึงเครื่องมือได้ง่าย

1.3 สมมติฐานของการศึกษาวิจัย

งานวิจัยนี้มุ่งเน้นในการออกแบบเครื่องมือที่สามารถทำงานได้ดี สามารถที่จะสามารถจำลองแบบสามมิติของวัตถุหรือบริเวณส่วนต่างๆของอวัยวะที่ต้องการ โดยใช้เวลาน้อยที่สุด และได้แบบจำลองที่มีคุณภาพ จากการประยุกต์ไมโครคอนโทรลเลอร์, กล้อง และการใช้การประมวลผลทางภาพ

1.4 ขอบเขตการวิจัย

1. เครื่องมือที่สร้างขึ้นนั้นสร้างจากวงล้ออะลูมิเนียมขนาดเส้นผ่านศูนย์กลาง 18 นิ้ว และแผ่นอะคริลิก 18 นิ้ว หนา 4.7 มิลลิเมตร สร้างบนฐานแผ่นไม้อัดแข็งหนา 6 มิลลิเมตร ขับเคลื่อนกลไกด้วยเฟืองขับ 36 ซี่เฟือง เฟืองตาม(แผ่นอะคริลิก 18 นิ้ว ข้างต้น) 360 ซี่เฟือง อัตราทดมอเตอร์กระแสตรงจาก 5000 รอบต่อนาที เป็น 50 รอบต่อนาที
2. ขนาดของวัตถุหรืออวัยวะ เนื่องจากเครื่องมือที่ใช้มีเส้นผ่านศูนย์กลางภายใน 30 เซนติเมตร ซึ่งวัตถุหรืออวัยวะที่ต้องการถ่าย ต้องมีขนาดไม่เกินค่าดังกล่าว รวมทั้งต้องคำนึงถึงระยะโฟกัสของกล้องในการถ่ายภาพ
3. สภาพแวดล้อมโดยรอบ ต้องไม่มีแสงสว่างมากเกินไปเนื่องจากอาจส่งผลกระทบต่อคุณภาพของภาพที่ถ่าย ควรจะมีสภาวะที่เหมาะสม ไม่มีการสั่นสะเทือนจากบริเวณโดยรอบ เนื่องจากอาจมีผลกระทบต่อการทำงานของภาพสามมิติได้ กล้องที่ใช้ถ่ายภาพเพื่อนำไปประมวลผลเป็นโมเดลกล้องแรสเบอร์รี่พาย(Raspberry Pi Camera Module)
4. วัตถุหรืออวัยวะที่ใช้ในการศึกษาวิจัย ใช้ สกรู, ลูกปิงปอง ในส่วนของอวัยวะใช้แบบจำลองพื้นผิวเสมือนจริงของคนไข้แขนขาพิการ เพื่อดูแบบจำลองสามมิติดังกล่าว จากการประมวลผลภาพด้วยการสร้างภาพสามมิติ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 สามารถประยุกต์ใช้ไมโครคอนโทรลเลอร์ อาร์ดูอีนโอ อนุโอ(Arduino Uno) กับ แรสเบอร์รี่พาย(Raspberry Pi) ให้ทำงานร่วมกันได้เป็นอย่างดี
- 1.5.2 สามารถประยุกต์ใช้งานด้านการประมวลผลทางภาพ รวมทั้งการสร้างภาพสามมิติ
- 1.5.3 สามารถจำลองภาพสามมิติที่มีความสมจริง ในเวลาอันสั้น
- 1.5.4 สามารถนำเครื่องมือไปใช้งานได้จริง เพื่อเป็นการช่วยเหลือคนไข้แขนขาพิการ

บทที่ 2

ทฤษฎีและความรู้พื้นฐานที่เกี่ยวข้อง

2.1 พยาธิสภาพของคนไข้แขนขาพิการ

2.1.1 คนไข้ขาพิการ

ส่วนใหญ่ในเวชปฏิบัติทั่วไปพบว่า คนไข้ต้องถูกตัดขา เนื่องจากโรคหลอดเลือดส่วนปลาย เบาหวาน หรือ ภัยอันตราย เมื่อจำเป็นต้องตัดขา ศัลยแพทย์ต้องพยายามเหลือความยาวของขาไว้ ให้มากที่สุดที่จะทำได้เพื่อใส่ขาเทียม ถ้าการตัดเหลือปลายไว้สั้น (Short Stump) คนไข้จะเคลื่อนไหวไม่ได้ดีเท่าที่ควรเมื่อใส่ขาเทียม และท่าเดินจะไม่ดี เหมือนคนไข้ที่การตัดเหลือปลายไว้ยาว (Long Stump) นอกจากนี้ การทำเวชศาสตร์ฟื้นฟูจะได้ผลดีก็เมื่อปลายที่เหลือของขาที่ตัดนั้นจะต้องดีรับกับขาเทียมที่ใส่ ถ้าเป็นไปได้ศัลยแพทย์ต้องพยายามเหลือข้อเข้าไว้ ในรายที่ตัดขาเหนือข้อเข้า ขาเทียมที่ใส่จะต้องมีข้อเข้าด้วย ซึ่งทำให้เพิ่มขนาดขาเทียมและพลังงานที่คนไข้ต้องใช้ ตลอดจนทำให้คนไข้เคลื่อนไหวไม่สะดวกมากขึ้น

ปัจจุบันนี้การตัดขาในโรคหลอดเลือดส่วนปลายหรือเบาหวาน ส่วนมากสามารถเก็บข้อเข้าไว้ได้ ซึ่งผิดกับสมัยเมื่อ 20 ปีก่อน ทั้งนี้เหตุผลที่สำคัญอันหนึ่งคือ เทคนิคหลังการตัดขา โดยใส่เปลือกปูนในแบบเบ้ารอง รับเอ็นสะบ้า (Patellar Tendon Bearing Socket) ทันทีหลังจากเย็บผิวหนังปิด โดยวิธีนี้ทำให้ไม่บวมและมีการดูดระบายเลือดที่ซึมหรือค้างอยู่ในแผลเป็นการป้องกันการบวม ช่วยเร่งให้เนื้อเยื่อหายเร็วยิ่งขึ้น

แบบของการตัดและส่วนประกอบของขาเทียมที่จะใช้ขึ้นอยู่กับบริเวณที่ตัดตามความจำเป็นของโรคหรือ ภัยอันตราย คนไข้ที่ได้รับการตัดนิ้วเท้า พวกนี้ไม่จำเป็นต้องใส่นิ้วเทียมสำหรับการเดิน ถ้าตัดผ่านบริเวณเมตาทาร์ซัล หรือทาร์ซัลควรใช้รองเท้าตัดพิเศษ การตัดผ่านข้อเท้า (Syme Amputation) มีข้อดีคือปลายขา (Stump) เป็นส่วนที่รับน้ำหนัก แต่รูปร่างของขาเทียมอันใหญ่ไม่สวยเหมือนขาที่ตัดได้เข้า ฉะนั้นการตัดขาแบบซัยม์นี้ในหญิงสาวต้องคิดให้ดีกว่าที่จะทำ ขาเทียมแบบซัยม์ (Syme Prosthesis) ก็เช่นเดียวกับขาเทียมแบบอื่นที่ประดิษฐ์ขึ้นเกือบทั้งหมดคือ ไม่มีข้อเท้าและส้นเท้าที่ยึดหยุ่นได้ (Solid Ankle, Cushioned Heel)

ระดับการตัดขาได้เข้านั้นทำได้ตั้งแต่เหนือตาตุ่ม จนถึงต่ำกว่าส่วนบนสุดของทิเบีย (Tibia Plateau) 5 ซม. บริเวณที่ตัดขาได้เข้านี้ควรจะขึ้นอยู่กักระดับผิวหนังและกล้ามเนื้อที่ดี ที่เหมาะสมที่สุดควรเหลือปลายขาได้เข้าไว้ยาว 15 ซม. ขาเทียมที่ใช้ในพวกตัดขาได้เข้าเป็นแบบรองรับที่เอ็นสะบ้า (Patellar Tendon Bearing) โดยเพิ่มสายรัดโคนขา หรืออาจไม่เพิ่มก็ได้ สายรัดโคนขานี้จำเป็นสำหรับคนที่มีปลายขาเหลือสั้นหรืออ่อนนุ่ม โดยสายรัดนี้จะช่วยกระจายการกดรอบๆ ปลายขาและเพิ่มความมั่นคง นอกจากนี้สายรัดยังใช้ช่วยยึดดึงขาเทียมในคนไข้ที่มีปลายขาสั้นมาก ขาเทียม ธรรมดาแบบเก่า (ปลายเปิด) ยังใช้เป็นประโยชน์ได้นานๆ ครั้ง โดยเฉพาะ ในคนไข้พวกที่ปวดปลายขา มีปัญหาทางผิวหนังหรือปลายขาบวม

ระดับการตัดขาเหนือเข่าควรอยู่ระหว่างบริเวณเหนือเข่า 10 ซม. จนถึงต่ำกว่าข้อต่อสะโพก 10 ซม. ระยะ 10 ซม. เหนือเข่าเหมาะกับการทำเข่าเทียม (Mechanical Knee) เข่าเทียมสำหรับคนไข้ที่ตัดขาเหนือเข่านั้น มีหลายแบบที่ใช้บ่อยที่สุดเป็นแกนเดี่ยว สำหรับขาเทียมมีเข่าเป็นแบบ สัมผัสทั้งหมด และบนสุดเป็นรูปสี่เหลี่ยมด้านไม่เท่ากัน ซึ่ง

ทำให้น้ำหนักส่วนใหญ่อยู่ที่ปุ่มอิสเคียล(Ischial Tuberosity) และส่วนที่เหลือกระจายรอบๆปลายขา ขาเทียมติดอยู่ได้โดยใช้หรือไม่ใช้การดูด(Suction) สำหรับเบ้าดูด(Suction Socket) ไม่ควรใช้ในรายที่มีการเปลี่ยนแปลงทางผิวหนัง บวมเฉพาะที่หรือแผลที่ผิวหนัง ทำเดินของคนไข้ที่ตัดขาเหนือเข่าจะไม่ดีเหมือนพวกที่ตัดได้เข่าและพลังงานที่ใช้ในการเดินด้วยขาเทียมเหนือเข่า นั้นมากกว่าพลังงานที่ใช้เดินด้วยขาเทียมได้เข่า นานๆครั้งเราจะพบคนไข้ที่จำเป็นต้องตัดข้อตะโพก(Hip Disarticulation) ตัดครึ่งเชิงกราน(Hemipelvectomy) หรือตัดครึ่งตัว(Hemicorporectomy) แบบเครื่องประดิษฐ์เทียมสำหรับคนไข้เหล่านี้มีตะโพกเทียม คานาเดียน(Canadian Hip Prosthesis) และเครื่องประดิษฐ์เทียมสำหรับพุงเชิงกรานแบบอื่นๆ พลังงานที่ต้องใช้ในการเดินด้วยเครื่องประดิษฐ์เทียมเหล่านี้ ยิ่งมากกว่าพลังงานที่ใช้สำหรับการเดินในพวกตัดขาเหนือเข่า

2.1.2 คนไข้แขนพิการ

จุดประสงค์ของการใช้แขนเทียมก็เพื่อให้คนไข้จับสิ่งของได้ และทำให้รูปร่างครบถ้วน การตัดแขนก็เช่นเดียวกับการตัดขา ควรเหลือความยาวไว้ให้มากเพื่อให้สวมแขนเทียมได้กระชับ โดยทั่วไปคนไข้ที่ได้รับการตัดต่ำกว่าข้อศอกจะใช้งานได้ดีกว่าคนไข้ที่ถูกตัดเหนือข้อศอก

ในแขนเทียมอันหนึ่งนั้นเป็นการยากที่จะประดิษฐ์ให้ใช้งานได้ทั้ง 2 อย่าง คือ ใช้จับสิ่งของและเพื่อความสวยงาม ส่วนใหญ่คนไข้จึงต้องใช้แขนเทียม 2 แบบ แบบหนึ่งใช้เป็นขอ(Hook) จับสิ่งของและอีกแบบหนึ่งเป็นมือเพื่อความสวยงาม โดยไม่ได้ออกแบบเพื่อใช้งาน สำหรับส่วนปลายแขนเทียมซึ่งเป็นขอจับและมือนี่เปลี่ยนกันได้ คนไข้ส่วนมากชอบแบบขอจับมากกว่ามือ เพราะแบบใช้ขอจับเบากว่า ทนทานกว่าและเหมาะสมสำหรับงานที่ต้องใช้มือ เช่นเดียวกับบริเวณขา การประดิษฐ์แขนเทียมต้องแล้วแต่ระดับการตัด

2.1.3 คนไข้ที่มีมือขาดบางส่วน ไม่กระชับกับมือเทียม

คนไข้ที่มีมือขาด ใส่มือเทียมทำงานได้ เนื่องจากยั้งคว้าและหงายมือได้เกือบเต็มที่ ส่วนปลายของมือเทียมและแขนเทียมเกือบทั้งหมด เป็นแบบที่บังคับให้เปิดหรือปิดได้โดยการควบคุมจากการกางกระดูกสะบัก การงอข้อไหล่หรือทั้ง 2 อย่างรวมกัน การเคลื่อนไหวเหล่านี้ติดต่อกับส่วนปลายด้วยสายลวดจากหัวไหล่

ในรายที่ตัดได้ข้อศอกซึ่งเหลือส่วนปลายแขนยาว(Longer Stump) จะกระชับกับมือเทียมเช่นเดียวกับคนไข้ที่ถูกตัดผ่านข้อมือ อย่างไรก็ตาม แขนเทียมแบบมินสเตอร์(Munster Prosthesis) ใช้ได้ดีในคนไข้ที่เหลือปลายแขนได้ ข้อศอกสั้น แขนเทียมแบบนี้เข้ารองรับปลายแขน(Socket) ยื่นไปด้านหลังรอบโอเลครานอนและกระชับชิดรอบๆกล้ามเนื้อไบเซพส์ แขนเทียมยึดเหนี่ยวแบบนี้ ไม่จำเป็นต้องใช้บานพับที่ข้อศอกและรัดกล้ามเนื้อไตรเซพส์ อย่างไรก็ตามคนไข้จะเหยียดและงอข้อศอกได้จำกัด ซึ่งการเสียหน้าที่ดังกล่าวมีความสำคัญน้อยในคนไข้ที่แขนขาดไปข้างเดียว

2.1.4 สำหรับการตัดบริเวณข้อศอก

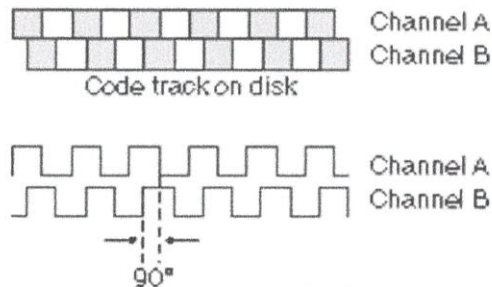
แขนเทียมที่ใส่นั้นทำงานได้ตามต้องการคือบิดเข้าในและบิดออกนอกได้เต็มที่ อย่างไรก็ตาม แขนเทียมที่ใช้ในใหญ่ที่ข้อศอก เนื่องจากต้องเพิ่มเครื่องกันบานพับด้านนอกทำให้น้ำหนักเพิ่ม คนไข้หญิงส่วนมากไม่ชอบที่ข้อศอกมีกลไกกุญแจสลับ คือการเคลื่อนไหว แบบหนึ่งจะจำกัด(Lock) ข้อศอก และการเคลื่อนไหวอีกแบบจะไม่จำกัด(Unlock) ข้อศอก การควบคุมส่วนปลาย และการงอข้อศอกใช้การเคลื่อนไหวอย่างเดียวกับการใช้มือเทียมสำหรับข้อมือขาด เมื่อข้อศอกไม่ถูกจำกัด ข้อศอกเทียมงอได้ แต่เมื่อข้อศอกถูกจำกัด การควบคุมการเคลื่อนไหว อันเดียวกันนี้จะบังคับให้ส่วนปลายทำงาน

ในรายที่ตัดเหนือข้อศอก ใช้แขนเทียมแบบเดียวกับการตัดที่ข้อศอกได้ เพียงแต่ตัดแปลงที่ข้อศอกและรูปร่างของเบ้าที่รองรับ คนไข้เหล่านี้ จำเป็นต้องใช้สิ่งประดิษฐ์เพื่อช่วยในการบิด คนไข้ที่ถูกตัดข้อไหล่ ใช้แขนเทียมซึ่งมีหน่วยพื้นฐานเช่นเดียวกับคนไข้ที่ถูกตัดเหนือข้อศอก แต่เบ้ารองรับต้องคลุมสะบักและกล้ามเนื้อ เพ็คตอรัล

2.2 เอนโคดเดอร์ (Encoder)

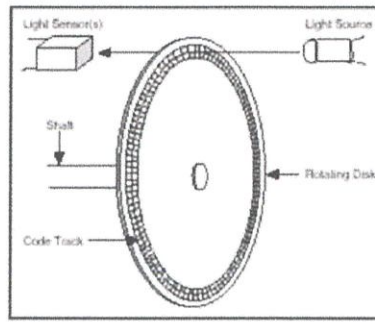
เอนโคดเดอร์เป็นเซนเซอร์ตรวจสอบการหมุน กล่าวได้ว่าการใช้งานหลักๆที่สำคัญส่วนใหญ่ใช้กับการหมุน โดยสามารถตรวจสอบการหมุนได้ 2 แบบคือ เชิงเส้น(Linear) และ เชิงมุม(Angular) แต่ส่วนใหญ่นิยมใช้แบบเชิงมุม หลักการของเอนโคดเดอร์คือเอนโคดเดอร์จะรับสัญญาณเชิงกล จากการหมุนต่างๆ มาแปลให้อยู่ในรูปแบบของจังหวะ (Pulse) ซึ่งเราสามารถนำจังหวะที่ได้ไปใช้งานต่อนั่นเอง ซึ่งเป็นการใช้หลักการของ อิเล็กโทรแมคคานิคอล (Electromechanical) ซึ่งสามารถวัดการเคลื่อนไหวหรือตำแหน่งได้ โดยส่วนใหญ่แล้วจะใช้เป็น ออปติคอลลเซ็นเซอร์ (Optical Sensor) ซึ่งจะทำให้ได้สัญญาณไฟฟ้าซึ่งอยู่ในรูปแบบของพัลส์ซึ่งจะมาเป็นช่วง ซึ่งสามารถแปลงเป็นการเคลื่อนไหว ทิศทางและตำแหน่งได้ ตามลำดับ

พื้นฐานเอนโคดเดอร์แบบควอดเรเจอร์(Quadrature Encoder Fundamentals) เป็นประเภทหนึ่งขอเอนโคดเดอร์ ส่วนใหญ่ในการวัดค่าการเพิ่มขึ้นของเอนโคดเดอร์จะใช้ค่าเอาต์พุตสองค่า คือ A กับ B ในการตอบสนองต่อตำแหน่งที่มีการเคลื่อนที่ โดยการใส่แถบตำแหน่งสองแถบที่มีเฟสต่างกันอยู่ 90 องศา ค่าเอาต์พุตที่ได้จากสองตำแหน่งที่ A กับ B ต่างกันอยู่ 90 องศา นั้นเรียกว่าควอดเรเจอร์เอนโคดเดอร์ ซึ่งสามารถบอกตำแหน่งและทิศทางของการหมุนได้ เช่น ถ้า A นำ B จะเป็นการหมุนแบบตามเข็มนาฬิกา แต่ถ้า B นำ A แสดงว่าเป็นการหมุนแบบทวนเข็มนาฬิกา



รูปที่ 2.1 เฟสของเอนโคดเดอร์ที่เหลื่อมกัน 90 องศา

โรตารีเอนโคดเดอร์(Rotary Encoder) ถูกใช้เพื่อวัดการเคลื่อนไหวแบบหมุนของแกน ดังรูปที่ 2.2 เป็นการแสดงให้เห็นถึงองค์ประกอบพื้นฐานของโรตารีเอนโคดเดอร์ซึ่งประกอบไปด้วยหลอดแอลอีดี(LED), แผ่นดิสก์ และตัวรับรู้แสง ซึ่งอยู่ด้านตรงข้ามของแผ่นดิสก์ ซึ่งแผ่นดิสก์ที่ติดอยู่กับเพลลาของมอเตอร์จะทำการหมุนไป ในส่วนที่เป็นบริเวณที่บของแผ่นดิสก์จะไปขัดขวางทางเดินของลำแสงจากหลอดแอลอีดี ซึ่งจะทำให้ลำแสงที่ปล่อยออกมาไม่สามารถตกกระทบได้ที่ตัวรับรู้แสง เมื่อมีการหมุนอย่างต่อเนื่องจะเกิดสัญญาณเป็นพัลส์สี่เหลี่ยมออกมา สามารถนำไปประมวลผลได้



รูปที่ 2.2 ส่วนประกอบของออปติคัลเอนโคเดอร์

2.3 การสร้างภาพ (Image Reconstruction)

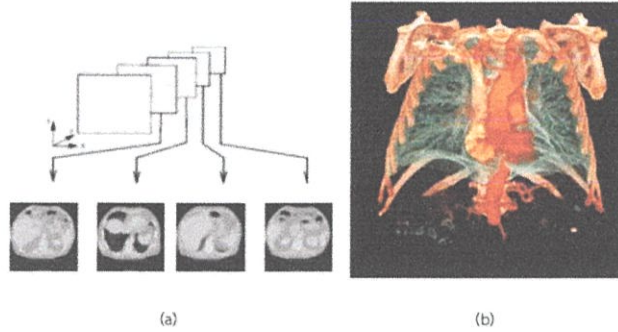
ปัญหาการสร้างภาพตัดขวางคืนจากข้อมูลโปรเจกชัน(Projection) ได้เพิ่มความสำคัญในหลายปีที่ผ่านมาโดยได้มีการนำเอาการสร้างภาพตัดขวางไปประยุกต์ใช้งานหลายอย่าง การประยุกต์ใช้งาน ในทางการแพทย์ที่ก่อให้เกิดการเปลี่ยนแปลงอย่างมากมาในทางรังสีวินิจฉัย คือการนำไปใช้ในเครื่องซีทีสแกนเนอร์(Computerized Tomography หรือ CT) ซึ่งรังสีเอ็กซ์เรย์ถูกนำไปใช้ในการสร้างข้อมูลโปรเจกชันสำหรับการสร้างภาพตัดขวางของร่างกายมนุษย์ การประยุกต์ ใช้งานทางการแพทย์อื่นๆอีก ได้แก่การนำไปใช้ในเวชศาสตร์นิวเคลียร์ ในเวชศาสตร์นิวเคลียร์เทคนิคนี้ได้ถูกนำไปใช้ดูการกระจายของความหนาแน่นของสารกัมมันตภาพรังสีที่ปล่อยรังสีแกมมาในส่วนตัดขวางใดๆของ ร่างกาย คลื่นอัลตราซาวด์ สามารถถูกนำไปใช้ในการสร้างภาพตัดขวางได้เช่นเดียวกับรังสีเอ็กซ์เรย์ อย่างไรก็ตามคลื่นอัลตราซาวด์มีการหักเห เมื่อผ่านเข้าไปในเนื้อเยื่อ ทำให้การสร้างภาพตัดขวางโดย ใช้คลื่นอัลตราซาวด์มีความยุ่งยากมากกว่า การประยุกต์ใช้งานในด้านอื่นที่ไม่ใช่ทางการแพทย์ของ การสร้างภาพตัดขวางสามารถพบได้ในการสำรวจทางภูมิฟิสิกส์(Geophysic Exploration) การใช้ในกล้องจุลทรรศน์อิเล็กตรอน(Electron Microscopy) เป็นต้น

2.4 แบบจำลองสามมิติ (3D Modeling)

ภาพตัดขวางที่ได้จากเครื่องเอ็กซเรย์ โทโมกราฟี(CT) หรือเครื่องตรวจวินิจฉัยโรคด้วยคลื่นแม่เหล็กไฟฟ้า (MRI) เมื่อนำมาเรียงกันจะได้ เป็นข้อมูลเชิง ปริมาตร ดังแสดงในรูป 2.3 (a) การสร้างแบบจำลอง 3 มิติทำการแปลงข้อมูลเชิงปริมาตรไปเป็นภาพ 2 มิติที่สามารถแสดงบนจอมอนิเตอร์ ดังแสดงในรูป 2.3 (b) แนวคิดของการสร้างแบบจำลอง 3 มิติมาจากเครื่องถ่ายภาพเอ็กซเรย์แบบธรรมดา โดยทำการฉายเอ็กซเรย์ทะลุผ่านร่างกายเกิดการลดทอนลงก่อนกระทบลงบนฟิล์มเอ็กซเรย์ ในกรณีนี้ร่างกายของเราเปรียบเสมือนข้อมูลเชิงปริมาตร ส่วนภาพบนฟิล์มเอ็กซเรย์คือภาพ 2 มิติที่สร้างได้ เราเรียกการสร้างแบบจำลอง 3 มิติโดยเปลี่ยนแบบการเกิดภาพจากเครื่องเอ็กซเรย์ว่าเป็นการสร้างภาพเชิงปริมาตร โดยมีการจำลองลำแสงทะลุเข้าไปในข้อมูลเชิงปริมาตรเหมือนการฉายเอ็กซเรย์แสงที่ทะลุผ่านข้อมูลเชิงปริมาตรจะถูกโปรเจกชันลงบนระนาบทำให้เกิดภาพ 2 มิติ มีการสร้างแบบจำลอง 3 มิติในอีกลักษณะหนึ่งที่คิดค้นโดยรอลเลนเซน(Lorenzen) และคลีน(Cline) ซึ่งเป็นการสร้างแบบจำลองเชิงพื้นผิวอัลกอริทึมที่ใช้รู้จักกันในชื่อ มาร์ชชิ่งคิวบ์(Marching Cube) โดยทำการสร้างภาพพื้นผิวโดยประมาณของข้อมูลที่มีค่าความเข้มเท่ากันที่เรียกว่าไอโซเซอร์เฟซ(Iso-surface) ภาพที่ได้เปรียบเสมือนการติดกระเบื้องรูปสามเหลี่ยมที่มีขนาดต่างๆกันลงไปทั่วทั้งไอโซเซอร์เฟซ ขนาดของกระเบื้องสามารถปรับขนาดได้ ขนาดที่เล็กมีค่าผิดพลาดจากการประมาณน้อยลง

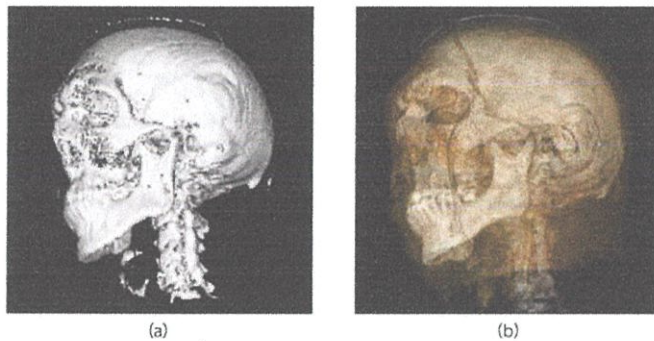
นอกจากนั้นทั้งการสร้างภาพเชิงปริมาตรและเชิงพื้นผิวสามารถจำลองการให้แสงและเงาซึ่งทำให้รูปแบบ 3 มิติที่ได้มีความสมจริงมากขึ้น ตัวอย่างเชิงพื้นผิวและเชิงปริมาตรแสดงในรูป 2.4

ในการอธิบายถึงการสร้างแบบจำลอง 3 มิติ ต้องมีการอ้างถึงหน่วยที่เล็กที่สุดของข้อมูลเชิงปริมาตรที่เรียกว่า ว็อกเซล (Voxel) ดังแสดงในรูป 2.5 โดยเปรียบเทียบกับหน่วยที่เล็กที่สุดของข้อมูล 2 มิติที่เรียกว่า พิกเซล (Pixel)



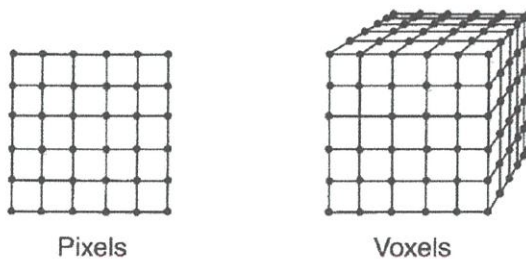
รูปที่ 2.3 ภาพตัดขวางที่ได้จากเครื่องเอกซเรย์ โทโมกราฟ (CT) หรือเครื่องตรวจวินิจฉัยโรคด้วยคลื่นแม่เหล็กไฟฟ้า (MRI)

(a) ข้อมูลเชิงปริมาตร (b) แบบจำลองสามมิติ



รูปที่ 2.4 เชิงพื้นผิวและเชิงปริมาตรแสดงในรูป

(a) ตัวอย่างภาพเชิงพื้นผิว (b) ตัวอย่างภาพเชิงปริมาตร



Pixels

Voxels

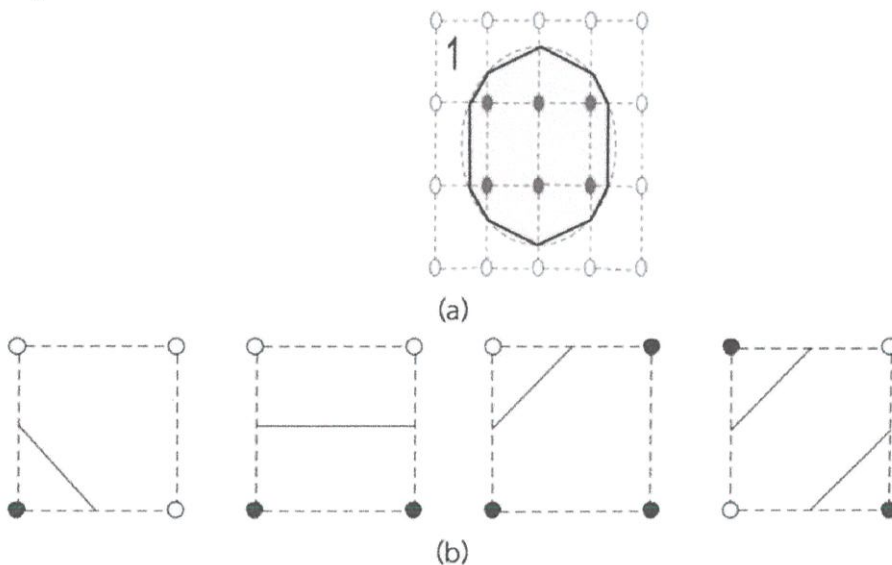
รูปที่ 2.5 หน่วยข้อมูลเล็กที่สุดของภาพและหน่วยข้อมูลเล็กที่สุดของข้อมูลเชิงปริมาตร

(a) หน่วยเล็กที่สุดของข้อมูลภาพ (b) หน่วยเล็กที่สุดของข้อมูลเชิงปริมาตร

2.4.1 การสร้างภาพ 3 มิติเชิงพื้นผิวด้วยอัลกอริทึมมาร์ชชิงคิวบ์ (Marching Cube)

ภาพที่เราเห็นในธรรมชาตินั้นจะเป็นภาพเชิงพื้นผิวเป็นส่วนใหญ่ เพราะเราสามารถมองเห็นเพียงแค່ภายนอก ในส่วนที่เป็นพื้นผิวไม่สามารถมองเห็นทะลุเข้าไปเห็นรายละเอียดภายในวัตถุนั้นๆได้ การเกิดภาพเชิงพื้นผิวในธรรมชาติ จะเกิดขึ้นได้ เมื่อแสงเดินทางไปกระทบวัตถุแล้วสะท้อนกลับออกมา การสะท้อนกลับนี้จะขึ้นอยู่กับคุณสมบัติของ พื้นผิววัตถุว่ามีการดูดกลืนแสงได้มากหรือน้อยเพียงใด และค่าของสเปกตรัมหรือค่าความถี่ของสีใดที่จะถูกสะท้อน ออกมา ทำให้เรามองเห็นวัตถุมีสีต่างๆ

มาร์ชชิงคิวบ์เป็นวิธีที่นิยมในการสร้างรูปแบบสามมิติเชิงพื้นผิวจากข้อมูลเชิงปริมาตร (Volumetric Data) โดยทำการประมาณพื้นผิวที่มีค่าความเข้มเดียวกัน หรือไอโซเซอร์เฟซ ด้วยพื้นผิวของสามเหลี่ยมเล็กๆที่ปะบนไอโซเซอร์เฟซเพื่อให้เกิดความเข้าใจในหลักการของมาร์ชชิงคิวบ์ พิจารณาอัลกอริทึมมาร์ชชิงเซลล์ ซึ่งเป็นกรณีของ 2 มิติ สมมติว่าเราต้องการประมาณรูปร่างของวัตถุที่มีค่าความเข้มเดียวในรูป 2.6 (a) ด้วยเซตของเส้นโครงในรูป 2.6 (b) พิจารณาทีละเซลล์ จากเซลล์แรกถึงเซลล์สุดท้ายเพื่อตรวจสอบค่าที่มุมทั้งสี่ของเซลล์ว่ามีมุมใดบ้างมีค่ามากกว่าค่าความเข้มของวัตถุ กำหนดค่าที่มุมนั้นเป็น “1” และแทนด้วยจุดสีขาว ส่วนมุมมีค่าน้อยกว่าค่าความเข้มของวัตถุ กำหนดค่าที่มุมนั้นเป็น “0” และแทนด้วยจุดสีดำ เมื่อพิจารณาครบสี่มุมแล้วตรวจสอบดูว่าตรงกับรูปแบบใดในรูป 2.6 (b) ทำการแทนเซลล์ที่กำลังพิจารณาด้วยรูปแบบดังกล่าว จากตัวอย่างในรูป 2.6 (a) ตำแหน่งเซลล์ที่ 1 ตรง กับรูปแบบที่ 1 ซึ่งมีรูปร่างสามเหลี่ยม



รูปที่ 2.6 ตัวอย่างข้อมูลมาร์ชชิงเซลล์

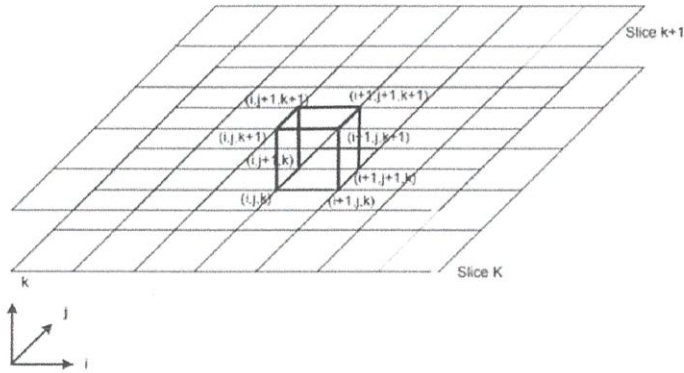
(a) เส้นโครงร่างที่ได้จากการทำมาร์ชชิงเซลล์ (เส้นทึบ)

(b) รูปแบบที่ใช้ในมาร์ชชิงเซลล์

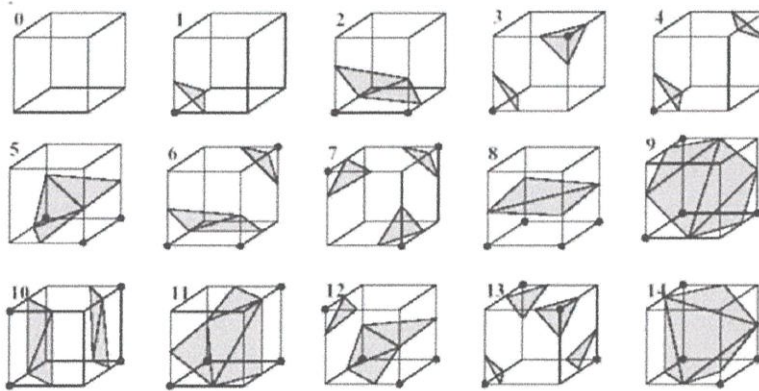
ตำแหน่งเซลล์ที่ 2 ตรงกับรูปแบบที่ 2 ซึ่งมีรูปร่างสี่เหลี่ยมขนาดของรูปแบบสี่เหลี่ยมและสามเหลี่ยมจะแตกต่างกัน ออกไปขึ้นกับตำแหน่งที่วัตถุตัดกับเซลล์ เมื่อพิจารณาครบทุกเซลล์เราจะได้เส้นโครงร่างที่ประมาณรูปร่างของวัตถุ

สำหรับกรณีของ 3 มิติมาร์ชชิงคิวบ์ทำการสแกนข้อมูล 3 มิติด้วยลูกบาศก์ (Cube) ดังแสดงใน รูปที่ 2.7 ณ แต่ละตำแหน่งของการสแกนทำการตรวจสอบดูว่าไอโซเซอร์เฟซตัดผ่านลูกบาศก์หรือไม่ ซึ่งทำได้โดยการกำหนดค่า “1” หรือจุดขาวให้กับมุม (Vertex) ของลูกบาศก์ที่มีค่าความเข้มมากกว่าค่าของไอโซเซอร์เฟซและกำหนดค่า “0”

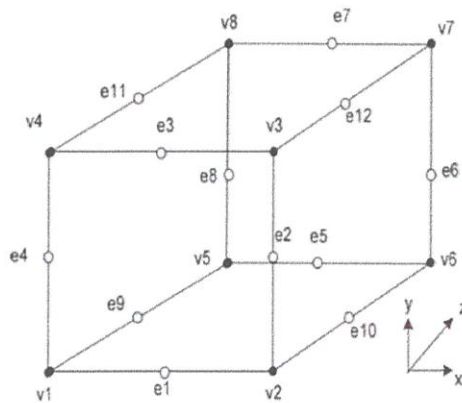
หรือจุดดำให้กับมุม(Vertex) ของลูกบาศก์ที่มีค่าความเข้มข้นน้อยกว่าค่าของไอโซเซอร์เฟซ เนื่องจากลูกบาศก์มีทั้งหมด 8 มุม ลักษณะที่ไอโซเซอร์เฟซตัดลูกบาศก์จึงมีได้ทั้งหมด $2^3=256$ รูปแบบ โดยอาศัยคุณสมบัติสมมาตรสามารถลดจำนวนรูปแบบของการตัดได้เหลือเพียง 14 รูปแบบดังแสดงในรูปที่ 2.7 กรณีที่มีอย่างน้อยหนึ่งมุมจากทั้งหมด 8 มุมที่เป็นไปตามเงื่อนไขแสดงว่าไอโซเซอร์เฟซตัดผ่านลูกบาศก์ทำการตรวจสอบรูปแบบของการตัดได้จากทั้งหมด 14 รูปแบบ และทำการประมาณไอโซเซอร์เฟซที่ตัดลูกบาศก์ด้วยเซตของพื้นผิวสามเหลี่ยมย่อยแสดงในรูป 2.9 เช่นกัน ขั้นตอนต่อไปต้องคำนวณว่าไอโซเซอร์เฟซตัดลูกบาศก์ที่ตำแหน่งใด ซึ่งมีผลให้ขนาดของสามเหลี่ยมแต่ละชิ้นแตกต่างกันออกไป



รูปที่ 2.7 มาร์ชชิ่งคิวบ์



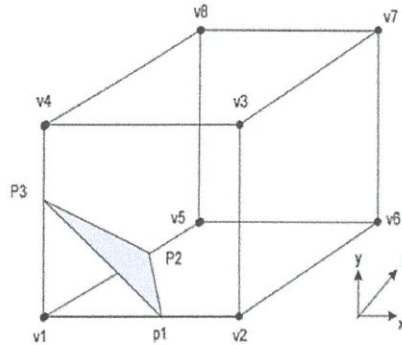
รูปที่ 2.8 รูปแบบของมาร์ชชิ่งคิวบ์



รูปที่ 2.9 การอ้างถึงลูกบาศก์

การคำนวณตำแหน่งที่ไอโซเซอร์เฟซตัดลูกบาศก์ต้องมีการอ้างอิงถึงมุม(Vertex) และด้าน(Edge) ในลูกบาศก์ ซึ่งประกอบด้วย 8 มุม ตั้งแต่ v_1 ถึง v_8 และ 12 ด้านตั้งแต่ e_1 ถึง e_{12} ดังแสดงในรูป 2.9

พิจารณารูปแบบที่ 1 ดังแสดงในรูป 2.10 จุด P1, P2 และ P3 เป็นจุดที่ไอโซเซอร์เฟซตัดลูกบาศก์ เราสามารถใช้การอินเตอร์โพลชันแบบเชิงเส้นในการหาตำแหน่งของจุดตัดดังกล่าว ตัวอย่างเช่น ตำแหน่งจุดตัด P1 สามารถคำนวณได้จาก



รูปที่ 2.10 การคำนวณตำแหน่งที่ไอโซเซอร์เฟซตัดลูกบาศก์

$$P1 = V1 + \frac{[IsoValue - f(V1)] \cdot (v2 - v1)}{f(V2) - f(V1)}$$

โดยที่

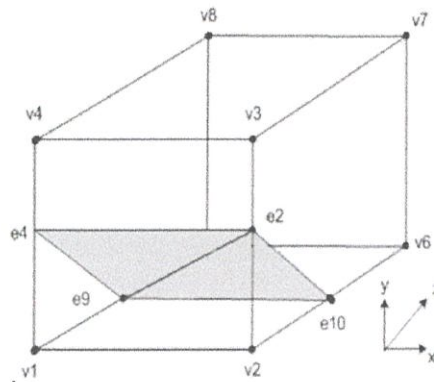
$V1$ คือจุดมุมที่ 1, $f(v1)$ คือค่าความเข้มที่ $V1$

$V2$ คือจุดมุมที่ 2, $f(v2)$ คือค่าความเข้มที่ $V2$

ไอโซแวลู(Iso-value) คือค่าความเข้มของไอโซเซอร์เฟซ

การเก็บรูปแบบของมาร์ชชิงคิวบ์ ทั้ง 14 แบบจะเก็บอยู่ในรูปของตารางที่เรียกว่าไตรเทเบิล(Tri-table) ต่อไปนี้เป็น การอธิบายการเข้าถึงตารางเพื่อให้ได้เซตของสามเหลี่ยมที่ประมาณไอโซเซอร์เฟซพิจารณาตัวอย่างของการตัดไอโซเซอร์เฟซรูปแบบที่ 2 ดังแสดงในรูป 2.11

กำหนดให้คิวบ์อินเด็กซ์(Cube-index)เป็นตัวแปรไบนารีขนาด 8 บิตเพื่อระบุว่ามีมุมใดบ้างที่มีค่าความเข้มต่ำกว่าไอโซแวลูกรณีในรูป 2.11 ได้แก่มุมที่ 1 และมุมที่ 2 ได้ว่า คิวบ์อินเด็กซ์ = 0000 0011 ซึ่งถ้าแปลงเป็นเลขฐานสิบได้เท่ากับ 5 จากนั้นนำค่าคิวบ์อินเด็กซ์เท่ากับ 5 ไปเปิดตารางไตรเทเบิลเป็นตัวแปรไบนารีขนาด 16 บิต เพื่อดูว่าเซตของสามเหลี่ยมอยู่ที่ด้านใดบ้างตามตัวอย่าง



รูปที่ 2.11 ตัวอย่างการตัดไอโซเซอเฟสรูปแบบที่ 2

ไตร-เทเบิล[5] = [10 9 2 9 4 2 -1 -1 -1 -1 -1 -1 -1 -1] ซึ่งระบุว่าประกอบด้วยสามเหลี่ยมสองชั้น ชั้นหนึ่งอยู่ระหว่างขอบ e10, e9 และ e2 ส่วนอีกชั้นอยู่ที่ e9, e4 และ e2 สามเหลี่ยมที่ได้ถูกนำมาประมาณไตรเทเบิลที่ตำแหน่งปัจจุบัน

2.4.2 แบบจำลองการให้แสง (Illumination Model)

เมื่อเราได้รับการประมาณไอโซเซอเฟสด้วยแบบจำลองที่มีลักษณะเป็นแผ่นเหมือนกระเบื้องสามเหลี่ยมติดไปทั่วทั้งไอโซเซอเฟสขั้นต่อไปคือเป็นขบวนการให้แสงกับพื้นผิวของแผ่นกระเบื้องแต่ละชั้น แบบจำลองในการให้แสงที่ใช้มากเป็นแบบจำลองของฟองก์(Phong) ซึ่งเป็นการเลียนแบบการสะท้อนของแสง(Reflection of Light) ของพื้นผิวรวมทั้งเลียนแบบผลของแสงที่มีต่อสีที่เราสังเกตเห็นได้ เมื่อเรามองไปยังพื้นผิวนั้น ตัวอย่างเช่นเมื่อภาพของลูกโบว์ลิงสีดำที่มีการฉายด้วยไฟฉายจะไม่ปรากฏให้ลูกโบว์ลิงเป็นต้นแบบจำลองของฟองก์ อธิบายผลของแสงทั้งสามชนิดได้แก่ แสงแอมเบียนท์(Ambient Light), แสงดิฟฟิวส์(Diffuse Reflection) และแสงสเปคคิวลาร์(Specular Reflection) ดังต่อไปนี้

2.4.2.1 แสงแอมเบียนท์ (Ambient Light)

แหล่งกำเนิดแสงแอมเบียนท์คือแสงที่มีความเข้มแสงเท่ากันปรากฏอยู่ทั่วไปในฉาก(Scene) ที่เราทำการสร้างแบบจำลอง ผลของแสงแอมเบียนท์ที่มีต่อพื้นผิวไม่ขึ้นกับมุมของแสงที่ตกกระทบกับพื้นผิว แสงแอมเบียนท์เป็นแสงที่ไม่มีทิศทาง เราสามารถอธิบายสีของพื้นผิวเมื่อมีแสงแอมเบียนท์ฉายไปถึงได้ดังนี้

$$C_0 = C_a K_a O_d \quad (2.1)$$

โดยที่ C_0 คือ สีที่ปรากฏบนพื้นผิวเมื่อมีการฉายแสงที่มีสี C_a ไปยังพื้นผิว K_a คือคุณสมบัติของวัสดุที่เรียกว่า สัมประสิทธิ์การสะท้อนแสงแอมเบียนท์(Ambient Reflection Coefficient) มีค่าระหว่าง 0 ถึง 1 วัตถุสีดำที่ดูดกลืนแสงมากกว่าวัตถุสีขาว ค่าสัมประสิทธิ์การสะท้อนแสงแอมเบียนท์ มีค่าเข้าใกล้ศูนย์ O_d คือ สีของพื้นผิว

2.4.2.2 การสะท้อนแสงดิฟฟิวส์ (Diffuse Reflection)

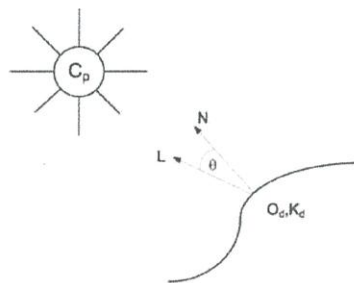
การสะท้อนแสงดิฟฟิวส์บนพื้นผิวเกิดจากการฉายแสงจากแหล่งกำเนิดแสงที่มีลักษณะเป็นจุดและมีความเข้มแสงเท่ากันในทุกทิศทาง สีของจุดบนพื้นผิวขึ้นกับระยะทางของแหล่งกำเนิดแสงจากพื้นผิว และการวางตัว (Orientation) ของพื้นผิวต่อแหล่งกำเนิดแสง บางครั้งผลของระยะทางอาจไม่นำมาคิด

ในกรณีนี้ถือว่าแหล่งกำเนิดแสงอยู่ไกลออกไปมากๆ เนื่องจากแสงดิฟฟิวส์เป็นแสงที่มีทิศทาง (Directional Light) พารามิเตอร์ที่ต้องคำนึงถึงคือ มุมระหว่างพื้นผิวและแหล่งกำเนิดแสง

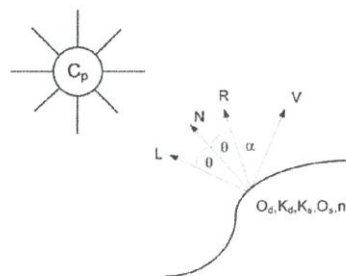
การเรียงตัวของพื้นผิวที่จุดที่ต้องการให้แสงเงาสามารถอธิบายด้วยเวกเตอร์นอร์มอล N เวกเตอร์ ที่ชี้จากพื้นผิวที่ตำแหน่งเดียวกันนี้ไปยังแหล่งกำเนิดแสงเรียกว่า L มุมระหว่าง L และ N คือมุม θ ดังแสดงในรูป 2.12 ดังนั้นแบบจำลองที่องค์การคิดการสะท้อนแสงดิฟฟิวส์คือ

$$C_0 = C_a K_a O_d + C_p K_d O_d \cos \theta \quad (2.2)$$

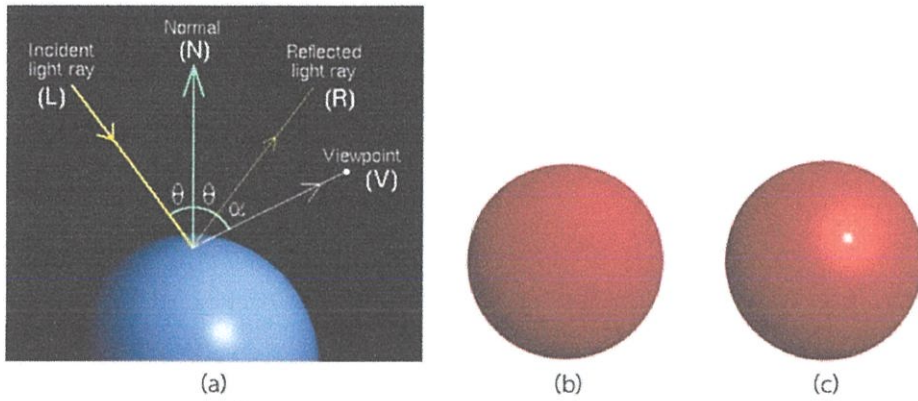
โดยที่ C_p คือ แสงของดิฟฟิวส์ที่ฉายไปยังพื้นผิว K_d คือคุณสมบัติของวัสดุที่เรียกว่าสัมประสิทธิ์การสะท้อนแสงดิฟฟิวส์ (Diffuse Reflection Coefficient) θ คือมุมระหว่างเวกเตอร์ของแหล่งกำเนิดแสง L และเวกเตอร์นอร์มอลของพื้นผิว N สังเกตว่าถ้ามุม θ มีค่า $90^\circ \cos \theta = 0$ ในกรณีนี้ไม่มีผลของแสงเงาที่เกิดจากแสงดิฟฟิวส์ในทางตรงข้ามถ้า θ มีค่า $0^\circ \cos 0 = 1$ ผลของแสงเงาที่เกิดจากแสงดิฟฟิวส์มีค่าสูงสุด



รูปที่ 2.12 การสะท้อนแสงดิฟฟิวส์



รูปที่ 2.13 การสะท้อนแสงสเป็คคิวลาร์



รูปที่ 2.14 ผลของแบบจำลองการให้แสงแบบต่างๆ

(a) การสะท้อนแสงสเป็คคิวลาร์และดิฟฟิวส์ (b) ผลของแสงดิฟฟิวส์ (c) ผลของแสงสเป็คคิวลาร์

2.4.2.3 การสะท้อนแสงสเป็คคิวลาร์ (Specular Reflection)

การสะท้อนแสงสเป็คคิวลาร์บนพื้นผิวเกิดจากการฉายแสงจากแหล่งกำเนิดแสงที่มีลักษณะเป็นจุดและมีความเข้มแสงไม่เท่ากัน การสะท้อนแสงสเป็คคิวลาร์ทำให้มีความเข้มแสงมากที่บริเวณกึ่งกลางของจุดที่แสงตกกระทบและความเข้มแสงลดลงอย่างมากเมื่อออกจากบริเวณกึ่งกลางของจุดการสะท้อนแสงสเป็คคิวลาร์ทำให้ลูกโบว์ลิงที่ฉายปรากฏลักษณะของจุดแสงการตกลงอย่างรวดเร็วของความเข้มแสงถูกควบคุมด้วยค่าเอ็กซ์โพเนนซ์ของการสะท้อนแสง n ในกรณีของวัสดุที่มีคุณสมบัติสะท้อนแสงได้ดีเช่น กระจกเงาค่า n จะมีค่าอนันต์ โดยปกติการสะท้อนแสงสเป็คคิวลาร์ มักไม่ขึ้นกับสีของพื้นผิวและแสงที่รับรู้ได้ทางตาปรากฏเป็นสีขาว รวมทั้งขึ้นกับทิศทางของการมองแบบจำลองฟ็องก์ที่คิดการสะท้อนแสงทุกชนิด(รูป 2.13) สามารถเขียนได้เป็น

$$C_0 = C_a K_a O_d + C_p K_d O_d \cos\theta + C_p K_s O_s (\cos\alpha)^n \tag{2.3}$$

โดยที่ O_s คือแสงของสเป็คคิวลาร์ K_s คือคุณสมบัติของวัสดุที่เรียกว่าสัมประสิทธิ์การสะท้อนแสงสเป็คคิวลาร์ (Specular Reflection Coefficient) α คือมุมระหว่างเวกเตอร์แสงสะท้อน R และเวกเตอร์ที่ชี้ไปยังมุมมองของผู้สังเกต V ดังแสดงในรูป 2.16 ความแตกต่างระหว่างผลที่เกิดจากการสะท้อนแสงดิฟฟิวส์และสเป็คคิวลาร์ แสดงในรูป 2.14

2.4.3 การประมาณเวกเตอร์นอร์มอล (Normal Vector Approximation)

การให้แสงเงาโดยคำนึงถึงการสะท้อนของแสง(Reflection of Light) ของพื้นผิวที่เกิดจากแสงทั้งสาม ชนิด ได้แก่แสงแอมเบียนท์, แสงดิฟฟิวส์ และแสงสเป็คคิวลาร์ ที่อธิบายในหัวข้อที่แล้วจำเป็นต้องมีการคำนวณค่าเวกเตอร์นอร์มอลที่มุมของลูกบาศก์โดยหลักที่ว่าพื้นผิวที่มีค่าความเข้มคงที่มีค่าองค์ประกอบของแอมเบียนท์ในแนวเส้นสัมผัส (Tangential Direction) เป็นศูนย์ ดังนั้นทิศทางของเวกเตอร์แอมเบียนท์จะตั้งฉากกับพื้นผิว ดังนั้นเวกเตอร์นอร์มอลที่เราสนใจสามารถหาได้จากเวกเตอร์แอมเบียนท์ซึ่งถูกนิยามว่าเป็นอนุพันธ์ของฟังก์ชันความเข้ม

$$g(x,y,z) = \nabla f(x,y,z) \tag{2.4}$$

ซึ่งสามารถค่าได้จากสมการผลต่างต่อไปนี้

$$G_x(i, j, k) = \frac{f(i+1, j, k) - f(i-1, j, k)}{\Delta x} \quad (2.5)$$

$$G_y(i, j, k) = \frac{f(i, j+1, k) - f(i, j-1, k)}{\Delta y} \quad (2.6)$$

$$G_z(i, j, k) = \frac{f(i, j, k+1) - f(i, j, k-1)}{\Delta z} \quad (2.7)$$

โดยที่ $f(i, j, k)$ คือ ค่าความเข้มที่พิกเซล (i, j) ในสไลด์ k และ $\Delta x, \Delta y, \Delta z$ คือ ความยาวของลูกบาศก์หลังจากคำนวณเวกเตอร์นอร์มอลที่มุมของลูกบาศก์ได้แล้ว จึงทำการอินเตอร์โพลเลขชั้นเวกเตอร์นอร์มอลลงไปที่ยุ่ของพื้นผิวสามเหลี่ยมเพื่อทำการแรเงาต่อไป

2.4.4 การแรเงา (Shading Model)

แบบจำลองการแรเงาบริเวณภายในของสามเหลี่ยมที่ใช้นามมี 2 แบบคือแบบจำลองของกูร์ราด(Gouraud) และ แบบจำลองของฟ็องก์

แบบจำลองการแรเงาของกูร์ราด ดังแสดงในรูป 2.15 (a) เริ่มจากการคำนวณเวกเตอร์นอร์มอลที่มุมของสามเหลี่ยม จากนั้นทำการคำนวณค่าสีที่มุมของสามเหลี่ยมโดยใช้แบบจำลองการให้แสงในหัวข้อที่ 2.4.2 ค่าสีภายในพื้นผิวสามเหลี่ยมได้ จากการอินเตอร์โพลเลขชั้นแบบเชิงเส้นจากค่าสีที่มุม ค่าสีที่จุด J ได้จากการอินเตอร์โพลเลขชั้นแบบเชิงเส้นระหว่าง RGB1 และ RGB2 ซึ่งเป็นค่าสีที่มุมของสามเหลี่ยม ส่วนค่าสีที่จุด K ได้จากการอินเตอร์โพลเลขชั้นแบบเชิงเส้นระหว่าง RGB1 และ RGB3 ขั้นตอนต่อไปจึงทำการอินเตอร์โพลเลขชั้นเชิงเส้นให้กับพิกเซลระหว่างเส้น JK

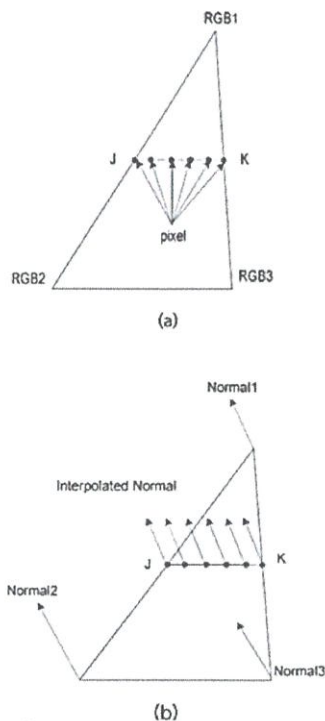
แบบจำลองการแรเงาของฟ็องก์ ดังแสดงในรูป 2.15 (b) เริ่มจากการคำนวณเวกเตอร์นอร์มอลที่มุมของสามเหลี่ยม จากนั้นทำอินเตอร์โพลเลขชั้นเวกเตอร์นอร์มอลให้กับทุกจุดภายในสามเหลี่ยม ค่านอร์มอลเวกเตอร์ที่จุด J ได้จากการอินเตอร์โพลเลขชั้นแบบเชิงเส้นระหว่าง Normal1 และ Normal2 ซึ่งเป็นค่าเวกเตอร์นอร์มอลเวกเตอร์ที่จุดมุมของสามเหลี่ยมค่าเวกเตอร์นอร์มอลเวกเตอร์ที่จุด K ได้จากการอินเตอร์โพลเลขชั้นแบบเชิงเส้นระหว่าง Normal1 และ Normal3 ขั้นตอนต่อไปจึงทำการอินเตอร์โพลเลขชั้นเชิงเส้นให้กับพิกเซลระหว่างเส้น ค่านอร์มอลเวกเตอร์ในแต่ละจุดจะถูกนำไปคำนวณค่าสีตามแบบจำลองการให้แสงในหัวข้อที่ 2.4.2

2.4.5 การสร้างภาพ 3 มิติเชิงปริมาตร

ภาพ 3 มิติเชิงปริมาตรเป็นภาพที่สามารถมองเห็นรายละเอียดได้มากกว่ารายละเอียดที่ปรากฏอยู่บนพื้นผิวของวัตถุ เพราะสามารถมองเห็นทะลุเข้าไปเห็นรายละเอียดถึงภายในเนื้อของวัตถุได้ในธรรมชาติ นั้นภาพเชิงปริมาตรจะเกิดขึ้นในวัตถุโปร่งแสงหรือวัตถุโปร่งใสต่างๆ ซึ่งวัตถุประเภทนี้จะมีคุณสมบัติยอมให้แสงเดินทางผ่านเข้าไปในเนื้อของวัตถุหรือทะลุออกไปในอีกทิศทางหนึ่งได้ ทำให้ภาพที่ปรากฏแก่ตาเป็นภาพที่มีความลึกของวัตถุมากกว่าการสะท้อนของแสงและเงาที่ผิวของวัตถุเพียงอย่างเดียวตัวอย่างที่สามารถเห็นได้ทั่วไป เช่น เราสามารถมองเห็นภายในเนื้อเยื่อ

บางส่วนของแมงกะพรุน หรือปลาบางชนิดที่มีความโปร่งแสงได้ เป็นต้น ซึ่งการที่จะสามารถมองเห็นรายละเอียดได้มากหรือน้อยจะขึ้นอยู่กับสี และความทึบแสง ซึ่งเป็นคุณสมบัติเฉพาะตัวของแต่ละวัตถุว่าจะยอมให้ แสงผ่านเข้าไปได้มากน้อยแค่ไหน และสะท้อนแสงออกมาในความถี่ของสีใด ขั้นตอนการสร้างภาพ 3 มิติเชิงปริมาตรสามารถแบ่งได้เป็น 4 ขั้นตอนดังนี้

1. การโปรเจ็คชันทางเดินแสง(Projection)
2. การแบ่งส่วนภาพ(Segmentation)
3. การหาค่าทึบแสงและค่าสี(Opacity and Color Mapping)
4. การประกอบภาพ(Composition)



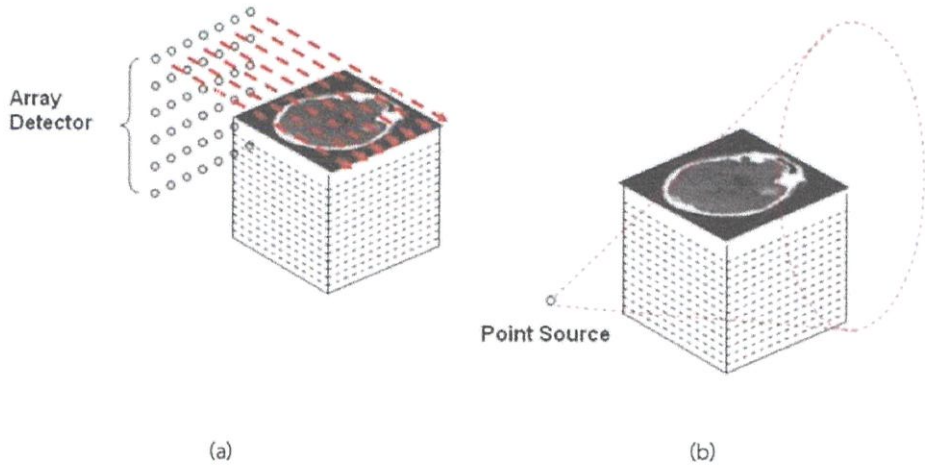
รูปที่ 2.15 แบบจำลองการแรเงา

(a) แบบจำลองของกัวร์ราต (b) แบบจำลองของฟ็องก์

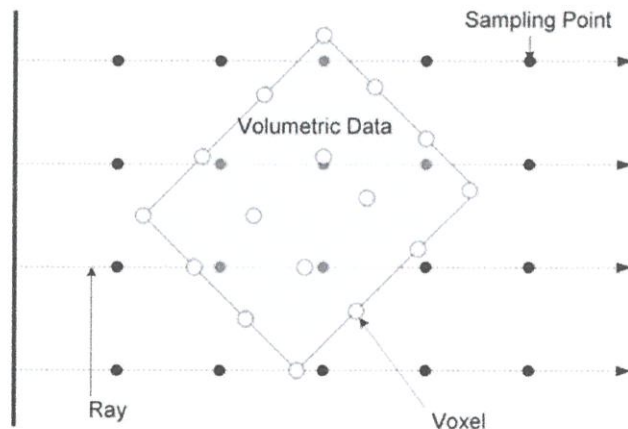
2.4.5.1 การโปรเจ็คชันทางเดินแสง (Projection)

การโปรเจ็คชันทางเดินแสงเป็นการคำนวณทางเดินของแสงที่ผ่านเข้าไปในข้อมูลเชิงปริมาตร ทางเดินแสงที่ใช้แบ่งออกเป็น 2 ประเภทได้แก่ ทางเดินแสงแบบขนานและทางเดินแสงแบบกรวย ดังแสดงในรูป 2.16 ทั้งแบบลำแสงขนานและแบบลำแสงกรวยเมื่อได้ทางเดินของแสงแล้ว ทำการกำหนดจุดสุ่มค่าความเข้มจากข้อมูลเชิงปริมาตรด้วยระยะทางที่เท่ากันตลอดทางเดินของแสง ดังแสดงในรูป 2.17 การคำนวณค่าให้จุดสุ่มต้องใช้เทคนิคการอินเตอร์โพลชันจากแปดจุดรอบด้าน เนื่องจากการอินเตอร์โพลชันใน 3 มิติ เทคนิคการอินเตอร์โพลชันสามารถแบ่งได้ เป็น 1. แบบใช้ค่าใกล้เคียงที่สุด 2. แบบเชิงเส้นและ 3. แบบคิวบิกคอนโวลูชัน แบบที่นิยมใช้มากที่สุดคือ แบบเชิงเส้น ซึ่งต้องใช้การอินเตอร์โพลชันใน 1 มิติแบบเชิงเส้นทั้งหมด 7 ครั้ง ดังแสดงในรูปที่ 2.18 นอกจากนั้น

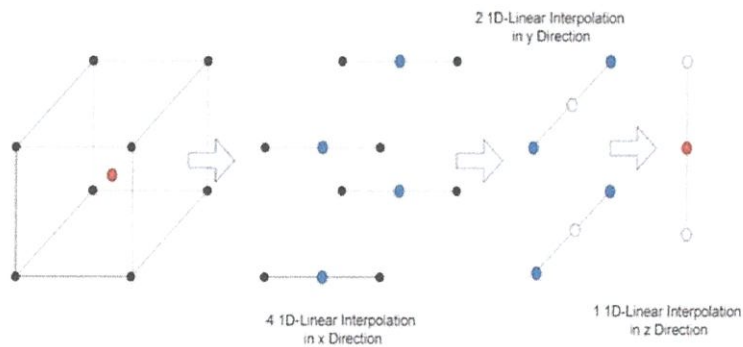
ก่อนทำการ โปรเจ็คชันทางเดินของแสงข้อมูลเชิงปริมาตรอาจถูกหมุนหรือซูม เพื่อเปลี่ยนมุมมองซึ่งเป็นสิ่งที่จำเป็นในการสร้างแบบจำลอง 3 มิติ



รูปที่ 2.16 การโปรเจคชันทางเดินแสง
(a) ลำแสงขนาน (b) ลำแสงกรวย



รูปที่ 2.17 การโปรเจคชันทางเดินแสงแบบขนานและการสุ่มข้อมูลความเข้ม(มองในระนาบเดียว)



รูปที่ 2.18 การอินเตอร์โพลเลชันแบบเชิงเส้นในสามมิติ

2.4.5.2 การแบ่งส่วนภาพ (Segmentation)

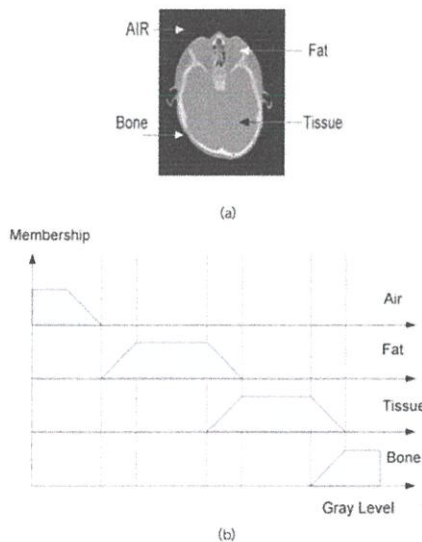
การสร้างแบบจำลอง 3 มิติจำเป็นต้องมีการแบ่งส่วนข้อมูลวัตถุประสงค์ของการแบ่งส่วนภาพ คือทำให้สามารถกำหนดค่าทึบแสง(Opacity) และค่าสีสำหรับแต่ละส่วนของกลุ่มข้อมูล ตัวอย่างการสร้างภาพ 3 มิติเชิงปริมาตรของร่างกายมนุษย์ เราทำการแบ่งกลุ่มข้อมูลออกเป็น 4 กลุ่ม ดังแสดงในรูป 2.19 ได้แก่ 1. อากาศ 2. ไขมัน 3. เนื้อเยื่อ และ 4. กระดูก จากนั้นกำหนดให้ค่าทึบแสงของกระดูกมีค่ามากที่สุดเท่ากับ 0.9 ในขณะที่กำหนดให้ค่าทึบแสงของอากาศมีค่าน้อยที่สุดหรือเท่ากับ 0 ในขณะที่ค่าทึบแสงของไขมันและเนื้อเยื่อมีค่าประมาณ 0.3 และ 0.5 ตามลำดับ ลักษณะนี้ทำให้เรามองผ่านทะลุชั้น อากาศ ไขมัน เนื้อเยื่อ จนถึงชั้นกระดูก ในกรณีที่เรต้องการให้การสร้างภาพ 3 มิติเชิงปริมาตรของร่างกายมนุษย์ให้ภาพผลลัพธ์ที่มีลักษณะคล้ายภาพ 3 มิติเชิงพื้นผิว เราสามารถทำได้โดยกำหนดให้กลุ่มข้อมูลที่ต้องการสร้างภาพเชิงพื้นผิวมีค่าทึบแสงเท่ากับ 1 ในขณะที่กลุ่มอื่นๆ มีค่าทึบแสงเท่ากับ 0 วัตถุประสงค์ อีกประการของการแบ่งส่วนภาพคือ เราสามารถกำหนดค่าสีให้กับกลุ่มของข้อมูลทำให้ภาพ 3 มิติที่แสดงมีค่าโทนสีต่างๆตามที่ต้องการ

เทคนิคที่นิยมใช้ในการแบ่งกลุ่มข้อมูลคือ วิธีการทำเทรซโซติงค์ โดยทำการเลือกสองระดับกัน T1 และ T2 ที่ซึ่งจุด $f(x, y)$ เป็นของข้อมูลกลุ่มหนึ่งถ้า $T1 < f(x,y) < T2$ ข้อจำกัดของวิธีนี้คือมีบางข้อมูลที่ไม่สามารถกำหนดได้ ว่า เป็นของกลุ่มใด (ย่านระหว่างเส้นประในรูปที่ 2.21) ทางแก้ไขคือนอกจากจะใช้ค่าความเข้มแสง $f(x, y)$ ยังมีการใช้ ค่าแกร์เดียน สมการ 2.4-2.6ช่วยในการแบ่งกลุ่มข้อมูล ซึ่งทำให้การแบ่งกลุ่มข้อมูลมีความถูกต้องมากขึ้น

2.4.5.3 การกำหนดค่าทึบแสงและค่าสี (Opacity and Color Mapping)

การกำหนดค่าทึบแสงให้กับกลุ่มข้อมูลทำได้โดยใช้ฟังก์ชันถ่ายโอน โดยตัวแปรของฟังก์ชัน คือค่าความเข้มและขนาดของแกร์เดียน หรืออาจเป็นฟังก์ชันอื่นเข้ามาร่วมด้วยก็ได้ การกำหนดค่าทึบแสง โดยใช้ฟังก์ชันถ่ายโอนสามารถเขียนอยู่ในรูป

$$\alpha_i = O(I_i, \nabla, \dots) \tag{2.8}$$



รูปที่ 2.19 การแบ่งส่วนภาพ (Segmentation)

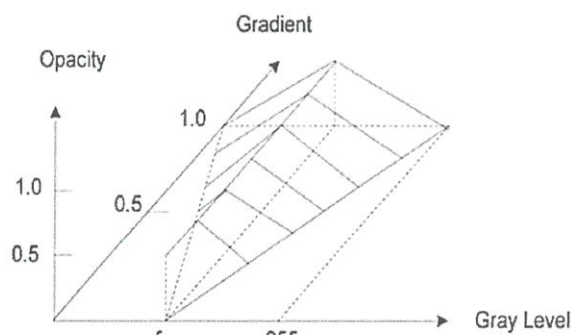
(a) ส่วนต่างๆจากภาพตัดขวางจาก CT (b) การแบ่งส่วนโดยใช้ค่าแบ่งส่วนเทา

ในกรณีของภาพถ่ายจากเครื่องซีทีสแกนเนอร์ ค่าความเข้มแสงของจุดภาพในแต่ละอวัยวะจะแตกต่างกันอย่างชัดเจน ดังนั้นการใช้ค่าความเข้มแสงแต่เพียงอย่างเดียวก็เพียงพอในการแบ่งส่วนของภาพอวัยวะต่างๆ ออกจากกันแต่ในกรณีของภาพถ่ายจากเครื่องเครื่องตรวจวินิจฉัยโรคด้วยคลื่นแม่เหล็กไฟฟ้า อวัยวะที่ต่างกันอาจมีค่าความเข้มแสงเท่ากันได้ การใช้ค่าความเข้มแสงเพียงอย่างเดียวในการแบ่งส่วนของภาพอวัยวะอาจไม่เพียงพอ ดังนั้นจึงต้องมีการพิจารณาค่าอื่นๆร่วมด้วย เลวอย(Levoy) ได้เสนอฟังก์ชันโอนถ่าย ซึ่งสามารถใช้ในการสร้างแบบจำลอง 3 มิติเชิงปริมาตรได้ อดีโดยใช้ค่าความเข้มแสงและขนาดของแกร์เดียน การใช้แกร์เดียนในการแบ่งส่วนของภาพเนื่องจากว่าแกร์เดียนบอกถึงการเปลี่ยนแปลงของค่าความเข้มแสง ในบริเวณรอยต่อระหว่าง เนื้อเยื่อ 2 ชนิด (ย่านระหว่างเส้นประในรูปที่ 2.20) จะมีความเร็วในการเปลี่ยนแปลงของความเข้มแสงอย่างคงที่ ฟังก์ชันโอนถ่ายของเลวอย มีลักษณะดังสมการต่อไปนี้

$$\alpha_i(r, f_v) = 1 - \frac{1}{r|\nabla_i|} |f_v - l_i| \quad \text{ถ้า} \quad |\nabla_i| > 0 \quad \text{และ} \quad l_i - r|\nabla_i| \leq f_v \leq l_i + r|\nabla_i|$$

$$\alpha_i(r, f_v) = 1 \quad \text{ถ้า} \quad |\nabla_i| = 0 \quad \alpha_i(r, f_v) = 0 \quad \text{ค่าอื่นๆ}$$

ค่าความเข้มแสงของจุดภาพที่ต้องการให้ปรากฏคือ f_v ซึ่งมีค่าที่บ่งแสงมากที่สุด ส่วนค่าความสว่างที่ไม่ใช่ f_v และที่ขนาดของแกร์เดียนค่าต่างๆ จะมีค่าที่บ่งแสงลดลงโดยฟังก์ชันถ่ายโอนของเลวอย แสดงในรูป 2.20 การกำหนดค่าสีให้กับกลุ่มของข้อมูลสามารถทำให้การแบ่งส่วนภาพทำได้ง่ายขึ้น โดยกลุ่มข้อมูลที่อยู่ในกลุ่มหนึ่งจะถูกกำหนดให้มีสีต่างๆ ซึ่งคล้ายกับการทำสีเทียม(Pseudo Coloring) ฟังก์ชันโอนถ่ายของการกำหนดค่าสีจะถูกทำในแต่ละค่าแดง เขียว น้ำเงิน ดังนี้



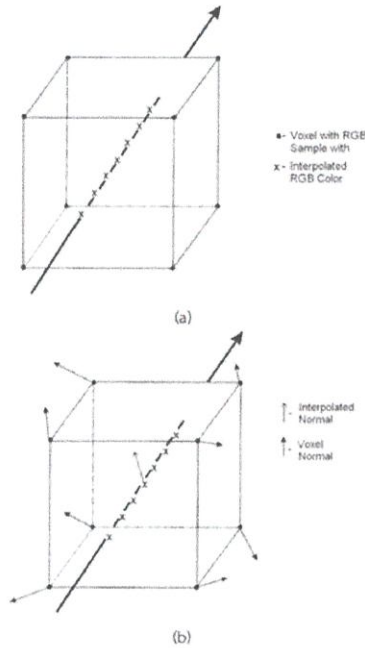
รูปที่ 2.20 การกำหนดค่าที่บ่งแสง

$$R_i = T_r(l_i, |\nabla_i|, \dots)$$

$$G_i = T_g(l_i, |\nabla_i|, \dots)$$

$$B_i = T_b(l_i, |\nabla_i|, \dots)$$

หลังจากขั้นตอนการกำหนดค่าทึบแสงและค่าสีทุกหรือกเซลในข้อมูลเชิงปริมาตรจะมีค่าทึบแสงและค่าสี ค่าแดง เขียว น้ำเงินขึ้นกับว่าว็อกเซลนั้นอยู่ที่กลุ่มไหน ต่อจากนั้นเมื่อมีการกำหนดเส้นแนวลำแสงที่ผ่านเข้าไปในข้อมูลเชิงปริมาตรค่าสีที่มุมทั้งแปดรวมทั้งค่าเวกเตอร์นอร์มอลที่คำนวณจากสมการ จะถูกอินเตอร์โพลให้กับจุดสุ่มตามแนวลำแสงโดยคำนึงถึงการให้แสงเงาในหัวข้อ 2.4.2 การอินเตอร์โพลเส้นแบบ 3 มิติสามารถทำได้ 2 วิธี เช่นเดียวกับการอินเตอร์โพลเส้นในระนาบ 2 มิติ กล่าวคือวิธีของฟ็องก์และวิธีของกูร์ราต ดังแสดงในรูปที่ 2.21



รูปที่ 2.21 การอินเตอร์โพลเส้นสามมิติของค่าสีตามแนวลำแสง สามารถทำได้สองวิธีคือ (a) วิธีของกูร์ราต (b) วิธีของฟ็องก์

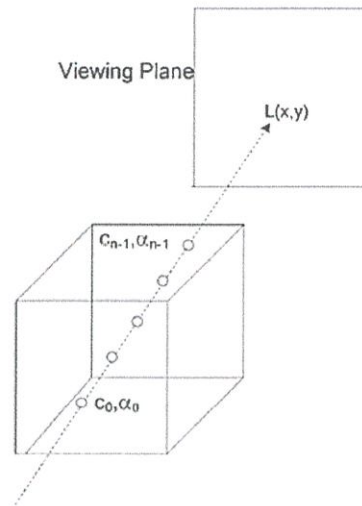
2.4.5.4 การประกอบภาพ (Composition)

การประกอบภาพเป็นขั้นตอนสุดท้ายของการสร้างภาพ 3 มิติเชิงปริมาตร โดยจะทำการรวมคุณสมบัติ ของจุดแสงที่คำนวณได้ อันได้แก่ค่าทึบแสงและค่าสี มาสร้างเป็นจุดภาพบนระนาบของการมอง โดยที่ภาพบนระนาบของการมองจะเกิดจากผลของแสงที่ส่องลงไปข้อมูลเชิงปริมาตรรวมกันตลอดหนึ่งลำแสง โดยสามารถเขียนเป็นสมการทางคณิตศาสตร์ได้ ดังนี้

$$L(\vec{x}) = \sum_{i=0}^{n-1} c_i \cdot \prod_{j=0}^{i-1} (1 - \alpha_j)$$

$$= c_0 + c_1(1 - \alpha_0) + c_2(1 - \alpha_0)(1 - \alpha_1) + \dots + c_{n-1}(1 - \alpha_0) \dots (1 - \alpha_{n-2})$$

โดยที่ n คือ จำนวนของจุดที่สุ่มตามแนวลำแสงในข้อมูลเชิงปริมาตร c_n คือ ค่าสีที่จุดสุ่ม nth α_n คือ ค่าทึบแสงของจุดสุ่มที่ nth และ $L(x)$ คือค่าสีบนระนาบของการมองที่จุด x ดังแสดงในรูป 2.22



รูปที่ 2.22 การประกอบภาพ

2.5 การประมวลผลข้อมูลในระบบเครือข่าย

2.5.1 ชนิดของสัญญาณอิเล็กทรอนิกส์

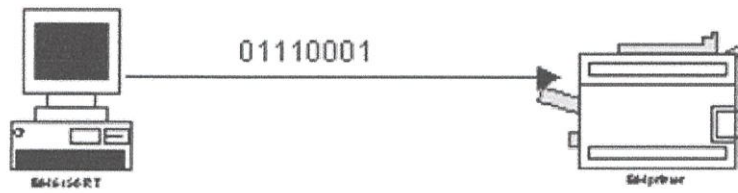
สัญญาณอิเล็กทรอนิกส์ที่ใช้ในการสื่อสารสามารถแบ่งได้เป็น 2 ประเภทคือ สัญญาณแอนะล็อก และสัญญาณดิจิทัล สัญญาณแอนะล็อกได้แก่สัญญาณเสียง และสัญญาณในธรรมชาติทั้งหมด ปัญหาที่สำคัญของสัญญาณ แอนะล็อกก็คือเรื่องสัญญาณรบกวน ซึ่งในบางครั้งอาจทำให้ระบบไม่สามารถใช้งานได้เลย ดังนั้นจึงมีการนำสัญญาณดิจิทัลเข้ามาแทนที่

1. สัญญาณแบบแอนะล็อก(Analog Signal) จะเป็นสัญญาณแบบต่อเนื่องที่ทุกๆ ค่า ที่เปลี่ยนแปลงไปของระดับสัญญาณจะมีความหมาย การส่งสัญญาณแบบนี้จะถูกรบกวนให้มีการแปลความหมายผิดพลาดได้ง่าย เนื่องจากค่าทุกค่าถูกนำมาใช้งาน ซึ่งสัญญาณแบบแอนะล็อกนี้จะเป็นสัญญาณที่สื่อกลางในการสื่อสารส่วนมากใช้อยู่เช่นสัญญาณเสียงในสายโทรศัพท์ เป็นต้น

2. สัญญาณแบบดิจิทัล(Digital Signal) จะประกอบขึ้นจากระดับสัญญาณเพียง 2 ค่าคือ สัญญาณระดับสูงสุด และสัญญาณระดับต่ำสุด ดังนั้นจะมีประสิทธิภาพ และความน่าเชื่อถือสูงกว่าแบบอนาลอกเนื่องจากการใช้งานค่าสองค่า เพื่อนำมาตีความหมายเป็นปิด/เปิด หรือ 0/1 เท่านั้น ซึ่งเป็นสัญญาณที่คอมพิวเตอร์ใช้ในการติดต่อสื่อสารกัน

2.5.2 รหัสที่ใช้แทนข้อมูลในการสื่อสาร

การติดต่อสื่อสารแบบใช้สัญญาณที่เป็นดิจิทัลนั้น ส่วนใหญ่เป็นการติดต่อระหว่างอุปกรณ์อิเล็กทรอนิกส์ เพื่อใช้ถ่ายทอดข้อมูลบางอย่างซึ่งกันและกัน ในระบบการสื่อสารข้อมูลนั้นจะต้องเกิดการส่งข้อมูลไม่ว่าจะเป็นตัวอักษร ตัวเลข หรือสัญลักษณ์พิเศษ จากจุดหนึ่งไปยังอีกจุดหนึ่ง แต่ข่าวสารแบบนี้จะถูกเปลี่ยนจากรูปแบบที่มนุษย์เข้าใจให้กลายเป็นข่าวสารที่เครื่องคอมพิวเตอร์จัดเก็บไว้ได้ รหัสที่ใช้ในการสื่อสารทั้งหมดจะอยู่ในรูปแบบของระบบไบนารีหรือ เลขฐานสอง ซึ่งรหัสมาตรฐานที่ใช้ในการสื่อสารข้อมูลเช่น รหัสแอสกี เป็นต้น ความเร็วในการส่งข้อมูลจะมีหน่วยเป็นบิตต่อวินาที(Bit per Second หรือ bps)



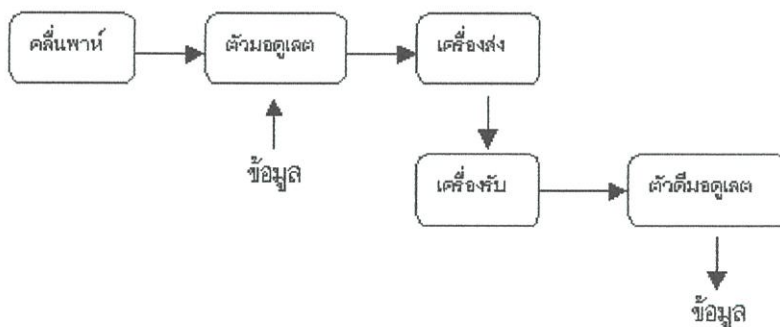
รูปที่ 2.23 ตัวอย่างการส่งรหัสแอสกีจากคอมพิวเตอร์ไปยังเครื่องพิมพ์

2.5.3 ระบบสื่อสารแอนะล็อก

การสื่อสารแอนะล็อกเป็นระบบที่ออกแบบให้ส่งข้อมูลสัญญาณแอนะล็อก เช่น สัญญาณเสียง แต่ได้มีการพัฒนาจนประยุกต์ให้สามารถส่งข่าวสารได้ด้วยในปัจจุบัน ปัญหาสำคัญสำหรับการสื่อสารแอนะล็อกก็คือเรื่องสัญญาณรบกวน แต่เนื่องจากสัญญาณในธรรมชาติทั้งหมดเป็นสัญญาณแอนะล็อก จึงยังคงเห็นการพัฒนาของการสื่อสารแบบแอนะล็อกในปัจจุบัน เช่น การมอดูเลตแอมพลิจูด (Amplitude Modulation หรือ AM) การมอดูเลตความถี่ (Frequency Modulation หรือ FM)

2.5.4 การมอดูเลต

การมอดูเลต (Modulation) เป็นการผสมสัญญาณของข้อมูลเข้ากับสัญญาณอีกสัญญาณหนึ่งเรียกว่า คลื่นพาห้ (carrier) ซึ่งสัญญาณนี้มีความถี่ที่เหมาะสมกับช่องสัญญาณนั้นๆ เพื่อให้ข้อมูลที่ส่งเข้าไปในช่องสัญญาณเดินทางได้ไกลมากขึ้น การเลือกวิธีมอดูเลตขึ้นอยู่กับปัจจัยหลายประการ เช่น ชนิดของสัญญาณ แบนด์วิดท์ ประสิทธิภาพของระบบที่ต้องการ และความต้านทานต่อสัญญาณรบกวน เป็นต้น



รูปที่ 2.24 รูปแบบของการสื่อสารในการรับส่งสัญญาณ

จากรูปที่ 2.24 แสดงรูปแบบของการสื่อสารในการรับส่งสัญญาณอย่างง่าย โดยคลื่นพาห้ผสมสัญญาณข้อมูลที่ตัวมอดูเลต (Modulator) แล้วส่งไปที่เครื่องส่ง จากเครื่องส่งไปยังเครื่องรับจะเป็นช่องสัญญาณสำหรับลำเลียงสัญญาณผสมนี้ สัญญาณผสมจากเครื่องรับจะไปเข้าตัวดีมอดูเลต (Demodulate) เพื่อแยกสัญญาณข้อมูลออกมา โดยการมอดูเลตที่ใช้อยู่ในปัจจุบัน 3 วิธีได้แก่

1. การมอดูเลตแอมพลิจูด (Amplitude Modulation หรือ AM) วิธีนี้แอมพลิจูดของคลื่นพาห้จะเปลี่ยนแปลงตามสัญญาณของข้อมูลที่เข้ามา การมอดูเลตแบบแอมพลิจูด เป็นวิธีที่ง่ายที่สุดในการมอดูเลต แต่คุณภาพของสัญญาณไม่ดี มีความต้านทานสัญญาณรบกวนต่ำ เหมาะกับข้อมูลที่ไม่ต้องการคุณภาพมากนัก เช่น สัญญาณเสียง เป็นต้น

2. การมอดูเลตความถี่(Frequency Modulation หรือ FM) วิธีการนี้เป็นการเปลี่ยนแปลงความถี่ของคลื่นพาห์ตามสัญญาณของข้อมูลที่เข้ามา การมอดูเลตแบบความถี่ ให้คุณภาพที่ดีกว่าการมอดูเลตแบบแอมพลิจูด แต่ระบบจะซับซ้อนกว่า
3. การมอดูเลตเฟส(Phase Modulation หรือ PM) เป็นการมอดูเลตที่ใช้ในการเปลี่ยนแปลงเฟสของคลื่นพาห์ตามสัญญาณข้อมูลที่เข้ามา ทั้งคุณภาพของสัญญาณและความซับซ้อนไม่ค่อยแตกต่างจากการมอดูเลตแบบความถี่เท่าใดนัก ข้อแตกต่างระหว่างการมอดูเลตแบบความถี่ กับการมอดูเลตแบบเฟส คือ การมอดูเลตแบบเฟสใช้คลื่นพาห์เพียงความถี่เดียว การมอดูเลตและดีมอดูเลตสามารถทำได้ประหยัดกว่า แต่ไม่ได้หมายความว่าซับซ้อนน้อยกว่า

2.5.5 ระบบสื่อสารดิจิทัล

ในยุคที่เริ่มมีการรับส่งข้อมูล ระบบต่างๆ ทำงานแบบแอนะล็อกทั้งหมด ต่อมาเมื่อเทคโนโลยีทางด้านดิจิทัลได้ก้าวหน้าขึ้นมา จึงได้เริ่มมีการนำเทคโนโลยีดิจิทัลเข้าไปทดแทนแบบแอนะล็อกเดิม ทั้งการนำไปทดแทนทั้งหมด เช่น สร้างโครงข่ายชนิดใหม่ หรือนำไปทดแทนบางส่วน เช่น โมเด็ม ทั้งนี้เนื่องจากข้อดีของสัญญาณแบบดิจิทัลนั่นเอง เช่น

1. ให้คุณภาพการรับส่งข้อมูลที่เท่ากันหรือดีกว่าแอนะล็อก
2. ง่ายต่อการบำรุงรักษา
3. เพิ่มเติม ปรับปรุง หรือเปลี่ยนแปลงความสามารถหรือบริการของระบบได้ง่าย
4. มีความเร็วในการรับส่งข้อมูลสูง ทนต่อสัญญาณรบกวนได้ดี

2.5.6 รูปแบบการมอดูเลตสัญญาณดิจิทัล

การมอดูเลตสัญญาณดิจิทัล(Digital Modulation) เข้ากับคลื่นพาห์ที่เป็นสัญญาณไซน์นั้นมียุทธวิธีหลายรูปแบบ ทั้งนี้เพื่อต้องการให้สัญญาณดิจิทัลเหล่านั้นสามารถส่งผ่านตัวกลางที่ออกแบบมาสำหรับสัญญาณแบบแอนะล็อกได้ เช่น โครงข่ายโทรศัพท์พื้นฐาน โมเด็ม เป็นต้น การมอดูเลตที่ใช้กันทั่วไปได้แก่

1. การมอดูเลตเชิงเลขทางแอมพลิจูด(Amplitude shift keying หรือ ASK)
2. การมอดูเลตเชิงเลขทางความถี่(Frequency shift keying หรือ FSK)
3. การมอดูเลตเชิงเลขทางเฟส(Phase shift keying หรือ PSK)
4. การมอดูเลตแบบควอดเรเจอร์แคเรียร์แอมพลิจูด(Quadrature carrier Amplitude Modulation หรือ

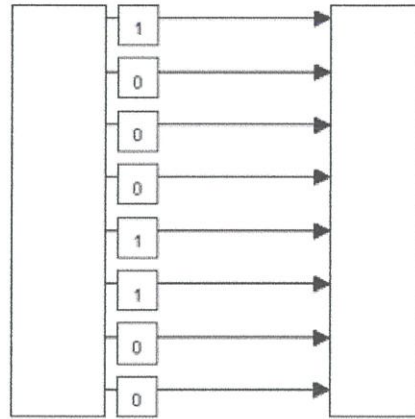
QAM)

2.5.7 รูปแบบของการส่งข้อมูล

รูปแบบของสายส่งสัญญาณสื่อสารอาจประกอบด้วยสายส่งตั้งแต่หนึ่งสายขึ้นไป ซึ่งทำให้เกิดช่องทางการส่งข้อมูลได้มากกว่าหนึ่งช่องทาง รูปแบบของการส่งผ่านข้อมูลสามารถแบ่งได้เป็น 2 รูปแบบคือ

2.5.7.1 การส่งผ่านข้อมูลแบบขนาน(Parallel Transmission)

ในรูปแบบการส่งผ่านข้อมูลในลักษณะนี้ทุกบิตที่แทนข้อมูลหนึ่งตัวอักษร จะถูกส่งผ่านไปตามสายส่ง หลายๆ สายขนานกันไป ดังนั้นทุกบิตจะเดินทางถึงผู้รับพร้อมๆ กัน และจำนวนสายส่งเพื่อให้เกิดช่องทางการส่งจะต้องมีจำนวนเท่ากับจำนวนบิตที่เข้ารหัสแทนตัวอักษรในแต่ละระบบ เช่นการส่งผ่านข้อมูลที่เข้ารหัสแบบแอสกี(ASCII) ก็จำเป็นต้องใช้ช่องทางการส่ง 8 ช่องทาง จึงจะทำให้ทุกบิตวิ่งผ่านสายส่งขนานกันไปได้อย่างรูป

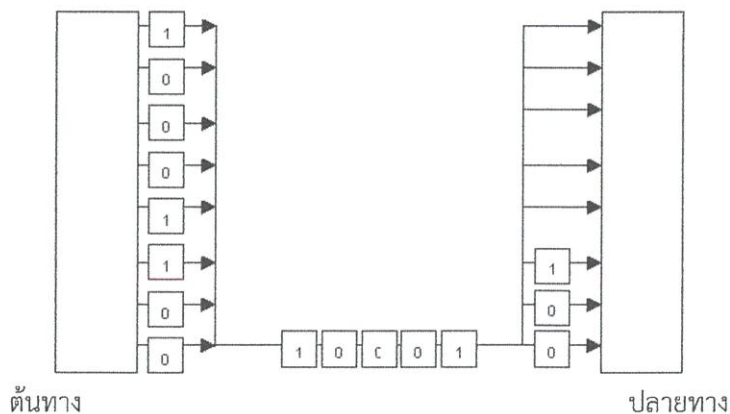


รูปที่ 2.25 การส่งข้อมูลแบบขนาน

จากรูปแสดงให้เห็นว่าทุกบิตที่เข้ารหัสตัวอักษรตัวหนึ่งๆ จะเดินทางขนานกันไป โดยเริ่มจากต้นทางผ่านสายส่งสัญญาณที่มีอยู่ด้วยกัน 8 เส้น ไปยังปลายทาง ดังนั้นปลายทางจะได้รับทุกบิตของตัวอักษรพร้อมๆ กัน การส่งผ่านข้อมูลแบบขนานนี้ ส่วนมากจะใช้ในการส่งผ่านข้อมูลในระยะใกล้ อันได้แก่การเคลื่อนย้ายข้อมูลระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์รอบข้างของมันเช่น ระหว่างเครื่องคอมพิวเตอร์กับเครื่องพิมพ์ เพราะสามารถทำได้ด้วยความเร็วสูง และถ้านำไปใช้ในระยะไกลๆ จะต้องเสีย ค่าใช้จ่ายสูง เนื่องจากต้องใช้สายส่งเท่ากับจำนวนบิตที่ใช้เข้ารหัสแทนข้อมูลตัวอักษร

2.5.7.2 การส่งผ่านข้อมูลแบบอนุกรม(Serial Transmission)

รูปแบบการส่งผ่านข้อมูลในลักษณะนี้ทุกบิตที่เข้ารหัสแทนข้อมูลหนึ่งตัวอักษรจะถูกส่งผ่านไปตามสายส่งเรียงลำดับกันไปทีละบิตในสายส่งเพียงเส้นเดียว ดังรูป



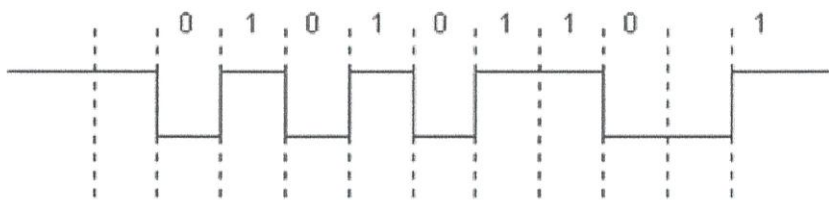
รูปที่ 2.26 การส่งข้อมูลแบบอนุกรม

จากรูปตัวอักษรจะประกอบด้วย 8 บิต เรียงเป็นลำดับ ข้อมูลจะถูกส่งออกมาทีละบิตระหว่างต้นทาง และปลายทาง และปลายทางจะรวบรวมบิตเหล่านี้ทีละบิตจนครบ 8 บิต เป็น 1 ตัวอักษร จะเห็นว่าการส่งข้อมูลแบบนี้จะช้ากว่าแบบขนาน แต่ค่าใช้จ่ายจะถูกลงกว่าแบบขนาน ซึ่งเหมาะสำหรับการส่งระยะทางไกลๆ

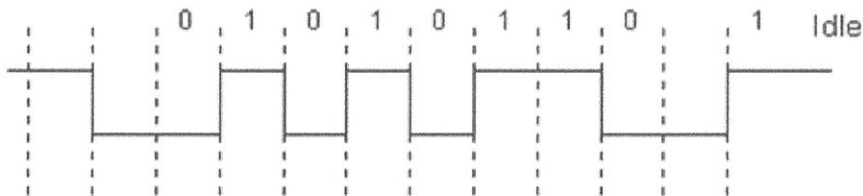
โดยทั่วไปแล้วการส่งข้อมูลนั้นจะประกอบไปด้วยกลุ่มของตัวอักษร ดังนั้นในการส่งข้อมูลแบบอนุกรมนี้จึงเกิดปัญหาขึ้นว่า แล้วต้นทางและปลายทางจะทราบได้อย่างไรว่า จะแบ่งแต่ละตัวอักษรตรงบิตใด จึงเกิดวิธีการสื่อสารข้อมูลขึ้น 2 แบบคือ การสื่อสารแบบอะซิงโครนัส(Asynchronous Transmission) และการสื่อสารแบบซิงโครนัส(Synchronous Transmission)

1. การสื่อสารแบบอะซิงโครนัส(Asynchronous Transmission)

การสื่อสารแบบอะซิงโครนัส หรือเรียกอีกอย่างหนึ่งว่าเป็น การสื่อสารแบบระบุจุดเริ่มต้น และจุดสิ้นสุด (Start-Stop Transmission) ลักษณะของสัญญาณที่ใช้ในการติดต่อสื่อสารกันจะประกอบไปด้วย บิตเริ่มต้น(Start Bit) บิตของข้อมูลที่สื่อสาร(Transmission Data) จำนวน 8 บิต บิตตรวจข้อผิดพลาด(Parity Bit) และบิตสิ้นสุด (Stop Bit) สำหรับบิตตรวจข้อผิดพลาดจะใช้หรือไม่ใช้ก็ได้ ดังนั้นสัญญาณจึงต้องประกอบด้วยส่วนประกอบอย่างน้อย 3 ส่วน ดังรูป



รูปที่ 2.27 การสื่อสารแบบอะซิงโครนัสที่ไม่ได้ใช้พาริตีบิต



รูปที่ 2.28 การสื่อสารแบบอะซิงโครนัสที่ใช้พาริตีบิต

จากรูปจะเห็นว่าขณะที่ไม่มีข้อมูลส่งออกมาสถานะของการส่งจะเป็นแบบว่าง(Idle) ซึ่งจะมีระดับของสัญญาณเป็น 1 ตลอดเวลา เพื่อความแน่ใจว่าปลายทาง หรือฝ่ายรับยังคงติดต่อกับต้นทาง หรือฝ่ายส่งอยู่ เมื่อเริ่มจะส่งข้อมูลสัญญาณของอะซิงโครนัสจะเป็น 0 หนึ่งช่วงสัญญาณนาฬิกา ซึ่งบิตนี้เราเรียกว่าบิตเริ่มต้น ตามหลังของบิตเริ่มต้นจะเป็นบิตข้อมูลสำหรับ 1 ตัวอักษร ตามหลังบิตข้อมูลก็จะเป็นบิตตรวจข้อผิดพลาด แล้วจะตามด้วยบิตสิ้นสุด ถ้าไม่ใช่บิตตรวจข้อผิดพลาด ตามหลังบิตข้อมูลก็จะเป็นบิตสิ้นสุดเลย หลังจากนั้นถ้าไม่มีข้อมูลส่งออกมาสัญญาณจะกลับไปอยู่ที่สถานะแบบว่างอีก เพื่อรอการส่งข้อมูลต่อไป

จะเห็นว่าการสื่อสารแบบอะซิงโครนัสนี้ มีลักษณะเป็นไปทีละตัวอักษร และสัญญาณที่ส่งออกมา มีบางส่วนเป็นบิตเริ่มต้น บิตสิ้นสุด และบิตตรวจข้อผิดพลาด ทำให้ความเร็วในการส่งข้อมูลต่อวินาทีนั้นน้อยลงไป เนื่องจากต้อง

สูญเสียช่องทางการสื่อสารให้กับ บิตเริ่มต้น บิตสิ้นสุด และบิตตรวจข้อผิดพลาด(ถ้ามีใช้) ตลอดเวลา การสื่อสารแบบอะซิงโครนัสนี้มักใช้ในการติดต่อระหว่างคอมพิวเตอร์กับอุปกรณ์รอบข้าง

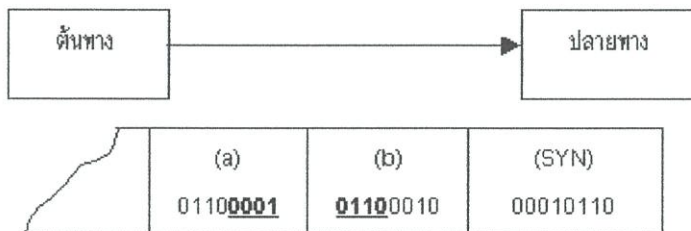
2. การสื่อสารแบบซิงโครนัส(Synchronous Transmission)

การสื่อสารแบบซิงโครนัส จะทำการจัดกลุ่มของข้อมูลเป็นกลุ่มๆ และทำการส่งข้อมูลทั้งกลุ่มไปพร้อมกันในทีเดียว เราเรียกกลุ่มของข้อมูลนี้ว่า บล็อกของข้อมูล (Block of Data) ซึ่งตัวอักษรตัวแรก และตัวถัดไปที่อยู่ในบล็อกเดียวกันจะไม่มีอะไรมาคั่นเหมือนอย่างแบบอะซิงโครนัส ที่ต้องใช้บิตเริ่มต้น และบิตสิ้นสุดคั่นทุกๆ ตัวอักษร แต่จะมีข้อมูลเริ่มต้นซึ่งเป็นลักษณะของบิตพิเศษที่ส่งมาเพื่อให้รู้ว่ำนั้นคือ จุดเริ่มต้นของกลุ่มตัวอักษรที่กำลังส่งเรียงกันเข้ามา เช่น อักขระซิง (SYN Character) โดยที่อักขระซิงมีรูปแบบบิต คือ 00010110 ตัวอย่างของการส่งแสดงได้ดังรูป



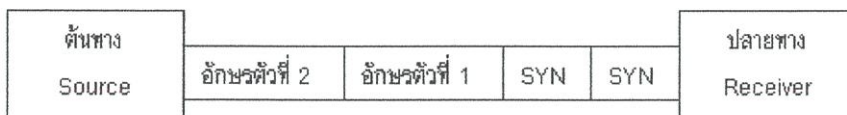
รูปที่ 2.29 อักขระซิง

จากรูปเมื่อปลายทางตรวจพบอักขระซิง หรือ 00010110 แล้วจะทราบได้ทันทีว่าบิตที่ตามมาคือบิตตัวอักษรแต่ละตัว แต่การใช้อักขระซิงเพียงตัวเดียวอาจเกิดข้อผิดพลาดได้ เช่น ถ้าเราส่งตัวอักษร b และตัวอักษร a ติดต่อกันไป ซึ่งตัวอักษร b มีรูปแบบบิตคือ 01100010 และตัวอักษร a มีรูปแบบบิตคือ 01100001 การส่งจะแสดงได้ดังรูป

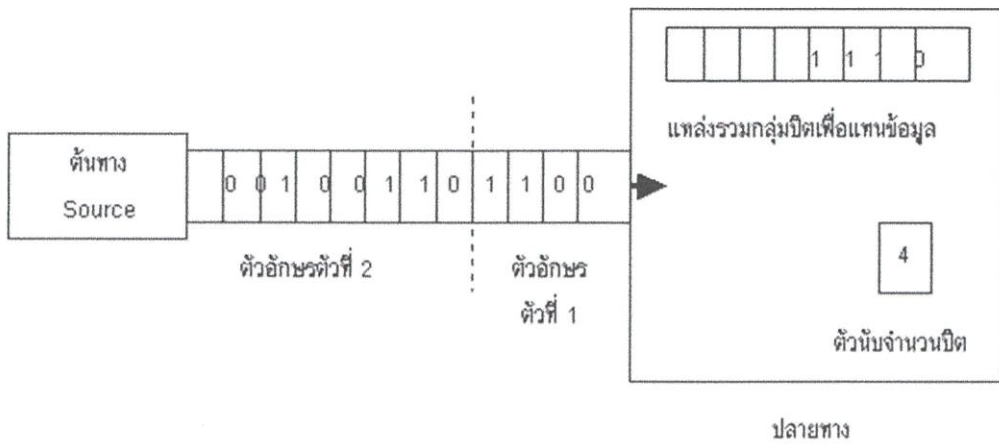


รูปที่ 2.30 ข้อผิดพลาดในการใช้อักขระซิง

จะเห็นว่าเครื่องปลายทางจะตรวจพบอักขระซิงระหว่างบิตของตัวอักษร b และตัวอักษร a ทำให้เข้าใจว่าบิตต่อไปจะเป็นบิตของกลุ่มข้อมูล ซึ่งจะทำให้การรับข้อมูลนั้นเกิดผิดพลาดขึ้นได้ ดังนั้นจึงแก้ปัญหาด้วยการใช้อักขระซิง 2 ตัวต่อกันเป็นลักษณะของบิตพิเศษที่บอกให้ทราบว่าเป็นจุดเริ่มต้นบิตของกลุ่มข้อมูล ตัวอย่างของการใช้อักขระซิง 2 ตัวในการสื่อสารแบบซิงโครนัส และการตัดแฉวของบิตข้อมูลออกเป็นกลุ่มทีละ 8 บิต เพื่อแทนข้อมูลแสดงได้ดังรูป



รูปที่ 2.31 ตัวอย่างการใช้อักขระซิง 2 ตัวในการสื่อสารแบบซิงโครนัส

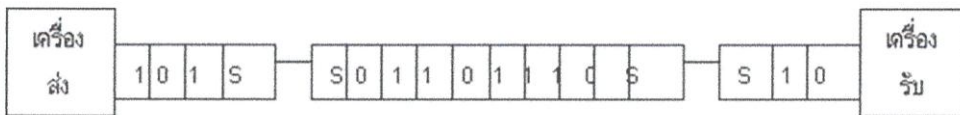


รูปที่ 2.32 แสดงการตัดแฉงของบิตออกเป็นกลุ่มๆ ละ 8 บิต

2.5.7.3 การสื่อสารแบบซิงโครไนส์มักใช้ในการติดต่อระหว่างคอมพิวเตอร์ ประสิทธิภาพของการส่งผ่านข้อมูลแบบอะซิงโครไนส์ และแบบซิงโครไนส์



รูปที่ 2.33 การส่งผ่านข้อมูลแบบซิงโครไนส์



รูปที่ 2.34 การส่งผ่านข้อมูลแบบอะซิงโครไนส์

จากรูปที่ 2.33 แสดงให้เห็นว่าการส่งผ่านข้อมูลแบบซิงโครไนส์นั้นส่วนมากแล้ว ตลอดทางของสายส่งจะใช้ส่งผ่านข้อมูลเต็มตลอดทั้งสาย ส่วนรูปที่ 2.34 แสดงให้เห็นว่าการส่งผ่านข้อมูลแบบอะซิงโครไนส์นั้นสายส่งจะขาดความต่อเนื่องของสัญญาณข้อมูลที่ส่งผ่าน หรือถ้ามีสัญญาณข้อมูลที่ส่งผ่านต่อเนื่องกันเต็มตลอดทั้งสายแล้ว ก็จะมีสัญญาณสูญเสียบางช่องทางในการส่งไปกับการส่งบิตเริ่มต้น และบิตสิ้นสุดของแต่ละตัวอักษร

ตัวอย่างเช่น กรณีที่ส่งผ่านข้อมูลที่อยู่ในรูปของรหัสแอสกี ซึ่งตัวอักษรหนึ่งตัวถูกแทนด้วย 8 บิต ถ้ามีการส่งกลุ่มของข้อมูล 240 ตัวอักษร ในกรณีการส่งผ่านข้อมูลแบบซิงโครไนส์มีการใช้ตัวอักษรซิง 3 ตัว และการส่งผ่าน ข้อมูลแบบอะซิงโครไนส์ไม่มีการใช้บิตตรวจข้อผิดพลาด ดังนั้นเราสามารถคำนวณหาอัตราส่วนระหว่างการส่งข้อมูลได้ ดังนี้

บิตทั้งหมดของตัวอักษรที่ส่งจะได้
 $240 \text{ ตัวอักษร} \times 8 \text{ บิต/ตัวอักษร} = 1920 \text{ บิต}$

1. แบบซิงโคนัส

บิตของตัวอักขระซิงที่ใช้จะได้ SYN 3 ตัว เท่ากับ $3 \times 8 \text{ บิต} = 24 \text{ บิต}$

ผลรวมของบิตที่ต้องส่งทั้งหมด = $1920 + 24 = 1944 \text{ บิต}$

อัตราส่วนระหว่างการส่งข้อมูลที่ต้องส่งจริง กับจำนวนบิตทั้งหมดที่จำเป็นต้องส่งคือ
1920หารด้วย 1944 จะได้ประมาณ 99 %

2. แบบอะซิงโคนัส

บิตเริ่มต้น และบิตสิ้นสุดที่ใช้จะได้ $2 \times 240 = 480 \text{ บิต}$

ผลรวมของบิตที่ต้องส่งทั้งหมด = $1920 + 480 = 2400 \text{ บิต}$

อัตราส่วนระหว่างการส่งข้อมูลที่ต้องส่งจริง กับจำนวนบิตทั้งหมดที่จำเป็นต้องส่งคือ
1920 หารด้วย 2400 จะได้ประมาณ 80 %

2.5.7.4 การใช้บิตตรวจข้อผิดพลาด

บิตตรวจข้อผิดพลาด หรือพาริตีบิต จะเป็นบิตที่ใช้เพื่อทำหน้าที่ตรวจสอบความถูกต้องของข้อมูลที่ส่ง ซึ่งมีอยู่ 2 แบบด้วยกันคือ การตรวจสอบจำนวนคี่(Odd Parity) และการตรวจสอบจำนวนคู่(Even Parity)

การตรวจสอบจำนวนคี่(Odd Parity) หมายถึง บิตตรวจสอบจะต้องนับบิตที่มีค่าของ 1 สำหรับกลุ่มของบิตที่จะส่งและต้องการตรวจสอบอยู่เป็นจำนวนคี่ เช่น ถ้านับบิตที่มีค่าของ 1 ในกลุ่มของบิตที่จะส่ง และต้องการตรวจสอบได้เป็นจำนวนคู่ บิตตรวจสอบนี้จะต้องมีค่าเป็น 1 เพื่อที่จะรวมเป็นจำนวนคี่ แต่ถ้าจำนวนนับได้เป็นจำนวนคี่ บิตตรวจสอบก็จะมีค่าเป็น 0

ตัวอย่าง

สมมติว่าถ้าข้อมูลที่ต้องการส่งมี 7 บิต คือ 0110011 บิตตรวจสอบจำนวนคี่จะต้องมีค่าเป็น 1 เพราะนับบิตที่มีค่าของ 1 ได้เท่ากับ 4 ตัว ซึ่งเป็นเลขคู่ เมื่อรวมกับบิตตรวจสอบจำนวนคี่ที่มีค่าเป็น 1 ก็จะได้เป็น 5 ตัว ซึ่งเป็นเลขคี่และการส่งข้อมูลพร้อมบิตตรวจสอบไปจะได้เป็น 10110011

การตรวจสอบจำนวนคู่(Even Parity) หมายถึง บิตตรวจสอบจะต้องนับบิตที่มีค่าของ 1 สำหรับกลุ่มของบิตที่จะส่งและต้องการตรวจสอบอยู่เป็นจำนวนคู่ เช่น ถ้านับบิตที่มีค่าของ 1 ในกลุ่มของบิตที่จะส่งและต้องการตรวจสอบได้เป็นจำนวนคู่ บิตตรวจสอบนี้จะต้องมีค่าเป็น 0 เพื่อที่จะรวมเป็นจำนวนคู่ แต่ถ้าจำนวนนับได้เป็นจำนวนคี่ บิตตรวจสอบก็จะมีค่าเป็น 1

ตัวอย่าง

สมมติว่าถ้าข้อมูลที่ต้องการส่งมี 7 บิต คือ 0110011 บิตตรวจสอบจำนวนคู่จะต้องมีค่าเป็น 0 เพราะนับบิตที่มีค่าของ 1 ได้เท่ากับ 4 ตัว ซึ่งเป็นเลขคู่ การส่งข้อมูลพร้อมบิตตรวจสอบไปจะได้เป็น 00110011

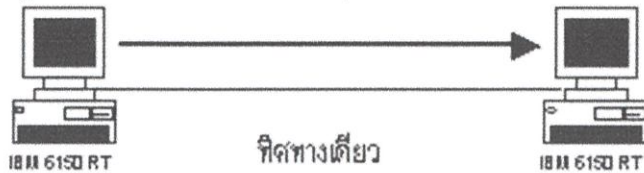
การตรวจสอบความถูกต้องทำได้โดยระหว่างต้นทางและปลายทางจะต้องตกลงกันว่าจะใช้ตัวตรวจข้อผิดพลาดชนิดใด ถ้าใช้ตัวตรวจข้อผิดพลาดแบบจำนวนคี่แล้วเมื่อปลายทางรับข้อมูลจะตรวจสอบจำนวนบิตที่มีค่าเป็น 1 ว่าเป็นจำนวนคี่หรือไม่ ถ้าไม่เป็นจำนวนคี่แสดงว่าข้อมูลเกิดความผิดพลาดขึ้น ปลายทางจะต้องแจ้งให้ต้นทาง

ทราบ อาจจะทำให้ต้นทุนส่งข้อมูลมาใหม่อีกครั้ง ส่วนการใช้ตัวตรวจข้อผิดพลาดแบบจำนวนคู่ก็จะใช้หลักการคล้ายๆกัน

2.5.7.5 ทิศทางของการสื่อสารข้อมูล

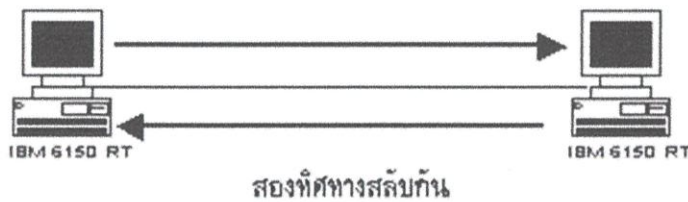
สามารถแบ่งทิศทางการสื่อสารของข้อมูลได้เป็น 3 แบบ คือ

1. แบบทิศทางเดียว(Simplex) เป็นทิศทางการสื่อสารข้อมูลแบบที่ข้อมูลจะถูกส่งจากทิศทางหนึ่งไปยังอีกทิศทางโดยไม่สามารถส่งข้อมูลย้อนกลับมาได้ เช่นระบบวิทยุ หรือโทรทัศน์



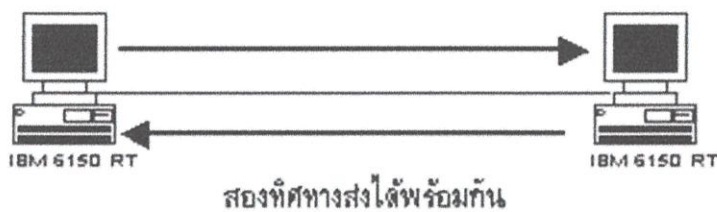
รูปที่ 2.35 การส่งข้อมูลแบบทิศทางเดียว

2. แบบกึ่งสองทิศทาง(Half Duplex) เป็นทิศทางการสื่อสารข้อมูลแบบที่ข้อมูลสามารถส่งกลับกันได้ 2 ทิศทาง แต่จะไม่สามารถส่งพร้อมกันได้ โดยต้องผลัดกันส่งครั้งละทิศทางเท่านั้น เช่น วิทยุสื่อสารแบบผลัดกันพูด



รูปที่ 2.36 การส่งข้อมูลแบบสองทิศทางสลับกัน

3. แบบสองทิศทาง(Full Duplex) เป็นทิศทางการสื่อสารข้อมูลแบบที่ข้อมูลสามารถส่งพร้อม ๆ กันได้ทั้ง 2 ทิศทาง ในเวลาเดียวกัน เช่น ระบบโทรศัพท์



รูปที่ 2.37 การส่งข้อมูลสองทิศทางส่งได้พร้อมกัน

2.6 ระบบเครือข่ายแบบเบสแบนด์และบรอดแบนด์

ระบบเครือข่ายแบบเบสแบนด์(Baseband) เป็นการสื่อสารข้อมูลที่สายสัญญาณหรือตัวกลางในการส่งผ่านสัญญาณ สามารถส่งได้เพียงหนึ่งสัญญาณในเวลาขณะใดขณะหนึ่งเท่านั้น นั่นคืออุปกรณ์ที่ใช้งานสายสัญญาณ

ขณะนั้นจะครอบครองช่องสัญญาณทั้งหมด โดยอุปกรณ์อื่นจะไม่สามารถร่วมใช้งานได้เลย เช่น ระบบโทรศัพท์ เป็นต้น การสื่อสารระหว่างคอมพิวเตอร์ส่วนมากจะเป็นการสื่อสารแบบนี้ รวมทั้งการสื่อสารระหว่างคอมพิวเตอร์ และ อุปกรณ์อื่นๆ ยกเว้นการสื่อสารผ่านระบบเครือข่ายแบบ B-ISDN ซึ่งเป็นแบบบรอดแบนด์

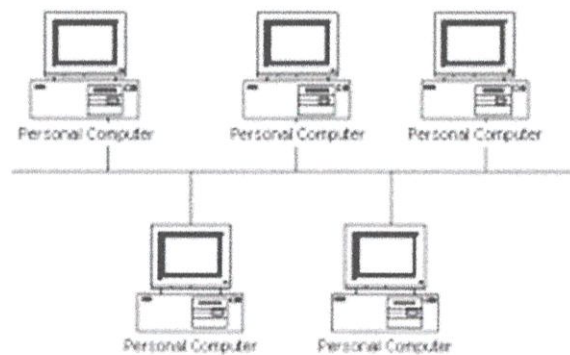
ระบบเครือข่ายแบบบรอดแบนด์(Broadband)เป็นการสื่อสารข้อมูลที่ตัวกลางในการส่งผ่านสัญญาณสามารถส่งสัญญาณผ่านได้หลายๆ ช่องทางพร้อมๆ กัน โดยใช้วิธีแบ่งช่องความถี่ออกจากกัน ทำให้อุปกรณ์ต่างๆ สามารถสื่อสารกันโดยใช้ช่องความถี่ของตนเองผ่านตัวกลางเดียว ตัวอย่างเช่น ระบบเครือข่ายเคเบิลทีวี เป็นต้น

2.6.1 สถาปัตยกรรมของระบบเครือข่าย

สถาปัตยกรรมของระบบเครือข่าย(Network Architecture) หรือโทโปโลยี(Topology) คือลักษณะทางกายภาพ(ภายนอก) ของเครือข่ายซึ่งหมายถึง ลักษณะของการเชื่อมโยงสายสื่อสารเข้ากับอุปกรณ์อิเล็กทรอนิกส์ต่างๆ ภายในเครือข่ายด้วยกันนั่นเอง โทโปโลยีของเครือข่าย แต่ละแบบมีความเหมาะสมในการใช้งาน แตกต่างกันไป จึงมีความจำเป็นที่เราจะต้องทำความเข้าใจลักษณะและคุณสมบัติ ข้อดีและข้อเสียของโทโปโลยีแต่ละแบบ เพื่อนำไปใช้ในการ ออกแบบ พิจารณาเครือข่ายให้เหมาะสมกับการใช้งาน รูปแบบของโทโปโลยีของเครือข่ายหลักๆ มีดังต่อไปนี้

2.6.1.1 โทโปโลยีแบบบัส(Bus Topology)

เป็นโทโปโลยีที่ได้รับความนิยมใช้กันมากที่สุดมาตั้งแต่อดีตจนถึงปัจจุบัน ลักษณะการทำงานของเครือข่าย โทโปโลยีแบบบัส คืออุปกรณ์ทุกชิ้นหรือโหนดทุกโหนด ในเครือข่ายจะต้องเชื่อมโยงเข้ากับสายสื่อสารหลักที่เรียกว่า"บัส"(BUS) เมื่อโหนดหนึ่งต้องการจะส่งข้อมูลไปให้ยังอีกโหนด หนึ่งภายในเครือข่าย จะต้องตรวจสอบให้แน่ใจก่อนว่าบัสว่างหรือไม่ ถ้าหากไม่ว่างก็ไม่สามารถจะส่งข้อมูลออกไปได้ ทั้งนี้เพราะสายสื่อสารหลักมีเพียงสายเดียว ในกรณีที่มีข้อมูลวิ่งมาในบัส ข้อมูลนี้จะวิ่งผ่านโหนดต่างๆ ไปเรื่อยๆ ในขณะที่แต่ละโหนดจะคอยตรวจสอบข้อมูลที่ผ่านมามีเป็นของตนเองหรือไม่ หากไม่ใช่ ก็จะปล่อยให้ข้อมูลวิ่งผ่านไป แต่หากเลขที่อยู่ปลายทาง ซึ่งกำกับมากับข้อมูลตรงกับเลขที่อยู่ของของตน โหนดนั้นก็จะรับข้อมูลเข้าไป



รูปที่ 2.38 โทโปโลยีแบบบัส

ข้อดีข้อเสียของโทโปโลยีแบบบัส

ข้อดี

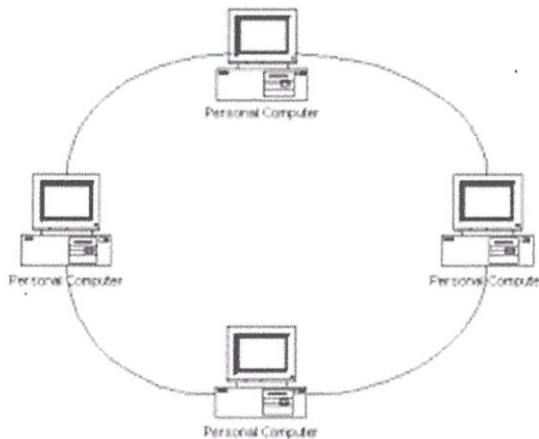
1. ใช้สายส่งข้อมูลน้อยและมีรูปแบบที่ง่ายในการติดตั้ง ทำให้ลดค่าใช้จ่ายในการติดตั้งและบำรุงรักษา
2. สามารถเพิ่มอุปกรณ์ชิ้นใหม่เข้าไปในเครือข่ายได้ง่าย

ข้อเสีย

1. ในกรณีที่เกิดการเสียหายของสายส่งข้อมูลหลัก จะทำให้ทั้งระบบทำงานไม่ได้
2. การตรวจสอบข้อผิดพลาดทำได้ยาก ต้องทำจากหลาย ๆ จุด

2.6.1.2 โทโปโลยีแบบวงแหวน(Ring Topology)

เป็นการเชื่อมต่ออุปกรณ์ต่างๆ เข้ากันเป็นวงกลม ข้อมูลข่าวสารจะถูกส่งจากโหนดหนึ่งไปยังอีกโหนดหนึ่ง วนอยู่ในเครือข่ายไปในทิศทางเดียวเหมือนวงแหวน (ในระบบเครือข่ายรูปวงแหวนบางระบบสามารถส่งข้อมูลได้สองทิศทาง) ในแต่ละโหนดหรือสถานี จะมีรีพีตเตอร์ประจำโหนด 1 ตัว ซึ่งจะทำหน้าที่เพิ่มเติมข่าวสารที่จำเป็นต่อการสื่อสาร ในส่วนหัวของแพ็กเกจข้อมูล สำหรับการส่งข้อมูลออกจากโหนด และมีหน้าที่รับแพ็กเกจข้อมูลที่ไหลผ่านมาจากสายสื่อสาร เพื่อตรวจสอบว่าเป็นข้อมูลที่ส่งมาให้โหนดตนหรือไม่ ถ้าใช่ก็จะคัดลอกข้อมูลทั้งหมดนั้นส่งต่อไปให้กับโหนดของตน แต่ถ้าไม่ใช่ก็จะปล่อยข้อมูลนั้นไปยังรีพีตเตอร์ของโหนดถัดไป



รูปที่ 2.39 โทโปโลยีรูปวงแหวน

ข้อดีข้อเสียของโทโปโลยีรูปวงแหวน

ข้อดี

1. การส่งข้อมูลสามารถส่งไปยังผู้รับหลาย ๆ โหนดพร้อมกันได้ โดยกำหนดตำแหน่งปลายทางเหล่านั้นลง ในส่วนหัวของแพ็กเกจข้อมูล รีพีตเตอร์ของแต่ละโหนดจะตรวจสอบเองว่ามีข้อมูลส่งมาให้ที่โหนดตนเองหรือไม่
2. การส่งข้อมูลเป็นไปในทิศทางเดียวกัน จึงไม่มีการชนกันของสัญญาณข้อมูล

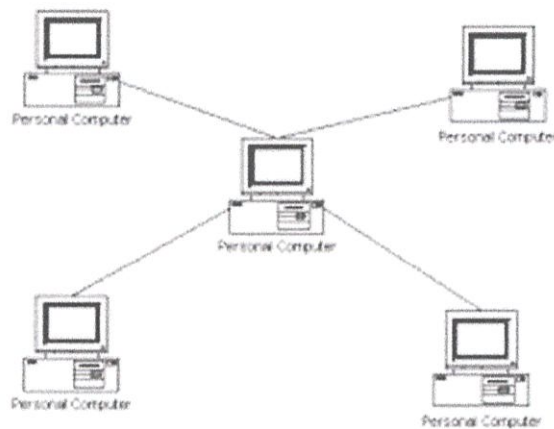
ข้อเสีย

1. ถ้ามีโหนดใดโหนดหนึ่งเกิดเสียหาย ข้อมูลจะไม่สามารถส่งผ่านไปยังโหนดต่อไปได้ และจะทำให้เครือข่ายทั้งเครือข่ายขาดการติดต่อสื่อสาร
2. เมื่อโหนดหนึ่งต้องการส่งข้อมูล โหนดอื่น ๆ ต้องมีส่วนร่วมด้วย ซึ่งจะทำให้เสียเวลา

2.6.1.3 โทโปโลยีรูปดาว(Star Topology)

เป็นการเชื่อมโยงการติดต่อสื่อสารที่มีลักษณะคล้ายรูปดาว หลายแฉก โดยมีสถานีกลาง หรือฮับ เป็นจุดผ่านการติดต่อกันระหว่างทุกโหนดในเครือข่าย สถานีกลางจึงมีหน้าที่เป็นศูนย์กลางควบคุมเส้นทางการสื่อสาร ทั้งหมด นอกจากนี้สถานีกลางยังทำหน้าที่เป็นศูนย์กลางคอยจัดส่งข้อมูลให้กับโหนดปลายทางอีกด้วย การสื่อสารภายในเครือข่ายแบบดาว จะเป็นแบบ 2 ทิศทางโดยจะอนุญาตให้มีเพียงโหนดเดียวเท่านั้นที่สามารถส่งข้อมูลเข้าสู่เครือข่ายได้ จึงไม่มีโอกาสที่หลายๆ โหนดจะส่งข้อมูลเข้าสู่เครือข่ายในเวลาเดียวกัน เพื่อป้องกันการชนกันของสัญญาณข้อมูล เครือข่ายแบบดาว เป็นโทโปโลยีอีกแบบหนึ่งที่เป็นที่นิยมใช้กันในปัจจุบัน³. โทโปโลยีรูปดาว(Star Topology)

เป็นการเชื่อมโยงการติดต่อสื่อสารที่มีลักษณะคล้ายรูปดาว หลายแฉก โดยมีสถานีกลาง หรือฮับ เป็นจุดผ่านการติดต่อกันระหว่างทุกโหนดในเครือข่าย สถานีกลางจึงมีหน้าที่เป็นศูนย์กลางควบคุมเส้นทางการสื่อสาร ทั้งหมด นอกจากนี้สถานีกลางยังทำหน้าที่เป็นศูนย์กลางคอยจัดส่งข้อมูลให้กับโหนดปลายทางอีกด้วย การสื่อสารภายในเครือข่ายแบบดาว จะเป็นแบบ 2 ทิศทางโดยจะอนุญาตให้มีเพียงโหนดเดียวเท่านั้นที่สามารถส่งข้อมูลเข้าสู่เครือข่ายได้ จึงไม่มีโอกาสที่หลายๆ โหนดจะส่งข้อมูลเข้าสู่เครือข่ายในเวลาเดียวกัน เพื่อป้องกันการชนกันของสัญญาณข้อมูล เครือข่ายแบบดาว เป็นโทโปโลยีอีกแบบหนึ่งที่เป็นที่นิยมใช้กันในปัจจุบัน



รูปที่ 2.40 โทโปโลยีแบบดาว

ข้อดีและข้อเสียของโทโปโลยีแบบดาว

ข้อดี

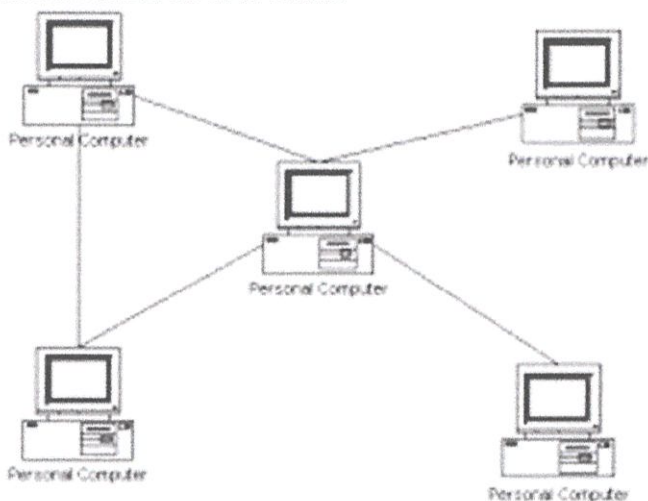
1. การติดตั้งเครือข่ายและการดูแลรักษาทำได้ง่าย
2. หากมีโหนดใดเกิดความเสียหายก็สามารถตรวจสอบได้ง่าย และเนื่องจากใช้อุปกรณ์ 1 ตัวต่อสายส่งข้อมูล 1 เส้น ทำให้การเสียหายของอุปกรณ์ใดในระบบไม่กระทบต่อการทำงานของจุดอื่นๆ ในระบบ
3. ง่ายในการให้บริการเพราะโทโปโลยีแบบดาวมีศูนย์กลางทำหน้าที่ควบคุม

ข้อเสีย

1. ถ้าสถานีกลางเกิดเสียขึ้นมาจะทำให้ทั้งระบบทำงานไม่ได้
2. ต้องใช้สายส่งข้อมูลจำนวนมากว่าโทโปโลยีแบบบัส และ แบบวงแหวน

2.6.1.4 โทโปโลยีแบบผสม(Hybrid Topology)

เป็นเครือข่ายการสื่อสารข้อมูลแบบผสมระหว่างเครือข่ายแบบใดแบบหนึ่งหรือมากกว่า เพื่อความถูกต้องแน่นอน ทั้งนี้ขึ้นอยู่กับความต้องการและภาพรวมขององค์กร



รูปที่ 2.41 โทโปโลยีแบบผสม

2.6.2 มาตรฐานการสื่อสารข้อมูล

การกำหนดมาตรฐานของการสื่อสารข้อมูลนั้น นับว่ามีความจำเป็นอย่างมากสำหรับระบบเครือข่ายที่มีองค์ประกอบของอุปกรณ์ต่างๆ หลากหลายผู้ผลิต ซึ่งอุปกรณ์ทั้งหมดเหล่านั้นจะต้องทำงานเข้ากันได้อย่างราบรื่น การกำหนดมาตรฐานต่างๆ นั้นจะเริ่มตั้งแต่โครงสร้างพื้นฐานของฮาร์ดแวร์ระบบเครือข่าย ได้แก่ ระบบสายเคเบิล อุปกรณ์ในการส่งสัญญาณข้อมูล ตลอดจนถึง เครื่องเซิร์ฟเวอร์ และซอฟต์แวร์ในการสื่อสารบนระบบเครือข่าย เพื่อเป็นการรับประกันว่าส่วนประกอบต่างๆ จะสามารถทำงานร่วมกันได้ ผู้ผลิตฮาร์ดแวร์และซอฟต์แวร์ระบบเครือข่าย จะต้องทำตามคำแนะนำตามมาตรฐานการออกแบบและสร้างผลิตภัณฑ์ ซึ่งกำหนดขึ้นโดย องค์กรมาตรฐานสากล (International Organization for Standardization - ISO) โดยมาตรฐานที่กำหนดขึ้นและได้ประกาศใช้ตั้งแต่ปี คศ.1984 เรียกว่า Open Systems Interconnection Reference Model เรียกสั้นๆ ว่า OSI Reference Model หรือ ISO/OSI Model

2.7 การประยุกต์ใช้งานของระบบเครือข่ายคอมพิวเตอร์

ระบบเครือข่ายทำให้เกิดการสื่อสาร และการแบ่งปันการใช้ทรัพยากรระหว่างเครื่องคอมพิวเตอร์ ซึ่งจะหมายความรวมถึงการสื่อสาร และการแบ่งปันการใช้ข้อมูลระหว่างบุคคลด้วย ซึ่งทั้งหมดนี้คืองานของระบบเครือข่ายนั่นเอง

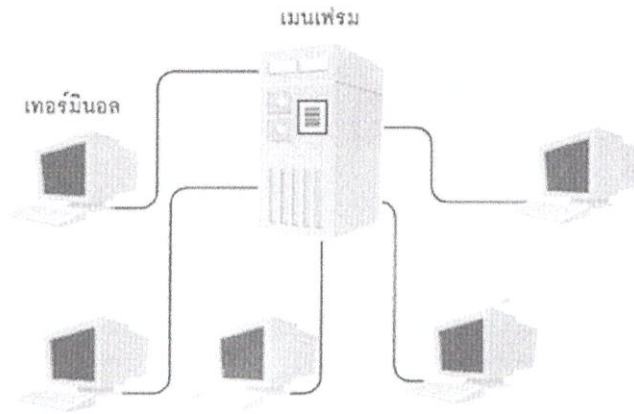
2.7.1 รูปแบบการใช้งานของระบบเครือข่ายคอมพิวเตอร์

ระบบเครือข่ายแบ่งตามลักษณะการทำงาน ได้เป็น 3 ประเภทคือ

1. ระบบเครือข่ายแบบรวมศูนย์กลาง(Centralized Networks)
2. ระบบเครือข่ายแบบเพียร์ทูเพียร์(Peer-to-Peer)
3. ระบบเครือข่ายแบบไคลเอนท์/เซิร์ฟเวอร์(Client/Server)

2.7.1.1 ระบบเครือข่ายแบบรวมศูนย์กลาง

เป็นระบบที่มีเครื่องหลักเพียงเครื่องเดียวที่ใช้ในการประมวลผล ซึ่งจะตั้งอยู่ที่ศูนย์กลางและมีการเชื่อมต่อไปยังเครื่องเทอร์มินอลที่อยู่รอบๆ โดยการเดินสายเคเบิลเชื่อมต่อกันโดยตรง เพื่อให้เครื่องเทอร์มินอลสามารถเข้าใช้งาน โดยส่งคำสั่งต่างๆ มาประมวลผลที่เครื่องกลาง ซึ่งมักเป็นเครื่องคอมพิวเตอร์เมนเฟรมประสิทธิภาพสูง ระบบเครือข่ายแบบรวมศูนย์กลางจะมีราคาสูง และไม่สามารถสนับสนุนระบบการประมวลผลแบบมัลติโปรเซสเซอร์ ได้ เท่ากับระบบเครือข่ายแบบไคลเอนท์/เซิร์ฟเวอร์ปัจจุบันระบบนี้จึงมีความนิยมในการใช้งานลดน้อยลง



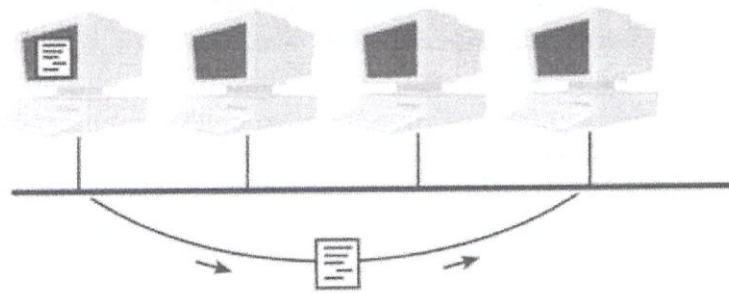
รูปที่ 2.42 ระบบเครือข่ายแบบรวมศูนย์กลาง

2.7.1.2 ระบบเครือข่ายเพียร์ทูเพียร์(Peer-to-Peer)

แต่ละสถานีงานบนระบบเครือข่ายเพียร์ทูเพียร์จะมีความเท่าเทียมกันสามารถที่จะแบ่งปันทรัพยากรให้แก่กันและกันได้ เช่นการใช้เครื่องพิมพ์หรือแฟ้มข้อมูลร่วมกันในเครือข่าย ในขณะที่เดียวกันเครื่องแต่ละสถานีงานก็จะมีขีดความสามารถในการทำงานได้ด้วยตัวเอง(Stand Alone) คือจะต้องมีทรัพยากรภายในของตัวเองเช่น ดิสก์สำหรับเก็บข้อมูล หน่วยความจำที่เพียงพอ และมีความสามารถในการประมวลผลข้อมูลได้

ข้อดีของระบบนี้คือ ความง่ายในการจัดตั้งระบบ มีราคาถูก และสะดวกต่อการบริหารจัดการ ซึ่งมักจะมอบเป็นภาระหน้าที่ของผู้ใช้ในแต่ละสถานีงานให้ รับผิดชอบในการดูแลพิจารณาการแบ่งปันทรัพยากรของตนเองให้กับสมาชิกผู้อื่นในกลุ่ม ดังนั้นระบบนี้จึงเหมาะสมสำหรับสำนักงานขนาดเล็ก ที่มีสถานีงานประมาณ 5-10 เครื่องที่วางอยู่ในพื้นที่เดียวกัน

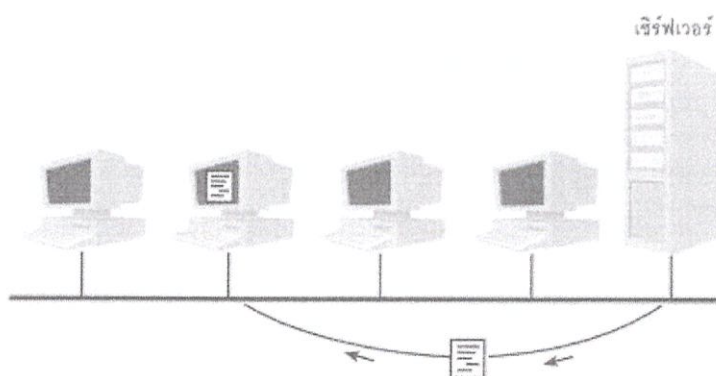
ข้อด้อยของระบบนี้คือ เรื่องการรักษาความปลอดภัยของข้อมูล เนื่องจากไม่มีระบบการป้องกันในรูปแบบของบัญชีผู้ใช้ และรหัสผ่าน ในการเข้าถึงทรัพยากรต่างๆ ของระบบ



รูปที่ 2.43 ระบบเครือข่ายแบบ Peer-to-Peer

2.7.1.3 ระบบเครือข่ายแบบไคลเอนต์/เซิร์ฟเวอร์

เป็นระบบเครือข่ายที่มีประสิทธิภาพสูง และมีการใช้งานกันอย่างกว้างขวางมากกว่าระบบเครือข่ายแบบอื่นที่มีในปัจจุบัน ระบบไคลเอนต์/เซิร์ฟเวอร์สามารถสนับสนุนให้มีเครื่องลูกข่ายได้เป็นจำนวนมาก และสามารถเชื่อมต่อกับเครื่องคอมพิวเตอร์ได้หลายแพลตฟอร์ม ระบบนี้จะทำงานโดยมีเครื่องเซิร์ฟเวอร์ที่ให้บริการ เป็นศูนย์กลางอย่างน้อย 1 เครื่อง และมีการบริหารจัดการทรัพยากรต่างๆ จากส่วนกลาง ซึ่งคล้ายกับระบบเครือข่ายแบบรวมศูนย์กลาง แต่สิ่งที่แตกต่างกันก็คือ เครื่องที่ทำหน้าที่ให้บริการในระบบไคลเอนต์/เซิร์ฟเวอร์นี้จะเป็นเครื่องที่มีราคาไม่แพงมาก ซึ่งอาจใช้เพียงเครื่อง ไมโครคอมพิวเตอร์สมรรถนะสูงในการควบคุมการให้บริการทรัพยากรต่างๆ นอกจากนี้เครื่องลูกข่ายยังจะต้องมีความสามารถในการประมวลผล และมีพื้นที่สำหรับจัดเก็บข้อมูลท้องถิ่นเป็นของตนเองอีกด้วย ระบบเครือข่ายแบบไคลเอนต์/เซิร์ฟเวอร์เป็นระบบที่มีความยืดหยุ่นสูง สนับสนุนการทำงานแบบมัลติโปรเซสเซอร์สามารถเพิ่มขยายขนาดของจำนวนผู้ใช้ได้ตามต้องการ นอกจากนี้ยังสามารถเพิ่มจำนวนเครื่องเซิร์ฟเวอร์สำหรับให้บริการต่างๆ เพื่อช่วยกระจายภาระของระบบได้ ส่วนข้อเสียของระบบนี้ก็คือ มีความยุ่งยากในการติดตั้งมากกว่าระบบเพียร์ทูเพียร์รวมทั้งต้องการบุคลากรเพื่อการบริหารจัดการระบบโดยเฉพาะอีกด้วย



รูปที่ 2.44 ระบบเครือข่ายแบบไคลเอนต์/เซิร์ฟเวอร์

2.7.2 การแบ่งปันการใช้ทรัพยากรของระบบเครือข่าย

เมื่อก้าวถึงทรัพยากรบนระบบเครือข่าย ในที่นี้จะครอบคลุมถึงทุกสิ่งทุกอย่างที่อำนวยความสะดวกกับผู้ใช้ในระบบ เช่น แฟ้มข้อมูล ฐานข้อมูล รูปภาพและสไลด์สำหรับเสนองาน ตลอดจนอุปกรณ์ฮาร์ดแวร์ต่างๆ ที่ติดตั้งอยู่บนระบบเครือข่าย ได้แก่ ฮาร์ดไดรฟ์ที่มีการแชร์ไว้สำหรับให้บริการพื้นที่สำหรับเก็บข้อมูลส่วนบุคคล หรือใช้เป็นที่เก็บข้อมูลชั่วคราวสำหรับการโยกย้ายแฟ้มข้อมูลระหว่างเครื่องคอมพิวเตอร์ เครื่องพิมพ์บนระบบเครือข่าย เครื่องโทรสาร เป็นต้น นอกจากนี้สิ่งต่างๆ เหล่านี้ ทรัพยากรที่มีความสำคัญอย่างยิ่งยวดในสภาพการณ์ปัจจุบันนั้นก็คือนิวส์สารข้อมูล ซึ่งอาจเป็นการส่งข่าวสารระหว่างผู้ใช้ด้วยกันเอง หรือการกระจายข่าวสารที่มีความสำคัญจากผู้บริหารหรือฝ่ายสารสนเทศขององค์กร สภาพแวดล้อมของระบบเครือข่ายที่เอื้ออำนวยต่อการติดต่อสื่อสารระหว่างผู้ใช้งานในระบบนี้ นับว่าเป็นปัจจัยสำคัญอันหนึ่งที่จะทำให้องค์กรประสบความสำเร็จในการประยุกต์ใช้งานระบบเครือข่ายได้อย่างเต็ม ประสิทธิภาพ

2.8 การประยุกต์ใช้งานเครือข่ายอินเทอร์เน็ต

เครือข่ายอินเทอร์เน็ต คือระบบเครือข่ายสากล ที่เกิดจากการรวมระบบเครือข่ายขนาดเล็กให้สื่อสาร และแลกเปลี่ยนข้อมูลซึ่งกันและกันได้ โดยเป็นเครือข่ายที่มีเทคโนโลยีระดับสูงซึ่งเปิดกว้างสู่สาธารณะอย่างแพร่หลายหรืออาจกล่าวได้ว่า อินเทอร์เน็ตเป็นการผสมผสานกันของระบบเครือข่ายที่แตกต่างกันทั่วโลกให้เป็นหนึ่งเดียวกัน สำหรับ ผู้ใช้ส่วนใหญ่จะสามารถเข้าถึงและใช้งานอินเทอร์เน็ตได้ โดยเชื่อมต่อผ่านทางโมเด็ม และสายโทรศัพท์ แต่ในความเป็นจริงนั้นเทคโนโลยีและอุปกรณ์ต่างๆ ที่ใช้ในเครือข่ายนี้มีอยู่มากมายหลากหลาย ซึ่งมีการพัฒนาและเปลี่ยนแปลงอยู่ตลอดเวลา โดยจะประกอบไปด้วยเครื่องเซิร์ฟเวอร์ (Server) เกตเวย์ (Gateway) เราเตอร์ (Router) และสายสื่อสาร เป็นจำนวนมากที่เชื่อมต่อสิ่งเหล่านี้เข้าด้วยกันเป็นเครือข่ายอินเทอร์เน็ต

ผู้ให้บริการเชื่อมต่อเครือข่ายอินเทอร์เน็ต หรือ ISP (Internet Service Provider) นั้นเป็นเสมือนผู้จำหน่ายที่จัดให้มีเส้นทางเข้าไปสู่เครือข่ายอินเทอร์เน็ต โดย ISP แต่ละแห่งจะมีช่องทางการเชื่อมต่อของตนเองอยู่กับ Backbone ของอินเทอร์เน็ต ด้วยสายการสื่อสารความเร็วสูงเช่น T1 เป็นต้น แม้ว่าเครือข่ายอินเทอร์เน็ตในระยะแรกจะอยู่บนพื้นฐานของการส่งข้อมูลที่เป็นข้อความ (Text) และรูปภาพ (Graphic) แต่ในปัจจุบัน ปริมาณและชนิดของข้อมูลบนเครือข่ายนี้ได้เพิ่มขึ้นอย่างมหาศาล ซึ่งมีทั้งภาพเคลื่อนไหวรูปแบบต่างๆ (Graphic Animation) ข้อมูลเสียง (Audio) และวิดีโอ (Video) เป็นต้น

2.8.1 การตั้งชื่อบนระบบเครือข่ายอินเทอร์เน็ต

เครือข่ายอินเทอร์เน็ตสร้างขึ้นจากแนวความคิดที่มีแบบแผน โดยมีการออกแบบและจัดการโดเมน (Domain) อย่างเป็นระบบเพื่อไม่ให้เกิดความสับสน และมีการเติบโตเป็นลำดับอย่างต่อเนื่องโดเมนเนมซิสเต็ม (DNS) เป็นระบบจัดการแปลงชื่อไปเป็นหมายเลขไอพีแอดเดรส (name-to-IP address mapping) โดยมีโครงสร้างของฐาน ข้อมูลแบบลำดับชั้น (hierarchical) ที่ประกอบด้วย โดเมนระดับบนสุด (Top-level Domain) โดเมนระดับรอง (Second-level Domain) และโดเมนย่อย (Sub domain) ตัวอย่างเช่น www.gnu.org โดยที่ .org คือโดเมนระดับบนสุด ซึ่งแสดงถึงเป็นประเภทขององค์กรซึ่งไม่ได้ค้ากำไร .gnu คือโดเมนระดับรองซึ่งเป็นชื่อย่อของโครงการ GNU's Not Unix ซึ่งอยู่ภายใต้โครงการฟรีซอฟต์แวร์ฟาวนด์ชัน (Free Software Foundation) และภายใต้ชื่อโดเมนดังกล่าวอาจมีโดเมนย่อยอื่นๆ ได้อีกเป็นจำนวนมาก

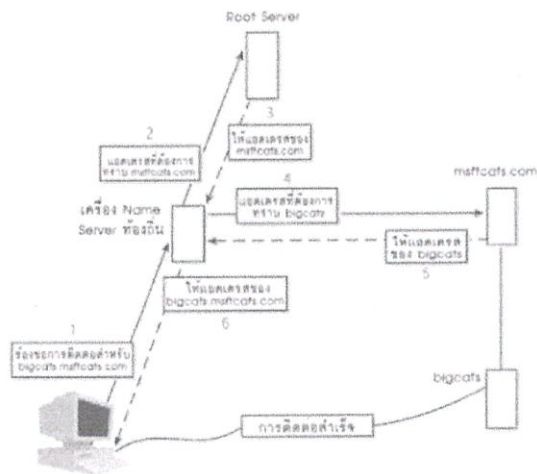
ข้อกำหนดที่สำคัญของโดเมนเนมซิสเทมคือชื่อในโดเมนลำดับบนสุดนั้น ได้มีการกำหนดชื่อเฉพาะซึ่งระบุรายละเอียดของกลุ่มเอาไว้ชัดเจนแล้ว ดังนี้

- .mil แทนกลุ่มของหน่วยงานทางทหารของสหรัฐอเมริกา
- .gov แทนกลุ่มของหน่วยงานของรัฐบาล
- .com แทนกลุ่มขององค์กรหรือบริษัทเอกชน
- .net แทนองค์กรที่ทำหน้าที่เป็นผู้ให้บริการเครือข่าย
- .edu แทนสถาบันการศึกษา
- .org แทนองค์กรหรือสมาคมต่างๆ ที่ดำเนินการโดยไม่ได้หวังผลกำไร
- .xx ใช้ตัวอักษร 2 ตัวแทนชื่อประเทศ

ต่อมาได้มีการแก้ไขเพิ่มเติมกลุ่มของ โดเมนลำดับบนสุดอีก 7 กลุ่มคือ

- .firm แทนองค์กรหรือบริษัทห้างร้านทั่วไป
- .store แทนบริษัทที่มีธุรกรรมทางการค้า
- .web แทนเว็บไซต์ที่ให้ข้อมูลต่างๆ
- .arts แทนกลุ่มที่มีกิจกรรมทางด้านประเพณีและวัฒนธรรม
- .rec แทนองค์กรหรือหน่วยงานที่ทำงานด้านนันทนาการ
- .info แทนองค์กรที่เป็นผู้ให้บริการข้อมูล
- .nom สำหรับบุคคลทั่วไป

ฐานข้อมูลโดเมนเนมซิสเทมจะทำการจับคู่ระหว่างชื่อที่ผู้ใช้จดจำได้ง่าย เข้ากับไอพีแอดเดรสโดยทำงานคล้ายสมุดโทรศัพท์ที่จับคู่ชื่อบุคคลต่างๆ กับหมายเลขโทรศัพท์ ในการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์นั้น จะมีการทำงานของโดเมนเนมซิสเทมเพื่อค้นหาหมายเลขไอพีของเครื่องเป้าหมายโดยมีขั้นตอนดังต่อไปนี้



รูปที่ 2.45 การทำงานของโดเมนเนม

เมื่อเครื่องคอมพิวเตอร์ A ต้องการติดต่อไปยังไซต์ B ซึ่งมีชื่อโดเมน bigcats.msftcats.com ขั้นตอนแรกเครื่องคอมพิวเตอร์ A จะต้องติดต่อเข้าไปยังโดเมนเนมซิสเทมเซอร์เวอร์ท้องถิ่นโดยแจ้งความจำนงว่าจะติดต่อไปยังไซต์ B ถ้าเครื่องโดเมนเนมซิสเทมเซอร์เวอร์ท้องถิ่นมีข้อมูลของไซต์ B อยู่แล้ว ก็จะทำการส่งที่อยู่คือไอพีแอดเดรสของไซต์ B กลับมาให้ทันที แต่ถ้าเครื่องโดเมนเนมซิสเทมเซอร์เวอร์ท้องถิ่นไม่มีข้อมูลอยู่ในฐานข้อมูล ก็จะส่งคำร้องขอนั้นขึ้นไปยังเครื่องโดเมนเนมซิสเทมเซอร์เวอร์ที่อยู่ในโดเมนระดับสูงกว่าถัดขึ้นไปตามลำดับ จนถึงเซอร์เวอร์ระดับบนสุดคือรูทเซอร์เวอร์ ซึ่งรูทเซอร์เวอร์แม้จะไม่สามารถค้นหาไอพีแอดเดรส ทั้งหมดสำหรับไซต์ B ได้ แต่ก็จะให้ข้อมูลกับโดเมนเนมซิสเทมเซอร์เวอร์ท้องถิ่นว่าจะต้องติดต่อไปยังหมายเลขไอพีแอดเดรสของ maftcats.com ซึ่งจะสามารถจะให้ไอพีแอดเดรสของ bigcat.msftcats.com ได้อย่างสมบูรณ์ กระบวนการค้นหาชื่อทั้งหมดนี้เรียกว่า การสอบถามแบบซ้ำๆ(Iterative Query) เนื่องจากการร้องขอจะถูกส่งไปซ้ำๆ หลายรอบโดยผ่านเครื่องโดเมนเนมซิสเทมเซอร์เวอร์ขึ้นไปตามลำดับชั้น จนกระทั่งทราบไอพีแอดเดรสที่แน่นอนหรือมีเวลานั้นก็จะแจ้งว่าไม่พบไซต์ดังกล่าว

2.8.2 จดหมายอิเล็กทรอนิกส์

จดหมายอิเล็กทรอนิกส์หรืออีเมล ได้มีการเริ่มใช้งานกันมานานตั้งแต่ยุคของเครื่องเมนเฟรมและมินิคอมพิวเตอร์ และมีการพัฒนามาโดยลำดับ จนถึงยุคของอินเทอร์เน็ตนี่คืออีเมลได้มีการพัฒนาโดยมีการทำงานในรูปแบบของไคลเอนต์/เซิร์ฟเวอร์ ซึ่งมีส่วนประกอบสองส่วนหลัก ดังนี้

1. ยูเซอร์เอเจนต์(User Agent) เป็นโปรแกรมคอมพิวเตอร์ทางด้านผู้ใช้งาน มีสองส่วนคือส่วนของผู้ส่งและส่วนของผู้รับ โดยยูเซอร์เอเจนต์จะติดต่อเข้าสู่เซิร์ฟเวอร์ของตนโดยผ่านระบบแลน หรือเชื่อมต่อผ่านโมเด็มยูเซอร์เอเจนต์นี้ก็คือส่วนที่มากับโปรแกรมไคลเอนต์ของอีเมล เช่น เอาลูคเอกซ์เพรส(Outlook Express) หรือ ยูโดรา(Eudora) เป็นต้น
2. เมลล์ทรานส์เฟอร์เอเจนต์(Mail Transfer Agent) เป็นโปรแกรมคอมพิวเตอร์ที่จะส่งอีเมลจากต้นทางไปยังผู้รับปลายทาง ซึ่งจะต้องส่งผ่านเครื่องจำนวนมากที่เชื่อมต่อกันในเครือข่าย โดยโปรแกรมเหล่านี้จะช่วยกันส่งต่ออีเมลเป็นทอดๆจนไปถึงเครื่องที่มีที่อยู่เมลบ็อกซ์ของผู้รับ และหากไม่สามารถส่งเมลถึงผู้รับได้ ไม่ว่าจะด้วยเหตุผลใดๆ ก็ตาม ก็จะทำหน้าที่ส่งข้อความแสดงความผิดพลาด (Error Mail) กลับมายังผู้ส่งได้อีก เครื่องที่เมลล์ทรานส์เฟอร์เอเจนต์ทำงานอยู่ มักจะเป็นเครื่องเมลล์เซอร์เวอร์ ซึ่งมีเมลล์บ็อกซ์ของผู้ใช้อยู่ด้วย

การทำงานของระบบอีเมลโดยสรุปมีสองประเภทคือ การส่งและการรับ การส่งอีเมลจะกระทำโดยใช้โปรโตคอล ซิมเปิลเมลล์ทรานส์เฟอร์โปรโตคอล(Simple Mail transfer Protocol) โดยจะมีการทำงานในขณะที่ ยูเซอร์เอเจนต์ส่งอีเมลไปยังเมลล์ทรานส์เฟอร์เอเจนต์และขณะที่เมลล์ทรานส์เฟอร์เอเจนต์รับส่งอีเมลระหว่าง เมลล์ทรานส์เฟอร์เอเจนต์ ด้วยกัน สำหรับการรับอีเมลนั้นมีโปรโตคอลที่นิยมใช้งานกันแพร่หลายอยู่ 2 แบบได้แก่ โปสออฟฟิศโปรโตคอล(Post Office Protocol) และอินเทอร์เน็ตเมสเซจแอกเซสโปรโตคอล(Internet Message Access Protocol) ซึ่งทั้งสอง โปรโตคอลนี้จะทำการดาวน์โหลดอีเมลจากเครื่องเมลล์เซิร์ฟเวอร์ไปยังเครื่องไคลเอนต์

2.8.3 การรับส่งแฟ้มข้อมูลบนอินเทอร์เน็ต

ไฟล์ทรานส์เฟอร์โปรโตคอล(File Transfer Protocol) เป็นโปรโตคอลพื้นฐานที่ใช้ในการถ่ายโอนแฟ้มข้อมูลบนอินเทอร์เน็ต ซึ่งกำเนิดมาจากคำสั่งพื้นฐานของระบบปฏิบัติการยูนิกซ์ คุณสมบัติพื้นฐานของไฟล์ทรานส์เฟอร์โปรโตคอลก็คือสามารถโหลดไฟล์มาจากเซิร์ฟเวอร์(Download) หรือส่งไฟล์ไปเก็บไว้ที่เซิร์ฟเวอร์(Upload) ได้ โปรแกรมไฟล์ทรานส์เฟอร์โปรโตคอลจะมีความทำงานแบบไคลเอนต์เซิร์ฟเวอร์ โดยได้รับการพัฒนาขึ้นตามโปรโตคอลที่ซีพี ซึ่งจะต้องมีการติดต่อเพื่อจองช่องสื่อสารก่อนทำการสื่อสารจริง เรียกว่าคอนเน็คชันออเรียนท์(Connection-oriented) ในการใช้งานไฟล์ทรานส์เฟอร์โปรโตคอลเพื่อเริ่มการติดต่อสื่อสารนั้น จะต้องระบุหมายเลขไอพีปลายทาง และจะมีการตรวจสอบชื่อบัญชีของผู้ใช้ โดยผู้ใช้อาจต้องแจ้งรหัสล็อกอินและรหัสผ่านก่อนจึงจะเข้าใช้งานได้

เซิร์ฟเวอร์ของไฟล์ทรานส์เฟอร์โปรโตคอลบางแห่งจะทำหน้าที่ให้บริการแก่ผู้ใช้ทั่วไปเพื่อดาวน์โหลดไฟล์ ซึ่งอาจเป็นไฟล์ข้อมูล รูปภาพ หรือโปรแกรมต่างๆ เซิร์ฟเวอร์เหล่านี้จะมีรหัสผู้ใช้กลางที่ยอมให้ผู้ใช้สามารถเข้าระบบเพื่อใช้งานเซิร์ฟเวอร์ไฟล์ทรานส์เฟอร์โปรโตคอลได้ทุกคน ชื่อที่ระบบส่วนใหญ่จะตั้งให้ใช้คือผู้ใช้นิรนาม(anonymous) คือไม่ต้องระบุชื่อที่แท้จริง ดังนั้นผู้ใช้เพียงแต่ทราบชื่อของเซิร์ฟเวอร์ ชื่อไฟล์ และชื่อไอดีเรคทอรี ก็สามารถดาวน์โหลดไฟล์ที่ต้องการได้ รหัสผู้ใช้นิรนามนี้เป็นชื่อสำหรับล็อกอินที่อาจมีการจำกัดสิทธิในการใช้งานต่างๆ เช่น ผู้ใช้อาจจะเห็นข้อมูลได้เฉพาะบางไอดีเรคทอรีเท่านั้น และไม่สามารถส่งไฟล์ไปเก็บยังเซิร์ฟเวอร์ได้ ในการล็อกอินของผู้ใช้นิรนามนั้น อาจต้องใส่รหัสผ่าน เป็นที่อยู่อีเมลของผู้ใช้ หรืออาจไม่ต้องใส่รหัสผ่านเลยก็ได้ ขึ้นอยู่กับข้อกำหนดของแต่ละระบบ

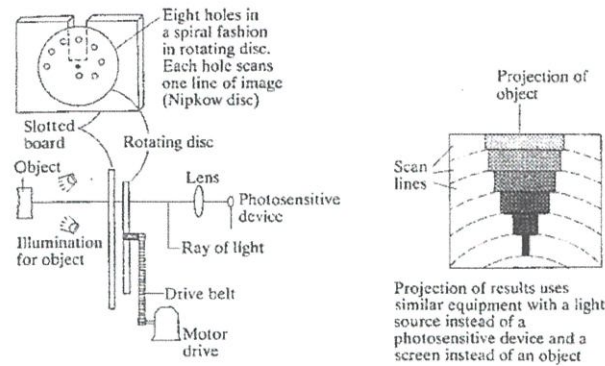
2.9 การได้มาของรูปภาพ (Image Acquisition)

รูปที่ได้จากการใช้กล้องถ่ายนั้นมาจากอุปกรณ์หลายๆอย่าง มีวิธีการจัดเก็บที่แตกต่างกัน ดังนั้นในหัวข้อนี้จะกล่าวถึงอุปกรณ์ที่ใช้ในการนำรูปภาพเข้าสู่คอมพิวเตอร์ และวิธีจัดเก็บของแต่ละอุปกรณ์

2.9.1 รูปที่ได้มาจากความเข้มของแสง (Intensity Images)

ความเข้มของแสงสามารถที่จะเปลี่ยนมาให้เป็นรูปภาพได้โดยใช้อุปกรณ์ทางด้านอิเล็กทรอนิกส์ที่มีการรับรู้ทางด้านแสงเช่นโฟโตเซนซิทีฟเซลล์(Photosensitive cell) ซึ่งอุปกรณ์ดังกล่าวนี้สามารถนำไปเป็นชิ้นส่วนของกล้องถ่ายรูปเพื่อใช้ในการแยกสัญญาณของแสงออกมาเป็นแต่ละจุดของภาพ ซึ่งถ้าต้องการรูปภาพทั้งหมดก็สามารถทำได้ด้วยวิธีการเดียวกันโดยการแสกนจุดทุกจุดที่ต้องการ

ที่กล่าวมานั้นเป็นหลักการที่ Baird ได้คิดค้นมาใช้ในระบบโทรทัศน์เป็นระบบแรก อุปกรณ์ที่ใช้ในการแสกนจะเป็นลักษณะแผ่นกลม ๆ มีช่องที่เจาะไว้ ซึ่งแต่ละช่องจะอนุญาตให้แสงจาก 1 จุดผ่านได้เท่านั้น เมื่อแผ่นดังกล่าวหมุนไปรอบ ๆ รูปภาพ ตัวที่ทำหน้าที่รับสัญญาณจะได้รับสัญญาณที่ผ่านมาจากแสงแต่ละช่อง ซึ่งในการแสกนนั้นจำทำการแสกนแบบที่ละบรรทัด(ดูรูปที่ 2.46 ประกอบ)



รูปที่ 2.46 แสดงการใช้กระบวนการจับภาพของวิดีโอและการแสดงผล

2.9.2 การจับรูปภาพแบบต่อเนื่อง (Real-time Capture)

ถ้ารูปภาพถูกจับแบบต่อเนื่องจำเป็นจะต้องใช้รูปภาพอย่างน้อย 25 รูปต่อหนึ่งวินาที เพื่อที่จะไม่ให้เห็นรูปที่ไม่ต่อเนื่อง ถ้ารูปที่ต้องการจัดเก็บเป็นโมโนมี 256 ระดับ จะต้องใช้คุณสมบัติดังนี้

1. การจัดเก็บใน 1 วินาทีจะต้องใช้เนื้อที่ 6.25 เมกะไบต์
2. ในการถ่ายโอนข้อมูลจำนวน 6.25Mbytes ต่อวินาทีไปยังเครื่องคอมพิวเตอร์จำเป็นต้องใช้อุปกรณ์เพิ่มเติม เช่นพาราเรลอินเตอร์เฟส(Parallel Interface) หรือซีเรียลอินเตอร์เฟส(Serial Interface) โดยมีอัตราการความเร็วอย่างน้อย 38 kbps
3. คอมพิวเตอร์จะจัดเก็บข้อมูลดังกล่าวไว้ในที่มีความเร็วพอเช่น ฮาร์ดดิสก์แต่ไม่ควรจะจัดเก็บไว้ในฟลอปปีดิสก์เนื่องจากมีความเร็วในการถ่ายโอนต่ำ
4. การนำข้อมูลลงสู่ฮาร์ดดิสก์จะทำการถ่ายโอน 1 นาทีต่อครั้ง

2.9.3 รูปภาพแบบสี (Color Images)

การนำภาพแบบมีสีเข้าสู่คอมพิวเตอร์จะเปรียบเสมือนการนำภาพเข้าสู่คอมพิวเตอร์ 3 รูปในเวลาเดียวกัน ซึ่งรายละเอียดขึ้นอยู่กับระบบชนิดของสี ในระบบอาร์จีบี(RGB) จะใช้ความเข้มเป็น สีแดง สีเขียว และ สีน้ำเงิน ในกล้องถ่ายรูปจะทำการแยกแถบสีดังกล่าวออกจากกันจึงทำให้ได้รูปภาพ 3 รูป เป็นของสีแดง สีเขียว และสีน้ำเงินตามลำดับ

2.9.4 รูปภาพแบบเป็นช่วง (Range Images)

ตัวอย่างของรูปภาพแบบเป็นช่วง ๆ เช่น จากการรับสัญญาณของเรดาร์ การรับภาพชนิดนี้ไม่ได้ใช้ความเข้มของแสง แต่ใช้ระยะห่างของวัตถุแทน ซึ่งระบบนี้จะใช้ในการชี้้นำการขับเคลื่อนของยานพาหนะต่าง ๆ ในการเคลื่อนของพาหนะจำเป็นจะต้องรู้ระยะห่างของวัตถุต่าง ๆ เพื่อที่จะให้การเดินทางเป็นไปด้วยความปลอดภัย ระบบความเข้มของแสงจะไม่สามารถนำมาใช้บอกระยะห่างของวัตถุได้

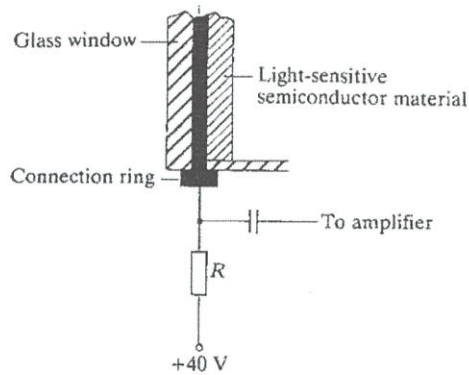
2.9.5 ระบบกล้องวิดีโอ

ระบบกล้องวิดีโอได้พัฒนาอย่างต่อเนื่อง โดยมีทั้งกล้องวิดีโอแบบโมนี กล้องวิดีโอแบบมีสี นอกจากนี้ยังได้พัฒนาในเรื่องของขนาด ความสว่าง และความคมชัดของรูปภาพ

2.9.6 กล้องวิดีโอแบบขาวดำ

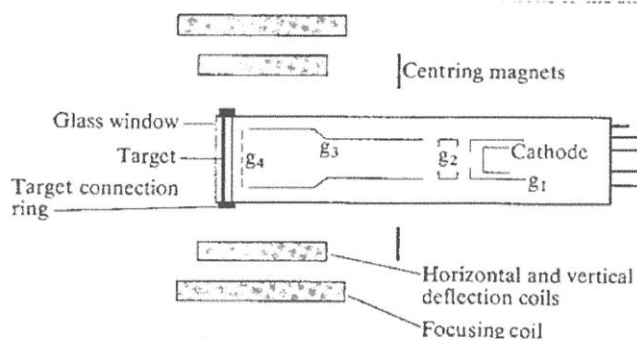
อุปกรณ์หลัก ๆ ของระบบนี้ ประกอบด้วย หลอดแก้วที่มีตัวรับรู้เกี่ยวกับแสงหุ้มอยู่และมีฟิล์มที่ประกอบด้วยออกไซด์ของดีบุกบาง ๆ หุ้มอยู่อีกชั้นหนึ่ง (ดูรูป 2.47 และ 2.48 ประกอบ) ตัวรับรู้เกี่ยวกับแสงทำมาจากวัสดุที่เป็นกึ่งตัวนำ ความต้านทานจะเพิ่มขึ้นเมื่อได้รับความเข้มของแสงสว่างเพิ่ม กลุ่มของเลนส์จะหันไปสู่วัตถุที่ต้องการแสกน การแสกนจะเริ่มจากหลอดแคโทดที่อยู่ตอนท้ายของหลอดแก้ว ปล่อยแสงที่เป็นลำของอิเล็กตรอนออกมา ลักษณะของการแสกนนี้จะเป็นแบบ Raster (แสกนตามแนวตั้งทีละบรรทัด) พื้นที่ของประจุอิเล็กตรอนจะขึ้นอยู่กับความเข้มของแสงที่ได้รับกลับมาจากวัตถุ

2.9.7 กล้องวิดีโอแบบสี



รูป 2.47 โครงสร้างของกล้องวิดีโอ

กล้องวิดีโอแบบสีอย่างง่ายจะเป็นส่วนขยายของกล้องวิดีโอแบบขาวดำ การกรองสีจะถูกนำมาใช้ โดยอยู่ระหว่างเลนส์และวัตถุ อุปกรณ์ที่ใช้ในการกรองที่รู้จักกันดีได้แก่ฟิลเตอร์รวมแสง (Filter Integrated Color) ซึ่งประกอบด้วยหลอดแก้วที่มีตัวกรองอย่างดี แต่ละบริเวณที่แตกต่างกันของวัตถุจะใช้สีแดง น้ำเงิน หรือ สีขาวเป็นตัวแยกค่าที่ได้จากการวัดของแต่ละจุดจะแตกต่างกันออกไปตามค่าของความเข้มของแสงแต่ละจุด



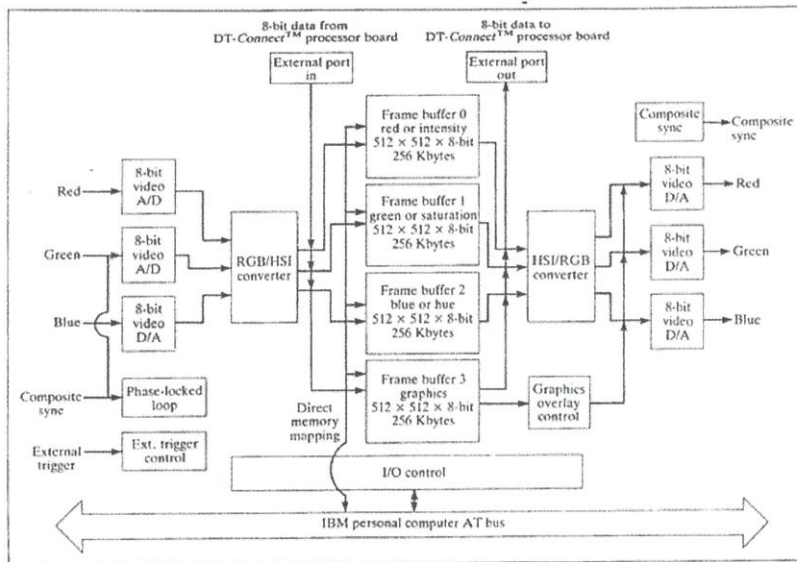
รูปที่ 2.48 ภาพตัดของกล้องวิดีโอ

2.9.8 การจับภาพ (Capture)

ผลลัพธ์ที่ได้จากกล้องวิดีโอส่วนมากจะเป็นสัญญาณในระบบอาร์จีบี ซึ่งถ้านำอุปกรณ์ดังกล่าวมาต่อกับคอมพิวเตอร์ จำเป็นต้องใช้อุปกรณ์เพิ่มเติมในการแปลงภาพดังกล่าวเข้าสู่คอมพิวเตอร์ โดยที่อุปกรณ์นั้นจะทำหน้าที่รับภาพเป็นเฟรม ๆ โดยอาจจะใช้หน่วยความจำของคอมพิวเตอร์เพื่อที่จะแปลงสัญญาณอนาลอกเป็นดิจิทัล หรืออาจมีหน่วยความจำเป็นของตนเอง ซึ่งหน่วยความจำดังกล่าวเรียกว่าเฟรมบัฟเฟอร์(Frame Buffer)

2.9.9 การแปลงสัญญาณจากอนาลอกไปยังสัญญาณดิจิทัล

ในส่วนของหน่วยความจำของคอมพิวเตอร์ที่ใช้นั้น จะต้องมียุทธวิธีที่ทำหน้าแปลงความต่างศักย์ของโวลต์ซึ่งเป็นอนาลอก ให้เป็นสัญญาณดิจิทัลที่เป็นบิต ๆ เพื่อที่จะให้คอมพิวเตอร์สามารถใช้งานได้ ความเร็วในการแปลงนี้ขึ้นอยู่กับความเร็วของอนาล็อกทูดิจิทัล(A/D) แต่ละตัว ในระบบการเก็บภาพแบบต่อเนื่องจำเป็นต้องใช้การแปลงอย่างน้อย 6 ล้านครั้งต่อวินาที รูปที่ 2.49 แสดงอุปกรณ์ที่ใช้ในการแปลงสัญญาณดังกล่าว



รูปที่ 2.49 แสดงอุปกรณ์ที่ใช้ในการแปลงสัญญาณ

2.6.10 สแกนเนอร์ (Scanners)

กระบวนการที่ใช้กับอุปกรณ์สแกนมีจำนวนมาก เพื่อที่จะให้ได้ภาพที่คมชัด เวลาที่ใช้ในการสแกนเร็ว อุปกรณ์สแกนนี้มีทั้งแบบที่เคลื่อนที่ และแบบที่อยู่กับที่ โดยที่ความละเอียดอย่างต่ำ 10 ดิพีไอ(จุดต่อนิ้ว) จนถึง 1000 ดิพีไอ บางชนิดจะมีตัวสแกนแบบเป็นแถวเพียงแถวเดียว เวลาจะสแกนจะทำการเคลื่อนแถวนั้น ๆ ไปตามบริเวณของรูปภาพ โดยปกติรูปภาพขนาด 2000X3000 จะใช้เวลาน้อยกว่า 20 วินาที

ปัญหาหลักของอุปกรณ์สแกนประกอบด้วย

1. สามารถสแกนได้ที่ละรูปเท่านั้น

2. บางกระบวนการแสแกนยังเชื่อถือไม่ได้
3. อุปกรณ์แสแกนที่ใช้มือเป็นตัวแสแกน การแสแกนจะขึ้นอยู่กับความดันและตำแหน่งของแสแกนที่เคลื่อนที่

2.6.10.1 อุปกรณ์จดจำตัวอักษร (Character Recognition Device)

อุปกรณ์เหล่านี้จะมีอยู่ในอุปกรณ์ที่ทำหน้าที่แสแกน โดยอาจเป็นฮาร์ดแวร์ หรือ ซอฟต์แวร์ เพื่อที่ทำการประมวลข้อความที่เป็นรูปภาพให้เป็นตัวอักษรที่คอมพิวเตอร์รู้จัก หรือรู้จักกันในชื่อของโอซีอาร์ โดยการประมวลผลนี้ผลลัพธ์ที่ได้จะขึ้นอยู่กับความคมชัดของส่วนอินพุต ถ้าส่วนอินพุตมีความคมชัดน้อย อาจทำให้ส่วนของเอาพุตผิดเพี้ยนได้

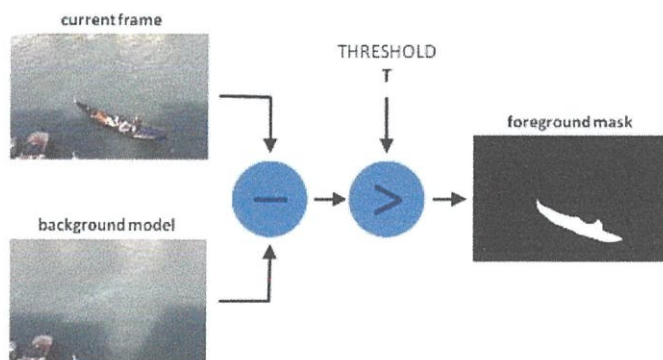
2.10 การลบภาพพื้นหลัง (Background Subtraction)

การลบภาพพื้นหลังคือการแยกวัตถุที่มีการเคลื่อนที่ออกจากฉากหลัง โดยการลบฉากหลัง ด้วยการนำภาพสองเฟรม ณ บริเวณเดียวกันในเวลาที่แตกต่างกัน มาทำการลบกัน หากมีการเคลื่อนที่เข้ามาของวัตถุในภาพเฟรมที่สอง เมื่อนำภาพทั้งสองเฟรมมาลบกัน ฉากหลังจะลบหักล้างกันระหว่างภาพสองเฟรม และมีส่วนผลต่างระหว่างภาพทั้งสองเฟรม ซึ่งผลต่างนี้เองคือการเคลื่อนไหของวัตถุที่เข้ามายังภาพเฟรมที่สอง เป็นไปดังสมการต่อไปนี้

$$O = I - B \quad (2.4)$$

เมื่อ O คือ วัตถุ(Object)
 I คือ ภาพของเฟรมหนึ่ง
 B คือ ภาพฉากหลัง

จากสมการข้างต้น เมื่อนำภาพทั้งสองเฟรมมาลบกันแล้ว ก็ต้องทำปรับค่าเทรชโซ ต่อเพื่อเป็นการแยกวัตถุที่เคลื่อนไหวในภาพ กับวัตถุที่ไม่ได้เคลื่อนไหว โดยค่าเทรชโซ นั้นก็แล้วแต่เราจะปรับค่าเอง ส่วนผลลัพธ์จะขึ้นอยู่กับค่าเทรชโซ ถ้าปรับให้มีค่ามากเกินไปจะทำให้มีน้อยช้เยาะ แต่ถ้าปรับไว้มีน้อยเกินไป จะทำให้ได้พื้นที่ของวัตถุที่กำลังเคลื่อนไหวมีน้อยเกินไปดังรูปที่ 2.50



รูปที่ 2.50 การการทำลบภาพพื้นหลังและการปรับเทรชโซ

2.11 แรสบेอร์รี่พาย(Raspberry Pi)

2.11.1. การกำเนิดแรสบेอร์รี่พาย

แรสบेอร์รี่พายถูกวางตลาดครั้งแรกในเดือนกุมภาพันธ์ ค.ศ.2012 โดยกำเนิดจากนักออกแบบคอมพิวเตอร์ชาวสหราชอาณาจักร อีเบน อัปตัน(Eben Upton) โดยมีแนวความคิดที่จะออกแบบและสร้าง แรสบेอร์รี่คือในขณะที่ทำงานของเขากับห้องปฏิบัติการคอมพิวเตอร์ของมหาวิทยาลัยแคมบริดจ์เกิดความไม่พอใจในการทำงานเพราะนักศึกษามีความสามารถในการเขียนโปรแกรมลดลงเมื่อเทียบกับนักศึกษาที่ผ่านมา อีกทั้งผู้คนทั่วไปก็คิดว่าคอมพิวเตอร์ก็คือ เครื่องมือสืบค้นข้อมูลทางอินเทอร์เน็ต พิมพ์เอกสาร หรือ คำนวณตารางเอกซ์เซล เท่านั้น ไม่ได้รู้ว่าคอมพิวเตอร์สามารถทำงานได้อีกหลากหลาย

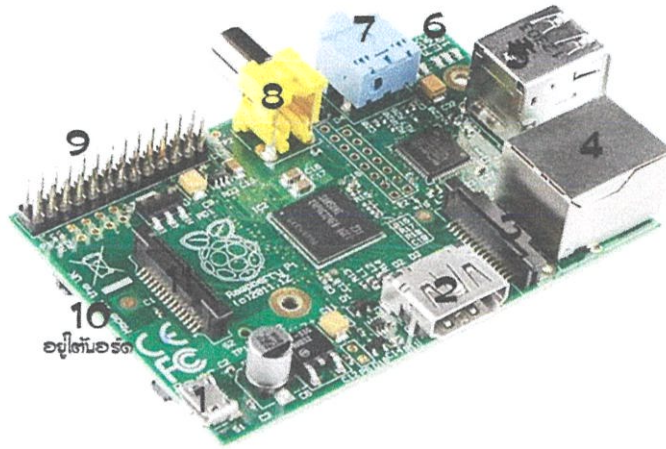
เมื่ออัปตันคิดอย่างนั้น จึงตั้งใจว่าจะต้องสร้างคอมพิวเตอร์ขนาดเล็ก และราคาถูกสุด ในตอนแรกได้ตั้งใจให้อยู่ที่25เหรียญสหรัฐ) และยังคงทำงานได้หลายอย่างแบบที่คอมพิวเตอร์สมัยนี้ทำได้อีก เขาคาดหวังว่าจะทำได้คนกลับมาเข้าใจการทำงานของคอมพิวเตอร์มากขึ้น สามารถเข้าถึงได้มากขึ้นเพราะมีราคาที่ถูกลง และฝึกเขียนโปรแกรมคอมพิวเตอร์กันมากขึ้น เขาได้คิดและลงมือทำอยู่ 6 ปี แรสบेอร์รี่พายโมเดลเอ(Raspberry Pi Model A) จึงได้เปิดตัวขึ้นและได้ให้เปิดจองกันในเดือนกุมภาพันธ์ ค.ศ.2012 และแล้วภายใน 3 เดือนก็มีคนสั่งจองกันราว แสน 5 และต เครื่อง้องรอนานกันเป็นเดือนกว่าจะได้ของมา และเมื่อผ่านไปเพียงปีเดียวก็สามารถขายได้เกิน 1 ล้านเครื่องทั่วโลก และในเดือนตุลาคม ค.ศ.2012 ได้ปล่อยแรสบेอร์รี่พายโมเดลบี(Raspberry Pi Model B) ออกมาขาย โดยปรับปรุงแรมเป็น 512 เมกะไบต์และเพิ่มพอร์ตแลนเข้าไปด้วย และยังมีการพัฒนาเพิ่มขึ้นอีก

เนื่องจากต้องการทำให้ราคาถูก นอกจากฮาร์ดแวร์ที่ใช้อย่างคุ้มค่าแล้ว ซอฟต์แวร์ก็จำเป็นต้องใช้แบบที่เป็นโอเพนซอร์ส(Open source) นั่นก็คือลินุกซ์(Linux) ดังนั้นระบบปฏิบัติการต่างๆ ของแรสบेอร์รี่พายจึงมีพื้นฐานมาจากลินุกซ์แล้วพัฒนาต่อให้มีอินเตอร์เฟซผู้ใช้งานที่คล้ายกับระบบปฏิบัติการวินโดวส์(Windows)

ความแตกต่างของแรสบेอร์รี่พายเมื่อเทียบกับอาดุยโน้นั้น ถ้าเปรียบเทียบกันในเรื่องของการใช้งานแล้ว แรสบेอร์รี่พายจะคล้ายกับเครื่องคอมพิวเตอร์ที่มีวินโดวส์หรือเครื่องแมค นั่นคือสามารถเปิดไปยังเว็บไซต์ต่างๆ สามารถลงโปรแกรมที่เป็นโอเพ่นออฟฟิศ เครื่องคิดเลข ดูหนัง ฟังเพลง ฝึกเขียนโปรแกรม และอื่นๆอีกเยอะ จึงสามารถกล่าวได้ว่าแรสบेอร์รี่พาย สามารถทำงานได้เช่นเดียวกับวินโดวส์เลยทีเดียว ซึ่งจะต่างจากอาดุยโน้ที่เป็นเพียงแค่มือใครคอนโทรลเลอร์เท่านั้น ไม่มีระบบปฏิบัติการเป็นของตัวเอง เขียนโปรแกรมแล้วสั่งให้ทำงานเป็นเรื่อยๆไปเท่านั้น

ถ้าต้องการให้แรสบेอร์รี่พายมาทำงานแบบอาดุยโน้ ก็สามารถทำได้ โดยเราสามารถเขียนโปรแกรมเพื่อควบคุมพอร์ตจีพีไอโอ(GPIO) โดยใช้ภาษาไพธอนหรือภาษาซีก็ได้ และยังมีความเร็วสูงกว่าอีกด้วย งานที่แรสบेอร์รี่พายจะถูกเลือกใช้เป็นส่วนมากนั้น จะที่เกี่ยวข้องกับกล้องถ่ายภาพนิ่ง วีดีโอ ประมวลผลภาพ เพราะงานในด้านนี้อาดุยโน้แบบทั่วไปจะประมวลผลภาพได้ไม่ทัน

2.11 2. จุดเชื่อมต่ออุปกรณ์และความสามารถของแต่ละตำแหน่งบนบอร์ดแรสเบอร์รี่พาย



รูปที่ 2.51 ส่วนประกอบของแรสเบอร์รี่พาย

หมายเลข 1 คือ ช่องเสียบไฟหรือช่องเสียบอะแดปเตอร์ เวลาต้องการจะจ่ายไฟให้กับบอร์ดแรสเบอร์รี่พาย เราจำเป็นต้องเสียบไฟจากแหล่งจ่ายเข้าตรงนี้และจากแหล่งจ่ายต้องมีแรงดันไฟ ไม่เกิน 3.3 โวลต์ ดังนั้นจำเป็นต้องหาอะแดปเตอร์ที่มีความเหมาะสมต่อการใช้งานด้วย

หมายเลข 2 คือช่องเสียบสายแอสดีเอ็มไอ(High Definition Multimedia Interface) ซึ่งมีไว้เพื่อเสียบ สายแอสดีเอ็มไอจากตัวบอร์ดแรสเบอร์รี่พายเพื่อเข้าสู่จอแสดงผลที่รองรับการเชื่อมต่อภาพด้วยแอสดีเอ็มไอเช่น จอคอมพิวเตอร์รุ่นใหม่ หรือสมาร์ททีวีในปัจจุบันก็รองรับการเชื่อมต่อเช่นกัน ซึ่งสามารถหาอุปกรณ์เหล่านี้และสายแอสดีเอ็มไอได้ง่ายตามร้านขายอุปกรณ์อิเล็กทรอนิกส์ทั่วไป

หมายเลข 3 คือ ซีเอสไอคอนเนคเตอร์(CSI Connector) ไว้สำหรับต่อกับโมดูลกล้องที่ออกแบบมาเพื่อบอร์ดแรสเบอร์รี่พายโดยเฉพาะ โดยลักษณะการต่อจะเป็นสายแพรที่มากับตัวกล้องต่อเข้าไปกับพอร์ชนี้และสามารถตั้งค่าการใช้งานกล้องได้อย่างสมบูรณ์แบบกว่าการนำกล้องแบบอื่นมาต่อเพราะกล้องที่นำมาต่อกับพอร์ตนี้ออกแบบมาเพื่อแรสเบอร์รี่พายโดยเฉพาะนั่นเอง

หมายเลข 4 คือ พอร์ตอีเทอร์เน็ตมีไว้สำหรับเชื่อมต่อบอร์ดแรสเบอร์รี่พาย เข้ากับระบบเครือข่ายคอมพิวเตอร์โดยผ่านสายแลน โดยการเชื่อมต่อนี้จะนำบอร์ดเข้าสู่ระบบเครือข่ายท้องถิ่น และสามารถออกอินเทอร์เน็ต ได้ด้วยทำให้เราสามารถอัปเดตซอฟต์แวร์ได้อิสระตามต้องการมากขึ้น และซึ่งเป็นปัจจัยหลักในการติดตั้งซอฟต์แวร์ต่างๆอีกด้วยเพราะเกือบจะ 100 % จะติดตั้งซอฟต์แวร์ผ่านระบบอินเทอร์เน็ต(ยกเว้น OS) เพิ่มเติมในส่วน of ระบบเครือข่าย บอร์ดแรสเบอร์รี่พายสามารถเชื่อมต่อกับระบบเครือข่ายไร้สาย(Wireless) ได้โดยการเพิ่มตัวรับไวไฟ(Wifi Dongle) เสียบเข้าทาง พอร์ตยูเอสบีซีสามารถทำงานได้เหมือนกันกับการเสียบสายและทุกประการแต่มีข้อเสียคือความเร็วต่ำกว่าการเสียบสายอยู่พอสมควร

หมายเลข 5 คือพอร์ตที่จำเป็นที่มีในทุกๆอุปกรณ์ในปัจจุบันนั้นก็คือพอร์ตยูเอสบีซีนั่นเอง เช่นเดียวกันกับพอร์ตยูเอสบีซีของแรสเบอร์รี่พาย นั้นก็สามารถเชื่อมต่ออุปกรณ์ต่างๆ ที่เป็นยูเอสบีซีได้หลากหลายมาก เช่น กล้องเว็บแคม, แม้าส์, คีย์บอร์ด,หรือแม้แต่ตัวรับไวไฟก็สามารถเชื่อมต่อและใช้งานได้เช่นกัน และยังมียูเอสบีซีอีกมากมายที่สามารถใช้งานเข้ากันได้

หมายเลข 6 คือสถานะไฟ เพื่อแสดงว่าสถานะของบอร์ด เช่น แสดงว่าเราเสียบไฟเข้าหรือไม่, สายแลนเชื่อมต่อหรือไม่ และมีไฟกระพริบที่บ่งบอกว่าบอร์ดมีการส่งข้อมูลกันกับเครือข่ายนั่นเอง

หมายเลข 7 คือ ช่องเสียบแจ็คเสียงเข้าพุท หรือ ก็คือช่องเสียบลำโพงนั่นเองแน่นอนว่าแอสเบอร์รี่พายสามารถมีเสียงและเปิดเพลงหรือมีเดีย ที่มีเสียงได้ในซอฟต์แวร์ของบอร์ดแอสเบอร์รี่พายนั่นเอง

หมายเลข 8 คือ ช่องต่อสัญญาณภาพเป็นชนิด แจ็คอาร์ซีเอ แบบธรรมดาสามารถต่อเข้ากับทีวีทั่วไปได้โดยประโยชน์ของมันคือเวลาเราเริ่มต้นใช้งานสามารถต่อสัญญาณภาพออกจากตรงนี้

หมายเลข 9 คือ จีพีไอโอ(General Purpose Input Output/GPIO)ส่วนที่เห็นเป็นเสาแหลมๆ สามารถต่อสายเพิ่มเข้าไปได้ เพราะจีพีไอโอ ซึ่งคือพอร์ตเชื่อมต่อที่เราสามารถกำหนดให้มันเป็นอินพุต หรือ เอาท์พุตได้โดยการเขียนโปรแกรมขึ้นเองหรือใช้ซอฟต์แวร์ที่มีมาให้ดาวน์โหลดใช้ เช่นไวร์ริงพาย เป็นต้น ทั้งนี้จีพีไอโอ สามารถทำงานได้หลายอย่างสามารถที่จะเชื่อมต่อเซ็นเซอร์ต่างๆ ที่เราต้องการแล้วป้อนค่าให้กับโปรแกรมตามที่เราต้องการได้เลย ขึ้นอยู่กับการเขียนโปรแกรม เช่น สามารถใช้จีพีไอโอแล้วเขียนโปรแกรมให้ส่งค่าลอจิก1 ออกไปนั่นหมายถึงว่าให้ ขาที่ถูกกำหนดมีแรงดันไฟส่งออกไป ให้กับบริเลย์เพื่อทำการเปิดไฟห้องให้สว่างได้ ทั้งนี้สามารถที่จะเขียนยูเซอร์อินเตอร์เฟซของมันออกมาเป็นหน้าเว็บเพจเพื่อการใช้งานที่สะดวกได้อีก

หมายเลข 10 (อยู่ด้านล่างบอร์ด) คือ ช่องเสียบ เมมโมรี่การ์ด ซึ่งเป็นหน่วยความจำที่เก็บ ระบบปฏิบัติการ และ ซอฟต์แวร์ทั้งหมดที่เราติดตั้งลงไปไว้

หมายเลข 11 คือ ดีเอสไอดีสเพลย์(DSI Display) ทำหน้าที่ไว้ต่อกับจอชนิดที่เป็นสายแพร

2.11.3 ข้อดีและข้อเสียของแอสเบอร์รี่พาย

ข้อดีคือ

- มีความเล็กกระทัดรัด เคลื่อนย้ายสะดวก
- ประหยัดพลังงาน
- ใช้ซอฟต์แวร์ที่เป็นโอเพ่นซอร์สทั้งหมด
- เป็นคอมพิวเตอร์ขนาดจิ๋วที่เหมาะสมกับการเริ่มต้นเขียนโปรแกรม
- สามารถทำเป็นเซิร์ฟเวอร์เพื่อรองรับการร้องขอจากเครื่องไคลเอนท์ได้
- มีจีพีไอโอ สามารถสามารถเชื่อมต่อกัน เซ็นเซอร์ต่างๆหรือบอร์ดคอนโทรลอื่นๆ ได้อย่างอิสระ
- มีทั้ง หน้าจอ อินเตอร์เฟซผู้ใช้งาน และ ช่องสั่งการ(Terminal)
- สามารถเชื่อมต่อผ่านซีเคียวเชลล์(Secure Shell) เข้าไปสั่งงานได้โดยใช้ โปรแกรมพุตตี(puTTY)เป็นต้น
- ราคาประหยัดและสามารถหาอุปกรณ์เสริมอื่นๆได้ง่าย

ข้อเสีย

- มีความเร็วค่อนข้างต่ำ แต่เมื่อใช้งานในโหมดสั่งการผ่านคอมพิวเตอร์จะค่อนข้างเร็ว
- ราคาคือเป็นระบบปฏิบัติการลินุกซ์ซึ่งอาจจะยากแก่ผู้ที่ไม่คุ้นเคย
- การตั้งค่าบางอย่างจำเป็นต้องใช้คำสั่งคอมมานด์ไลน์ (ซึ่งจะขึ้นอยู่กับความถนัดแต่ละคน)

2.11 4. พอร์ตจีพีไอโอของแรสเบอร์รี่พาย (Raspberry Pi GPIO)

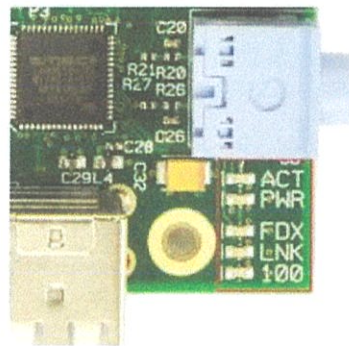
พอร์ตจีพีไอโอ ซึ่งในโมเดลเอและบี(Revision 1) ทุกพินจะเหมือนกัน แต่โมเดลบี(Revision 2) จะแตกต่างกันรายละเอียดดังรูป

Raspberry Pi Model A & B (Revision 1)				Raspberry Pi Model B (Revision 2)			
3.3V	1	2	3V	3.3V	1	2	3V
I2C0 SDA	3	4	DNC	I2C1 SDA	3	4	5V
I2C0 SCL	5	6	GROUND	I2C1 SCL	5	6	GROUND
GPIO4	7	8	UART TXD	GPIO4	7	8	UART TXD
DNC	9	10	UART RXD	GROUND	10	11	UART RXD
GPIO 17	11	12	GPIO 18	GPIO 17	11	12	GPIO 18
GPIO 21	13	14	DNC	GPIO 27	13	14	GROUND
GPIO 22	15	16	GPIO 23	GPIO 22	15	16	GPIO 23
DNC	17	18	GPIO 24	3.3V	17	18	GPIO 24
SP10 MOSI	19	20	DNC	SP10 MOSI	19	20	GROUND
SP10 MISO	21	22	GPIO 25	SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N	SP10 SCLK	23	24	SP10 CE0 N
DNC	25	26	SP10 CE1 N	GROUND	25	26	SP10 CE1 N

แหล่งที่มา: <http://www.hobbytronics.co.uk/raspberry-pi-gpio-pins/>

รูปที่ 2.52 พอร์ตจีพีไอโอของแรสเบอร์รี่พาย

แอลอีดีแสดงสถานะของบอร์ด อยู่ในบริเวณกรอบสีแดง ดังภาพ



รูปที่ 2.53 แอลอีดีแสดงสถานะของแรสเบอร์รี่พาย

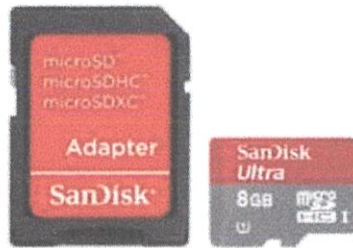
- ACT คือ ไฟสถานะเมมโมรี่การ์ดพร้อมใช้งาน (สีเขียว)
- PWR คือ ไฟสถานะ 3.3 โวลต์ (สีแดง)
- FDX คือ ไฟสถานะ ฟูลดัพเพลิกซ์ แลน โมเดลบี(สีเขียว)
- LNK คือ ไฟสถานะ ลิงก์/แอคทีวิตี แลน โมเดลบี (สีเขียว)
- 100 คือ ไฟสถานะ 10/100เมกะไบต์ต่อวินาที แลน โมเดลบี (สีเหลือง)

2.11 5. ติดตั้งระบบปฏิบัติการ

ก่อนเริ่มต้นการใช้งานบอร์ดแรสเบอร์รี่พาย จำเป็นที่จะต้องติดตั้งระบบปฏิบัติการให้กับบอร์ดก่อนเนื่องจากบอร์ดไม่มีหน่วยความจำแบบแฟลชเมมโมรี่มาบนบอร์ดด้วย ดังนั้น จำเป็นที่จะต้องเตรียมอุปกรณ์ต่างๆ ให้พร้อม เพื่อให้สามารถใช้งานบอร์ดได้ ซึ่งมีรายละเอียดอุปกรณ์ดังนี้

1. บอร์ดแรสเบอร์รี่พาย

2. เม็มโมรี่การ์ดสำหรับติดตั้งระบบปฏิบัติการลินุกซ์ ต้องมีความจุมากกว่า 2 จิกกะไบต์ ขึ้นไป แต่แนะนำให้ใช้ ขนาด 4 จิกกะไบต์ หรือมากกว่า สำหรับคู่มือฉบับนี้จะใช้ขนาด 8 จิกกะไบต์ควรเลือกใช้การ์ดที่มีความเร็วสูงอย่าง คลาส 10 เพื่อประสิทธิภาพการทำงานของระบบโดยรวม



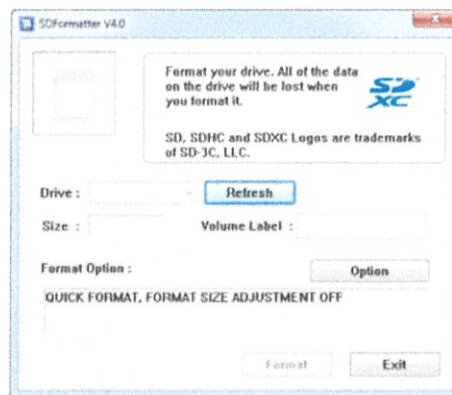
รูปที่ 2.54 ตัวอย่างเม็มโมรี่การ์ดที่ใช้ติดตั้งระบบปฏิบัติการ

3. เม้าส์และคีย์บอร์ดแบบยูเอสบี
4. สายไมโครยูเอสบีเพื่อจ่ายไฟเลี้ยงวงจร สามารถเลือกใช้แหล่งจ่ายไฟจากพอร์ตยูเอสบีของเครื่องคอมพิวเตอร์ได้
5. สายเชื่อมต่อสัญญาณภาพและเสียง(HDMI Cable) เพื่อเชื่อมต่อกับจอแสดงผล หากเลือกใช้จอมอนิเตอร์ที่ไม่มีพอร์ตเชื่อมต่อสัญญาณภาพและเสียง

การเตรียมซอฟต์แวร์สำหรับติดตั้งระบบปฏิบัติการลินุกซ์ ลงบนบอร์ดแรสเบอร์รี่พาย

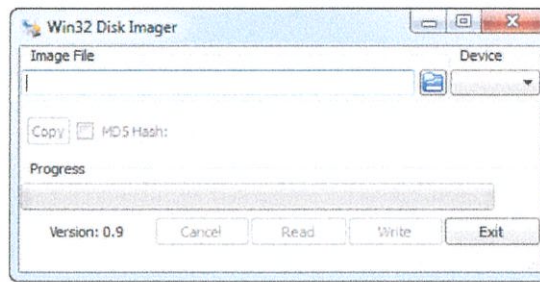
คู่มือฉบับนี้จะจัดเตรียมซอฟต์แวร์ที่รองรับระบบปฏิบัติการวินโดวเป็นหลัก และต้องติดตั้งลงบนเครื่องคอมพิวเตอร์ดังนี้

1. โปรแกรม SD Formatter V4.0 ใช้สำหรับฟอร์แมตเม็มโมรี่การ์ดสามารถดาวน์โหลดได้จากลิงค์



รูปที่ 2.55 โปรแกรม SD Formatter V4.0

- 2.โปรแกรม Win32 DiskImager ใช้สำหรับเขียนไฟล์ระบบปฏิบัติการที่เป็นไฟล์ Image (*.img) ลงบนเม็มโมรี่การ์ดสามารถดาวน์โหลดได้จากลิงค์

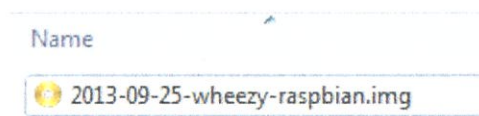


รูปที่ 2.56 โปรแกรม Win32 Disk Imager

.3ไฟล์ระบบปฏิบัติการ คู่มือนี้ติดตั้งระบบปฏิบัติการแรสเบียน(Raspbian) เป็นระบบปฏิบัติการเดเบียน วิชาซี(Debian Wheezy) ที่ถูกปรับแต่งให้ใช้สำหรับบอร์ดแรสเบอร์รี่พายโดยเฉพาะ เป็นลินุกซ์ที่ให้ใช้งานได้ฟรี สามารถดาวน์โหลดได้จากลิงค์

ขั้นตอนการติดตั้งระบบปฏิบัติการ Raspbian ให้กับบอร์ด Raspberry Pi

1. หากมีข้อมูลอยู่ในเมมโมรี่การ์ดให้ทำการฟอร์แมตด้วยโปรแกรม SD Formatter 4.0 หรือโปรแกรมอื่นๆ ก็ได้ ถ้าหาก ฟอร์แมต แล้วให้ข้ามขั้นตอนนี้ได้เลย
2. เมื่อดาวน์โหลดไฟล์ระบบปฏิบัติการแรสเบียนมาแล้วจะได้เป็นไฟล์ซิป(.zip) ให้แตกไฟล์จะได้เป็นไฟล์อิมเมจ(Image (*.img)) มาแสดงดังรูป



รูปที่ 2.57 การแตกไฟล์ซิปของไฟล์ระบบปฏิบัติการแรสเบียน

การตั้งค่าไอพีแอดเดรสบนแรสเบอร์รี่พาย

ไอพีแอดเดรสจำเป็นในการเชื่อมต่อกับคอมพิวเตอร์เพื่อการถ่ายโอนข้อมูลจากคอมพิวเตอร์มายังแรสเบอร์รี่พาย หรือจากแรสเบอร์รี่พายไปยังคอมพิวเตอร์ โดยการกำหนดค่าเกี่ยวกับเน็ตเวิร์กนั้นใช้

ifconfig

```
File Edit Tabs Help
pi@raspberrypi ~ $ ifconfig
eth0    Link encap:Ethernet  HWaddr b8:27:eb:07:f8:4d
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:681 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:71406 (69.7 KiB)  TX bytes:0 (0.0 B)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:392 errors:0 dropped:0 overruns:0 frame:0
        TX packets:392 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:32208 (31.4 KiB)  TX bytes:32208 (31.4 KiB)

wlan0   Link encap:Ethernet  HWaddr e8:4e:06:1c:20:24
        inet addr:192.168.137.74  Bcast:192.168.137.255  Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:1130 errors:0 dropped:5252 overruns:0 frame:0
        TX packets:409 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:294162 (287.2 KiB)  TX bytes:50821 (49.6 KiB)
```

รูปที่ 2.58 การแสดงผลลักษณะการเชื่อมต่อเครือข่าย

โดยที่eth0, lo, wlan0 เป็นชื่อของลักษณะการเชื่อมต่อเครือข่ายที่ใช้งานภายในระบบ

-eth0 มักจะแสดงถึงการเชื่อมต่ออินเทอร์เน็ตกับเครือข่ายผ่านสายเคเบิล ซึ่งรวมถึงสายแลนด้วย

-lo เป็นการเชื่อมต่อพิเศษที่ระบบสามารถสื่อสารกับตัวเองได้

-wlan0 เป็นชื่อการเชื่อมต่อเครือข่ายไร้สายอันแรกในระบบ

ในการตั้งค่าไอพีแอดเดรสของแรสเบอร์รี่พายเพื่อการเชื่อมต่อแลกเปลี่ยนข้อมูลกับคอมพิวเตอร์ผ่านสายแลนนั้นจึงต้องตั้งค่าในส่วนของอินเทอร์เน็ต โดยพิมพ์คำสั่งบนคอมพิวเตอร์ดังนี้

```
sudo ifconfig eth0 192.168.1.2
```

เมื่อพิมพ์คำสั่งนี้ลงในคอมพิวเตอร์แล้ว ค่าไอพีแอดเดรสจะถูกเปลี่ยนไป ซึ่งแสดงอยู่ในรูปที่ 2.18 ในผลลัพธ์บรรทัดที่ 2 ของอินเทอร์เน็ต จะพบดังนี้

```
inet addr:192.168.1.2
```

แสดงให้เห็นว่า ค่าไอพีจะถูกตั้งค่าเป็น 192.168.1.2 แล้วเพื่อสามารถเชื่อมต่อกับคอมพิวเตอร์ผ่านสายแลนได้ สร้างความสะดวกในการสื่อสารกันระหว่างแรสเบอร์รี่พายและคอมพิวเตอร์

```
File Edit Tabs Help
pi@raspberrypi ~ $ sudo ifconfig eth0 192.168.1.2
pi@raspberrypi ~ $ ifconfig
eth0    Link encap:Ethernet  HWaddr b8:27:eb:07:f8:4d
        inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:681 errors:0 dropped:0 overruns:0 frame:0
        TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:71406 (69.7 KiB)  TX bytes:2732 (2.6 KiB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:392 errors:0 dropped:0 overruns:0 frame:0
        TX packets:392 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:32208 (31.4 KiB)  TX bytes:32208 (31.4 KiB)

wlan0   Link encap:Ethernet  HWaddr e8:4e:06:1c:20:24
        inet addr:192.168.137.74  Bcast:192.168.137.255  Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:1141 errors:0 dropped:5252 overruns:0 frame:0
        TX packets:409 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:296985 (290.0 KiB)  TX bytes:50821 (49.6 KiB)
```

รูปที่ 2.59 แสดงการตั้งค่าไอพีแอดเดรสบนแรสเบอร์รี่พาย

2.12 โปรแกรมคอมพิวเตอร์

2.12.1 การเขียนโปรแกรมคอมพิวเตอร์ (Computer programming)

การเขียนโปรแกรมคอมพิวเตอร์หรือเรียกให้สั้นลงว่า การเขียนโปรแกรม(Programming) หรือ การเขียนโค้ด (Coding) เป็นขั้นตอนการเขียน ทดสอบ และดูแลซอร์สโค้ดของโปรแกรมคอมพิวเตอร์ ซึ่งซอร์สโค้ดนั้นจะเขียนด้วยภาษาโปรแกรม ขั้นตอนการเขียนโปรแกรมต้องการความรู้ในหลายด้านด้วยกัน เกี่ยวกับโปรแกรมที่ต้องการจะเขียน และขั้นตอนวิธีที่จะใช้ ซึ่งในวิศวกรรมซอฟต์แวร์นั้น การเขียนโปรแกรมถือเป็นเพียงขั้นหนึ่งในวงจรชีวิตของการพัฒนาซอฟต์แวร์

การเขียนโปรแกรมจะได้มาซึ่งซอร์สโค้ดของโปรแกรมนั้นๆ โดยปกติแล้วจะอยู่ในรูปแบบของ ข้อความธรรมดา ซึ่งไม่สามารถนำไปใช้งานได้ จะต้องผ่านการคอมไพล์ตัวซอร์สโค้ดนั้นให้เป็นภาษาเครื่อง(Machine Language) เสียก่อนจึงจะได้เป็นโปรแกรมที่พร้อมใช้งาน

ภาษาโปรแกรมแต่ละภาษาจะมีลักษณะหรือรูปแบบการเขียนที่แตกต่างกัน การเลือกภาษาโปรแกรมหรือภาษาคอมพิวเตอร์เพื่อนำมาเขียนโปรแกรมนั้นขึ้นอยู่กับปัจจัยหลาย ๆ อย่าง เช่น นโยบายของบริษัท, ความเหมาะสมของโปรแกรมกับลักษณะงานที่จะถูกนำไปใช้, การเข้ากันได้กับโปรแกรมอื่น ๆ, หรืออาจเป็นความถนัดของแต่ละคน ภาษาโปรแกรมที่มีแนวโน้มในการนำมาเขียนมักเป็นภาษาที่มีคนที่สามารถเขียนได้ทันที หรือหากมีความจำเป็นที่จะต้องเลือกใช้ภาษาอื่น เช่นต้องการเน้นประสิทธิภาพในการทำงานของโปรแกรม ก็อาจจำเป็นต้องหานักเขียนโปรแกรมขึ้นมาจำนวนหนึ่งซึ่งมีความรู้ความเข้าใจในภาษาโปรแกรมที่ต้องการ และต้องมีคอมไพเลอร์ที่รองรับภาษาเหล่านั้นด้วย

2.12.2 ภาษาที่ใช้เขียนโปรแกรม

2.12.2.1 ภาษาเครื่อง (Machine Languages)

ภาษาเครื่อง คือ กลุ่มของคำสั่งเครื่องที่กระทำการโดยตรงโดยหน่วยประมวลผลกลาง(CPU) ของคอมพิวเตอร์ คำสั่งเครื่องแต่ละคำสั่งจะปฏิบัติงานเฉพาะกิจงานเดียวเท่านั้น เช่น การบรรจุ, การกระโดด หรือการดำเนินการผ่านหน่วยคำนวณและตรรกะ(ALU) บนหน่วยของข้อมูลในหน่วยความจำหรือเรจิสเตอร์ ทุกๆโปรแกรมที่กระทำการโดยหน่วยประมวลผลกลางสร้างขึ้นจากอนุกรมของคำสั่งเครื่องเช่นว่านั้น

รหัสเครื่องเชิงตัวเลข(ซึ่งไม่ใช่รหัสแอสเซมบลี) อาจพิจารณาได้ว่าเป็นตัวแทนระดับต่ำสุดของโปรแกรมคอมพิวเตอร์ที่ได้คอมไพล์และ/หรือเขียนด้วยภาษาแอสเซมบลี หรือเป็นภาษาโปรแกรมแบบดั้งเดิมและขึ้นอยู่กับฮาร์ดแวร์ ถึงแม้ว่าเราจะสามารถเขียนโปรแกรมด้วยรหัสเครื่องเชิงตัวเลขโดยตรงก็ได้ แต่การจัดการบิดต่าง ๆ เป็นเอกเทศ และการคำนวณตำแหน่งที่อยู่กับค่าคงตัวเชิงตัวเลขด้วยมือ จะทำให้น่าเบื่อหน่ายและมีแนวโน้มที่จะเกิดความผิดพลาด ดังนั้นการเขียนรหัสเครื่องจึงไม่ค่อยกระทำกันในทุกวันนี้ เว้นแต่ในสถานการณ์ที่ต้องการทำให้เหมาะสมอย่างที่สุดหรือแก้จุดบกพร่อง

ปัจจุบันนี้โปรแกรมเกือบทั้งหมดในทางปฏิบัติเขียนขึ้นด้วยภาษาแอสเซมบลีหรือภาษาระดับสูงกว่า แล้วแปลเป็นรหัสเครื่องที่กระทำการได้โดยคอมไพเลอร์และ/หรือแอสเซมเบลอร์ กับลิงเกอร์ อย่างไรก็ตาม โปรแกรมนั้นที่เขียนด้วยภาษาที่แปลด้วยอินเทอร์พรีเตอร์จะไม่ถูกแปลเป็นรหัสเครื่อง ถึงแม้ว่าอินเทอร์พรีเตอร์ (ตัวกระทำการ หรือ ตัวประมวลผล) โดยทั่วไปประกอบขึ้นจากรหัสเครื่องที่กระทำการได้โดยตรง

หน่วยประมวลผลหรือตระกูลของหน่วยประมวลผลทุก ๆ ชนิดมีชุดของคำสั่งเครื่องที่เป็นรหัสของมันเป็นเอง คำสั่งเครื่องคือแบบรูปต่าง ๆ ของบิต ซึ่งการออกแบบเชิงกายภาพสอดคล้องกับคำสั่งงานที่แตกต่างกันของเครื่อง ดังนั้นชุดของคำสั่งเครื่องจึงใช้ได้กับประเภทของหน่วยประมวลผลที่ใช้สถาปัตยกรรมเดียวกัน การออกแบบหน่วยประมวลผลรุ่นหลังหรือรุ่นต่อๆ ไปก็มักจะรวมคำสั่งทั้งหมดของรุ่นก่อนหน้าไว้ และอาจเพิ่มเติมคำสั่งใหม่เข้าไปอีกด้วย หน่วยประมวลผลรุ่นหลังอาจยกเลิกหรือเปลี่ยนแปลงรหัสของคำสั่งเครื่องในบางครั้ง (สาเหตุทั่วไปก็คือมันจำเป็นสำหรับจุดประสงค์ใหม่) ส่งผลกระทบต่อความเข้ากันได้ของรหัสในบางขอบข่าย แม้กระทั่งหน่วยประมวลผลที่เข้ากันได้เกือบสมบูรณ์ก็อาจแสดงพฤติกรรมต่างไปจากเดิมเล็กน้อยสำหรับบางคำสั่ง แต่ปัญหานี้พบได้น้อยมาก ระบบต่าง ๆ ก็อาจแตกต่างกันในรายละเอียดอื่น เช่น การจัดการหน่วยความจำ ระบบปฏิบัติการ หรืออุปกรณ์รอบข้าง เนื่องจากตามปกติแล้วโปรแกรมจะยึดถือปัจจัยดังกล่าว ระบบที่แตกต่างกันก็จะไม่ทำงานด้วยรหัสเครื่องที่เหมือนกัน ถึงแม้ว่าใช้หน่วยประมวลผลชนิดเดียวกันก็ตาม

รหัสของชุดของคำสั่งเครื่องอาจมีความยาวเท่ากันหมดทุกคำสั่งหรือมีความยาวแปรผันก็ได้ วิธีการจัดการแบบรูปของรหัสเครื่องขึ้นอยู่กับสถาปัตยกรรมนั้น ๆ เป็นอย่างยิ่ง และมักจะขึ้นอยู่กับชนิดของคำสั่ง คำสั่งเครื่องส่วนมากมีฟิลด์ออปโค้ด(Opcode) หนึ่งฟิลด์หรือมากกว่าซึ่งใช้ระบุชนิดของคำสั่งพื้นฐาน (เช่น เลขคณิต ตรรกศาสตร์ การกระโดด ฯลฯ) และการดำเนินการแท้จริง (เช่น การบวก การเปรียบเทียบ) และมีฟิลด์อื่น ๆ ที่อาจใช้สำหรับระบุชนิดของตัวถูกดำเนินการ(Operand) ภาวะการกำหนดตำแหน่งที่อยู่(Addressing Mode) ออฟเซตของตำแหน่งที่อยู่ หรือดัชนี หรือค่าแท้จริงโดยตัวมันเอง (ตัวถูกดำเนินการที่เป็นค่าคงตัวที่บรรจุอยู่ในคำสั่งเครื่องเช่นนั้นเรียกว่า ค่าใช้ทันที, Immediate)

ไม่ใช่ว่าทุกเครื่องหรือทุกคำสั่งจะมีตัวถูกดำเนินการชัดเจน เครื่องที่ใช้ตัวสะสม(Accumulator Machine) มีตัวถูกดำเนินการข้างซ้ายแบบผสม และคืนค่าผลลัพธ์ในตัวสะสม(Accumulator) ปริยาย สำหรับคำสั่งเลขคณิตส่วนใหญ่ สถาปัตยกรรมอื่น(เช่น 8086 และตระกูล x86) มีคำสั่งธรรมดาในรุ่นของตัวสะสมอยู่ด้วย ซึ่งคำสั่งที่ยาวกว่าจะทำเรจิสเตอร์อื่นหนึ่งในเรจิสเตอร์ทั่วไปเป็นตัวสะสม เครื่องที่ใช้กองซ้อน(Stack Machine) มีตัวถูกดำเนินการส่วนใหญ่หรือทั้งหมดอยู่บนกองซ้อนปริยาย คำสั่งเครื่องที่มีจุดประสงค์พิเศษก็มักจะขาดตัวถูกดำเนินการชัดเจน (ตัวอย่างเช่น ซีพียูไอดีในสถาปัตยกรรม x86 เขียนค่าต่าง ๆ ลงในเรจิสเตอร์ปลายทางเป็นปริยาย 4 เรจิสเตอร์) ความแตกต่างระหว่างตัวถูกดำเนินการชัดเจนกับปริยายเช่นนี้สำคัญต่อโปรแกรมสร้างรหัสเครื่อง โดยเฉพาะอย่างยิ่งในเรื่องการจัดสรรเรจิสเตอร์และส่วนติดตามพิสัยแบบสด โปรแกรมทำรหัสให้เหมาะที่สุด(Code Optimizer) ที่ดีสามารถติดตามตัวถูกดำเนินการทั้งชัดเจนและปริยาย ซึ่งอาจช่วยให้เกิดการแพร่กระจายค่าคงตัว(Constant Propagation) ได้บ่อยยิ่งขึ้น การพับทบค่าคงตัว(Constant Folding) ของเรจิสเตอร์ (เรจิสเตอร์ที่กำหนดให้เป็นผลลัพธ์จากนิพจน์ของค่าคงตัวจะถูกแทนที่ด้วยค่าคงตัวนั้น) และการปรับปรุงรหัสอื่น ๆ ให้ดีขึ้น

2.12.2.2 ภาษาแอสเซมบลี (Assembly)

ภาษาแอสเซมบลี(Assembly Language) หมายถึง ภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่งซึ่งจะทำงานโดยขึ้นกับรุ่นของไมโครโพรเซสเซอร์ หรือ "หน่วยประมวลผล" (CPU) ของเครื่องคอมพิวเตอร์

การใช้ภาษาแอสเซมบลีจำเป็นต้องผ่านการแปลภาษาด้วยคอมไพเลอร์เฉพาะเรียกว่า แอสเซมเบลอร์(Assembler) ให้อยู่ในรูปของรหัสคำสั่งก่อน(เช่น .OBJ) โดยปกติ ภาษานี้ค่อนข้างมีความยุ่งยากในการใช้งาน และการเขียนโปรแกรมเป็นจำนวนบรรทัดมากกว่า เมื่อเปรียบเทียบกับการใช้ภาษาระดับสูง เช่น ภาษาซีหรือภาษาเบสิก แต่จะทำให้ได้ผลลัพธ์การทำงานของโปรแกรมเร็วกว่า และขนาดของตัวโปรแกรมมีขนาดเนื้อที่น้อยกว่าโปรแกรมที่สร้างจากภาษาอื่นมาก จึงนิยมใช้ภาษานี้เมื่อต้องการประหยัดเวลาทำงานของเครื่องคอมพิวเตอร์ และเพิ่มประสิทธิภาพของโปรแกรม

เนื่องจากตัวคำสั่งภายในภาษาอ้างอิงเฉพาะกับรุ่นของหน่วยประมวลผล ดังนั้นถ้ามีการเปลี่ยนแปลงไปใช้กับหน่วยประมวลผลอื่นหรือระบบอื่น (เช่น หน่วยประมวลผล x86 ไม่เหมือนกับ z80) จะต้องมีการปรับแก้ตัวคำสั่งภายในซึ่งบางครั้งอาจไม่สามารถปรับปรุงแก้ไขได้อย่างสมบูรณ์

2.12.2.3 ภาษาระดับสูง (High-level Languages)

ภาษาโปรแกรมระดับสูง หมายถึงภาษาโปรแกรมที่มีภาวะนามธรรมอย่างสูงจากรายละเอียดการทำงานของคอมพิวเตอร์ หากเปรียบเทียบกับภาษาโปรแกรมระดับต่ำแล้ว ภาษาโปรแกรมระดับสูงอาจมีองค์ประกอบเป็นภาษาธรรมชาติ ใช้งานง่ายกว่า ทำให้กระบวนการพัฒนาโปรแกรมตามข้อกำหนดเรียบง่ายกว่าและสามารถทำความเข้าใจได้ดีกว่า ระดับของภาวะนามธรรมที่ภาษาโปรแกรมจัดเตรียมไว้ให้ เป็นตัวกำหนดว่าภาษานั้นมี "ระดับสูง" มากน้อยแค่ไหน

ประเภทของภาษาโปรแกรมระดับสูงที่ใช้กัน มีดังนี้

- ภาษาซี (C)

ภาษาซี (C) เป็นภาษาโปรแกรมสำหรับวัตถุประสงค์ทั่วไป เริ่มพัฒนาขึ้นช่วง พ.ศ. 2512-2516 (ค.ศ. 1969-1973) โดยเดนนิส ริชชี (Denis Retchie) ที่เอทีแอนด์ทีเบลล์แล็บส์(AT&T Bell Labs) ภาษาซีเป็นภาษาที่มีความยืดหยุ่นในการเขียนโปรแกรมและมีเครื่องมืออำนวยความสะดวกสำหรับการเขียนโปรแกรมเชิงโครงสร้างและอนุญาตให้มีขอบข่ายตัวแปร(Scope) และการเรียกซ้ำ(Recursion) ในขณะที่ระบบชนิดตัวแปรอพลวัตก็ช่วยป้องกันการดำเนินการที่ไม่ตั้งใจหลายอย่าง เหมือนกับภาษาโปรแกรมเชิงคำสั่งส่วนใหญ่ในแบบแผนของภาษาอัลกอล การออกแบบของภาษาซีมีคอนสตรัคต์(Construct) ที่โยงกับชุดคำสั่งเครื่องทั่วไปได้อย่างพอเพียง จึงทำให้ยังมีการใช้ในโปรแกรมประยุกต์ซึ่งแต่ก่อนลงรหัสเป็นภาษาแอสเซมบลี คือซอฟต์แวร์ระบบอันโดดเด่นอย่างระบบปฏิบัติการคอมพิวเตอร์ ยูนิกซ์

ภาษาซีเป็นภาษาโปรแกรมหนึ่งที่ใช้กันอย่างแพร่หลายมากที่สุดตลอดกาล และตัวแปลโปรแกรมของภาษาซีมีให้ใช้งานได้สำหรับสถาปัตยกรรมคอมพิวเตอร์และระบบปฏิบัติการต่าง ๆ เป็นส่วนมาก

ภาษาหลายภาษาในยุคหลังได้หยิบยืมภาษาซีไปใช้ทั้งทางตรงและทางอ้อม ตัวอย่างเช่น ภาษาดี ภาษาโก ภาษาอาร์สดี ภาษาจาวา จาวาสคริปต์ภาษาลิมโบ ภาษาแอลพีซี ภาษาซีชาร์ป ภาษาอ็อบเจกทีฟ-ซี ภาษาเพิร์ล ภาษาพีเอชพี ภาษาไพทอน ภาษาเวอริล็อก(ภาษาพรรณนาฮาร์ดแวร์) และซีเชลล์ของยูนิกซ์ ภาษาเหล่านี้ได้ตั้งโครงสร้างการควบคุมและคุณลักษณะพื้นฐานอื่น ๆ มาจากภาษาซี ส่วนใหญ่มีวากยสัมพันธ์คล้ายคลึงกับภาษาซีเป็นอย่างมากโดยรวม (ยกเว้นภาษาไพทอนที่ต่างออกไปอย่างสิ้นเชิง) และตั้งใจที่จะผสมผสานนิพจน์และข้อความสั่งที่จำแนกได้ของวากยสัมพันธ์ของภาษาซี ด้วยระบบชนิดตัวแปร ตัวแบบข้อมูล และอรรถศาสตร์ที่อาจแตกต่างกันโดยมูล

ฐาน ภาษาซีพลัสพลัสและภาษาอ็อบเจกทีฟ-ซีเดิมเกิดขึ้นในฐานะตัวแปลโปรแกรมที่สร้างรหัสภาษาซี ปัจจุบัน ภาษาซีพลัสพลัสแทบจะเป็นเซตใหญ่ของภาษาซี ในขณะที่ภาษาอ็อบเจกทีฟ-ซีก็เป็นเซตใหญ่อันเคร่งครัดของภาษาซี

ก่อนที่จะมีมาตรฐานภาษาซีอย่างเป็นทางการ ผู้ใช้และผู้พัฒนาต่างก็เชื่อถือในข้อกำหนดอย่างไม่เป็นทางการ ในหนังสือที่เขียนโดยเคนนิส ริตซี และไบรอัน เคอร์นิกัน (Brian Kernighan) ภาษาซีรุ่นนั้นจึงเรียกกันโดยทั่วไปว่า ภาษาเคแอนดอร์ซี(K&R C) ต่อมา พ.ศ. 2532 สถาบันมาตรฐานแห่งชาติของสหรัฐอเมริกา(ANSI) ได้ตีพิมพ์มาตรฐานสำหรับภาษาซีขึ้นมา เรียกกันว่า ภาษาแอนซีซี(ANSI C) หรือ ภาษาซี89(C89) ในปีถัดมา องค์การระหว่างประเทศว่าด้วยการมาตรฐาน(ISO) ได้อนุมัติให้ข้อกำหนดเดียวกันนี้เป็นมาตรฐานสากล เรียกกันว่า ภาษาซี90(C90) ในเวลาต่อมาอีก องค์การฯ ก็ได้เผยแพร่ส่วนขยายมาตรฐานเพื่อรองรับสากลวิวัตน์(Internationalization) เมื่อ พ.ศ. 2538 และมาตรฐานที่ตรวจชำระใหม่เมื่อ พ.ศ. 2542 เรียกกันว่า ภาษาซี99(C99) มาตรฐานรุ่นปัจจุบันก็ได้รับอนุมัติเมื่อเดือนธันวาคม พ.ศ. 2554 เรียกกันว่า ภาษาซี11(C11)

ภาษาซีเป็นภาษาที่ใช้ในการมีปฏิสัมพันธ์เช่น เชิงคำสั่ง (หรือเชิงกระบวนการ) ถูกออกแบบขึ้นเพื่อใช้แปลด้วยตัวแปลโปรแกรมแบบการเชื่อมโยงที่ตรงไปตรงมา สามารถเข้าถึงหน่วยความจำในระดับล่าง เพื่อสร้างภาษาที่จับคู่อย่างมีประสิทธิภาพกับชุดคำสั่งเครื่อง และแทบไม่ต้องการสนับสนุนใด ๆ ขณะทำงาน ภาษาซีจึงเป็นประโยชน์สำหรับหลายโปรแกรมที่ก่อนหน้านี้เคยเขียนในภาษาแอสเซมบลีมาก่อน

หากคำนึงถึงความสามารถในระดับล่าง ภาษาซีถูกออกแบบขึ้นเพื่อส่งเสริมการเขียนโปรแกรมที่ขึ้นอยู่กับเครื่องใดเครื่องหนึ่ง(Machine-independent) โปรแกรมภาษาซีที่เขียนขึ้นตามมาตรฐานและเคลื่อนย้ายได้ สามารถแปลได้บนแพลตฟอร์มคอมพิวเตอร์และระบบปฏิบัติการต่าง ๆ อย่างกว้างขวาง โดยแก้ไขรหัสต้นฉบับเพียงเล็กน้อยหรือไม่ต้องแก้ไขเลย

ภาษาซียังมีลักษณะเฉพาะต่อไปนี้เพิ่มเติม

- ตัวแปรอาจถูกซ่อนในบล็อกซ้อนใน
- ชนิดตัวแปรไม่เคร่งครัด เช่นข้อมูลตัวอักษรสามารถใช้เป็นจำนวนเต็ม
- เข้าถึงหน่วยความจำคอมพิวเตอร์ในระดับต่ำโดยแปลงที่อยู่ในเครื่องด้วยชนิดตัวแปรตัวชี้(Pointer)
- ฟังก์ชันและตัวชี้ข้อมูลรองรับการทำงานในภาวะหลายรูปแบบ(Polymorphism)
- การกำหนดดัชนีแถวลำดับสามารถทำได้ด้วยวิธีรอง คือนิยามในพจน์ของเลขคณิตของตัวชี้
- ตัวประมวลผลก่อนสำหรับการนิยามแมโคร การรวมไฟล์รหัสต้นฉบับ และการแปลโปรแกรมแบบมีเงื่อนไข
- ความสามารถที่ซับซ้อนเช่น ไอ/โอ การจัดการสายอักขระ และฟังก์ชันทางคณิตศาสตร์ รวมอยู่ในไลบรารี
- คำหลักที่สงวนไว้มีจำนวนค่อนข้างน้อย
- ตัวดำเนินการแบบประสมจำนวนมาก อาทิ +=, -=, *=, ++ ฯลฯ

โครงสร้างการเขียน คล้ายภาษาพีมากกว่าภาษาอัลกอล ตัวอย่างเช่น

- ใช้วงเล็บปีกกา { ... } แทนที่จะเป็น begin ... end ในภาษาอัลกอล 60 หรือวงเล็บโค้ง (...) ในภาษาอัลกอล 68

- เท่ากับ = ใช้สำหรับกำหนดค่า เหมือนภาษาฟอร์แทรน แทนที่จะเป็น := ในภาษาอัลกอล
- เท่ากับสองตัว == ใช้สำหรับเปรียบเทียบความเท่ากัน แทนที่จะเป็น .EQ. ในภาษาฟอร์แทรนหรือ = ในภาษาเบสิกและภาษาอัลกอล
- ตรรกะ "และ" กับ "หรือ" แทนด้วย && กับ || ตามลำดับ แทนที่จะเป็นตัวดำเนินการ \wedge กับ \vee ในภาษาอัลกอล แต่ตัวดำเนินการดังกล่าวจะไม่ประเมินค่าตัวถูกดำเนินการทางขวา ถ้าหากผลลัพธ์จากทางซ้ายสามารถพิจารณาได้แล้ว เหตุการณ์เช่นนี้เรียกว่าการประเมินค่าแบบลัดวงจร (short-circuit evaluation) และตัวดำเนินการดังกล่าวก็มีความหมายต่างจากตัวดำเนินการระดับบิต & กับ |

- ภาษาซีพลัสพลัส (C++)

ภาษาซีพลัสพลัส(C++ Language) เป็นภาษาโปรแกรมคอมพิวเตอร์เนกประสงค์ มีโครงสร้างภาษาที่มีการจัดชนิดข้อมูลแบบสถิติก(Statically Typed) และสนับสนุนรูปแบบการเขียนโปรแกรมที่หลากหลาย(Multi-paradigm Language) ได้แก่ การโปรแกรมเชิงกระบวนการคำสั่ง, การนิยามข้อมูล, การโปรแกรมเชิงวัตถุ, และการโปรแกรมแบบเจเนริก(Generic Programming)

เบียเนอ สตราสตร็อบ(Bjarne Stroustrup) จากเบลล์แล็บส์(Bell Labs) เป็นผู้พัฒนาภาษาซีพลัสพลัส(เดิมใช้ชื่อ "C with classes") ในปี ค.ศ. 1983 เพื่อพัฒนาภาษาซีดั้งเดิม สิ่งที่พัฒนาขึ้นเพิ่มเติมนั้นเริ่มจากการเพิ่มเติมการสร้างคลาสจากนั้นก็เพิ่มคุณสมบัติต่างๆ ตามมา ได้แก่ เวอร์ชวลฟังก์ชัน การโอเวอร์โหลดโอเปอเรเตอร์ การสืบทอดหลายสาย เหมเพลต และการจัดการเอกเซพชันมาตรฐานของภาษาซีพลัสพลัสได้รับการรับรองในปี ค.ศ. 1998 เป็นมาตรฐาน ISO/IEC 14882:1998 เวอร์ชันล่าสุดคือเวอร์ชันในปี ค.ศ. 2003 ซึ่งเป็นมาตรฐาน ISO/IEC 14882:2003 ในปัจจุบันมาตรฐานของภาษาในเวอร์ชันใหม่(รู้จักกันในชื่อ C++0x) กำลังอยู่ในขั้นพัฒนา รูปแบบของการออกแบบมีดังนี้

- ภาษาซีพลัสพลัสได้ถูกออกแบบมาเพื่อเป็นภาษาสำหรับการเขียนโปรแกรมทั่วไป สามารถรองรับการเขียนโปรแกรมในระดับภาษาเครื่องได้ เช่นเดียวกับภาษาซี
- ในทางทฤษฎี ภาษาซีพลัสพลัสควรมีความเร็วเทียบเท่าภาษาซี แต่ในการเขียนโปรแกรมจริงนั้น ภาษาซีพลัสพลัสเป็นภาษาที่มีการเปิดกว้างให้โปรแกรมเมอร์เลือกรูปแบบการเขียนโปรแกรม ซึ่งทำให้มีแนวโน้มที่โปรแกรมเมอร์อาจใช้รูปแบบที่ไม่เหมาะสม ทำให้โปรแกรมที่เขียนมีประสิทธิภาพต่ำกว่าที่ควรจะเป็น และภาษาซีพลัสพลัสนั้นเป็นภาษาที่มีความซับซ้อนมากกว่าภาษาซี จึงทำให้มีโอกาสเกิดบั๊กขณะคอมไพล์มากกว่า
- ภาษาซีพลัสพลัสได้รับการออกแบบเพื่อเข้ากันได้กับภาษาซีในเกือบทุกกรณี
- มาตรฐานของภาษาซีพลัสพลัส ถูกออกแบบมาเพื่อไม่ให้เกิดการเจาะจงแพลตฟอร์มคอมพิวเตอร์
- ภาษาซีพลัสพลัสถูกออกแบบมาให้รองรับการเขียนโปรแกรมที่หลากหลาย(Multi-Paradigm)

- ภาษาซีชาร์ป (C#)

ภาษาซีชาร์ป(C# Programming Language) เป็นภาษาโปรแกรมแบบหลายโมเดล ที่ใช้ระบบชนิดข้อมูลแบบรัดกุม (strong typing) และสนับสนุนการเขียนโปรแกรมเชิงคำสั่ง การเขียนโปรแกรมเชิงประกาศ การเขียน

โปรแกรมเชิงฟังก์ชัน การเขียนโปรแกรมเชิงกระบวนการ การเขียนโปรแกรมเชิงวัตถุ(แบบคลาส) และการเขียนโปรแกรมเชิงส่วนประกอบ พัฒนาเริ่มแรกโดยบริษัทไมโครซอฟท์เพื่อทำงานบนดอตเน็ตเฟรมเวิร์ก โดยมีแอนเดอร์เฮลส์เบิร์ก(Anders Hejlsberg) เป็นหัวหน้าโครงการ และมีรากฐานมาจากภาษาซีพลัสพลัสและภาษาอื่นๆ (โดยเฉพาะภาษาเดลไฟและจาวา) โดยมีจุดมุ่งหมายให้เป็นภาษาสมัยใหม่ที่ไม่ซับซ้อน ใช้งานได้ทั่วไป(General-Purpose) และเป็นเชิงวัตถุเป็นหลัก

ปัจจุบันภาษาซีชาร์ปมีการรับรองให้เป็นมาตรฐานโดยเอ็กมาอินเตอร์เนชันแนล(Ecma International) และองค์การระหว่างประเทศว่าด้วยการมาตรฐาน(ISO) และมีรุ่นล่าสุดคือ C# 5.0 ที่ออกมาเมื่อวันที่ 15 สิงหาคม พ.ศ. 2555

- ภาษาโคบอล (COBOL)

ภาษาโคบอล(COBOL programming language) เป็นภาษาโปรแกรมระดับสูงภาษาหนึ่งที่อยู่มาอย่างยาวนาน COBOL ย่อมาจาก Common Business Oriented Language เป็นภาษาที่นิยมนำไปใช้ทางธุรกิจ ถูกพัฒนาขึ้นเมื่อ ค.ศ. 1959 โดยนักคอมพิวเตอร์กลุ่มหนึ่งที่เรียกตัวเองว่า Conference on Data Systems Languages(CODASYL) และตั้งแต่ปี ค.ศ. 1959 ภาษาโคบอลมีการแก้ไขและปรับปรุงอยู่ตลอด ดังนั้น เพื่อขจัดปัญหาความแตกต่างของตัวภาษาโคบอลในแต่ละเวอร์ชัน สถาบันมาตรฐานแห่งชาติอเมริกา (ANSI) จึงได้พัฒนามาตรฐานกลางขึ้นมาในปี ค.ศ. 1968 เป็นที่รู้จักกันในนามของ ANS COBOL ต่อมาเมื่อ ปี ค.ศ. 1974 ทาง ANSI ได้นำเสนอ ANS COBOL รุ่นใหม่ที่มีคุณสมบัติที่ดีกว่ารุ่น 1968 และในปี ค.ศ. 1985 ANSI ก็นำเสนออีกรุ่นหนึ่งที่มีคุณสมบัติมากกว่ารุ่นปี 1974

รูปแบบภาษาโคบอลแบ่งออกเป็น 4 ดิวิชั่น คือ

1. Identification division การกำหนดชื่อโปรแกรมและชื่อผู้เขียน
2. Environment division การอธิบายเกี่ยวกับคอมพิวเตอร์
3. Data division การอธิบายเกี่ยวกับการประมวลผลข้อมูล
4. Procedure division การอธิบายเกี่ยวกับขั้นตอนการประมวลผล

- ภาษาปาสกาล (Pascal)

ภาษาปาสกาล เป็นภาษาโปรแกรมที่ใช้กันอย่างกว้างขวาง โดยเฉพาะในวงการศึกษา คิดค้นขึ้นโดย นิเคลาส์แวร์ท(Niklaus Wirth) นักวิทยาการคอมพิวเตอร์ชาวสวิสเซอร์แลนด์ ในปี ค.ศ. 1970 เพื่อช่วยในการเรียนการสอน การเขียนโปรแกรมโครงสร้าง(Structured Programming) ภาษาปาสกาลนั้นพัฒนาขึ้นมาจาก ภาษาอัลกอล(Algol), และชื่อปาสกาลนั้น ตั้งเพื่อเป็นเกียรติแก่ แบลส ปาสกาล(Blaise Pascal) นอกเหนือจากภาษาปาสกาลแล้ว เวอร์ทได้พัฒนา ภาษาโมดูลาทู(Modula-2) และ โอเบอรอน(Oberon) ซึ่งมีโครงสร้างคล้ายกับภาษาปาสกาล แต่สามารถรองรับการเขียนโปรแกรมเชิงวัตถุ(Object-oriented Programming)

โปรแกรมภาษาปาสกาลทุกอัน จะเริ่มต้นด้วยคีย์เวิร์ด Program และส่วนของโค้ดจะอยู่ระหว่างคีย์เวิร์ด Begin และ End ภาษาปาสกาลนั้นไม่สนใจความแตกต่างระหว่างตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก ("end" มีผลเท่ากับ "End"). เซมิโคลอน (;) ใช้เพื่อแบ่งคำสั่ง และ มหัพภาค(.) ใช้เมื่อจบโปรแกรม (หรือยูนิต)

ภาษาปาสกาลเป็นภาษาที่มีโครงสร้างที่ตายตัว เช่นการประกาศตัวแปร จะอยู่ระหว่างโปรแกรมกับปีกินโดยไม่สามารถไปประกาศที่อื่นได้เหมือนกับภาษาวิซวลเบสิก, ภาษาซี หรือภาษาอื่น ๆ ทำให้ผู้เรียนได้ทราบถึงขั้นตอนการเขียนโปรแกรมที่ถูกต้อง เพื่อง่ายต่อการตรวจสอบในภายหลัง

- ภาษาเบสิก (BASIC)

ภาษาเบสิก(BASIC Programming Language) เป็นภาษาโปรแกรมที่ออกแบบมาให้ใช้งานได้ง่าย และยังได้รับความนิยมมาจนถึงทุกวันนี้ เบสิกออกแบบมาให้ใช้กับคอมพิวเตอร์ตามบ้าน

ชื่อภาษาเบสิกย่อมาจาก Beginner's All-purpose Symbolic Instruction Code ต้องเขียนด้วยตัวพิมพ์ใหญ่เสมอ

บริษัทไมโครซอฟท์ได้นำภาษาเบสิกมาปรับปรุงให้ทันสมัย และพัฒนาเครื่องมือพัฒนาโปรแกรมวิซวลเบสิก (Visual Basic) ทำให้เบสิกได้รับความนิยมในการพัฒนาโปรแกรมยุคใหม่ รุ่นล่าสุดของวิซวลเบสิกเรียกว่า VB.NET

- ภาษาฟอร์แทรน (FORTRAN)

ภาษาฟอร์แทรน(Fortran Programming Language หรือ FORTRAN) เป็นภาษาที่เก่าแก่ที่สุดของวงการคอมพิวเตอร์ ถูกพัฒนาขึ้นในยุคคริสต์ทศวรรษ 1950 นิยมนำไปใช้ในการคำนวณทางคณิตศาสตร์และวิทยาศาสตร์ จนถึงปัจจุบันนี้ ภาษาฟอร์แทรนก็ยังถูกใช้ในทางวิทยาศาสตร์อยู่

- ภาษาจาวา (Java)

ภาษาจาวา(Java programming language) เป็นภาษาโปรแกรมเชิงวัตถุ(Object Oriented Programming) พัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆ ที่ ซัน ไมโครซิสเต็มส์ ภาษาจาวาถูกพัฒนาขึ้นในปี พ.ศ. 2534 (ค.ศ. 1991) โดยเป็นส่วนหนึ่งของ โครงการกรีน(The Green Project) และสำเร็จออกสู่สาธารณะในปี พ.ศ. 2538 (ค.ศ. 1995) ซึ่งภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษาซีพลัสพลัส(C++) โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษาอ็อบเจกต์ทีฟซี(Objective-C) แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก(Oak) ซึ่งตั้งชื่อตามต้นโอ๊กใกล้ที่ทำงานของ เจมส์ กอสลิง แต่ว่ามีปัญหาทางลิขสิทธิ์ จึงเปลี่ยนไปใช้ชื่อ "จาวา" ซึ่งเป็นชื่อกาแฟแทน

และแม้ว่าจะมีชื่อคล้ายกัน แต่ภาษาจาวาไม่มีความเกี่ยวข้องใด ๆ กับภาษาจาวาสคริปต์(JavaScript) ปัจจุบันมาตรฐานของภาษาจาวาดูแลโดยจาวาคอมมิวนิตีโพรเซส(Java Community Process) ซึ่งเป็นกระบวนการอย่างเป็นทางการ ที่อนุญาตให้ผู้ที่สนใจเข้าร่วมกำหนดความสามารถในจาวาแพลตฟอร์มได้

- ภาษาจาวาสคริปต์ (JavaScript)

จาวาสคริปต์ (JavaScript) เป็นภาษาสคริปต์ ที่มีลักษณะการเขียนแบบโปรโตไทป์ (Prototyped-based Programming) ส่วนมากใช้ในหน้าเว็บเพื่อประมวลผลข้อมูลที่ฝั่งของผู้ใช้งาน แต่ก็ยังมีใช้เพื่อเพิ่มเติมความสามารถในการเขียนสคริปต์โดยฝังอยู่ในโปรแกรมอื่นๆ

ชั้น ไมโครซิสเต็มส์เป็นเจ้าของเครื่องหมายการค้า "JavaScript" โดยมันถูกนำไปใช้ภายใต้สัญญาอนุญาตเพื่อการพัฒนาเทคโนโลยีโดย เน็ตสเคป และมูลนิธิมอซิลลา

- ภาษาเพิร์ล (Perl)

ภาษาเพิร์ล (Perl Language) ย่อมาจาก Practical Extraction and Report Language เป็นภาษาโปรแกรมแบบไดนามิก พัฒนาโดยนายแลร์รี วอลล์ (Larry Wall) ในปี ค.ศ. 1987 เพื่อใช้งานกับระบบปฏิบัติการยูนิกซ์ ภาษาเพิร์ลนั้นถูกออกแบบมาให้ใช้งานได้ง่าย โครงสร้างของภาษาจึงไม่ซับซ้อน มีลักษณะคล้ายกับภาษาซี นอกจากนี้เพิร์ลยังได้แนวคิดบางอย่างมาจากเชลล์สคริปต์, ภาษา AWK, sed และ Lisp ปัจจุบันเวอร์ชันล่าสุดคือ 5.18.0

- ภาษาพีเอชพี (PHP)

พีเอชพี (PHP) คือ ภาษาคอมพิวเตอร์ในลักษณะเซิร์ฟเวอร์-ไซด์ สคริปต์ โดยลิขสิทธิ์อยู่ในลักษณะโอเพนซอร์ส ภาษาพีเอชพีใช้สำหรับจัดทำเว็บไซต์ และแสดงผลออกมาในรูปแบบ HTML โดยมีรากฐานโครงสร้างคำสั่งมาจากภาษา ภาษาซี ภาษาจาวา และ ภาษาเพิร์ล ซึ่ง ภาษาพีเอชพี นั้นง่ายต่อการเรียนรู้ ซึ่งเป้าหมายหลักของภาษานี้คือให้นักพัฒนาเว็บไซต์สามารถเขียน เว็บเพจ ที่มีการตอบโต้ได้อย่างรวดเร็ว พีเอชพีรุ่นล่าสุดคือ PHP 5.4.0 ส่วนรุ่นพัฒนาคือ PHP 6.0.0-dev

- ภาษาไพทอน (Python)

ภาษาไพทอน (Python Programming Language) เป็นภาษาโปรแกรมระดับสูง เพื่อใช้งานทั่วไป แบบอินเทอร์พรีเตอร์ ที่สร้างโดยกิด ฟาน รอสซัม (Guido van Rossum) ในพ.ศ. 2533 ปัจจุบันดูแลโดย มูลนิธิซอฟต์แวร์ไพทอน จุดเด่นของภาษาไพทอน คือ

ความเป็นภาษาสคริปต์

เนื่องจากไพทอนเป็นภาษาสคริปต์ ทำให้ใช้เวลาในการเขียนและคอมไพล์ไม่มาก ทำให้เหมาะกับงานด้าน การดูแลระบบ (System administration) เป็นอย่างยิ่ง

ไวยากรณ์ที่อ่านง่าย

ไวยากรณ์ของไพทอนได้กำจัดการใช้สัญลักษณ์ที่ใช้ในการแบ่งบล็อกของโปรแกรม และใช้การย่อหน้าแทน ทำให้สามารถอ่านโปรแกรมที่เขียนได้ง่าย นอกจากนั้นยังมีการสนับสนุนการเขียน docstring ซึ่งเป็นข้อความสั้นๆ ที่ใช้อธิบายการทำงานของฟังก์ชัน, คลาส, และโมดูลอีกด้วย

ความเป็นภาษา Glue

ไพทอนเป็นภาษา Glue (Glue Language) ได้อย่างดีเนื่องจากสามารถเรียกใช้ภาษาโปรแกรมอื่นๆ ได้หลายภาษา ทำให้เหมาะที่จะใช้เขียนเพื่อประสานงานโปรแกรมที่เขียนในภาษาต่างกันได้

- ภาษาโปรล็อก (Prolog)

ภาษาโปรล็อก (Prolog Language) เป็นภาษาสำหรับการเขียนโปรแกรมเชิงตรรกะ ได้ชื่อมาจาก *PRO*grammation en *LOG*ique (Logic Programming) สร้างขึ้นโดย Alain Colmerauer ราว ค.ศ. 1972 ภาษาโปรล็อกเกิดจากความพยายามที่จะสร้างภาษาที่อาศัยวิธีการทางตรรกศาสตร์แทนที่จะกำหนดคำสั่งอย่างละเอียดให้กับคอมพิวเตอร์

ภาษาโปรล็อกถูกนำไปใช้ในโปรแกรมสำหรับปัญญาประดิษฐ์ และภาษาศาสตร์เชิงคำนวณ โดยเฉพาะการประมวลผลภาษาธรรมชาติ ไวยากรณ์และความหมายของภาษานั้นเรียบง่ายและชัดเจน งานวิจัยจำนวนมากที่ทำให้เกิดการพัฒนภาษาโปรล็อกในปัจจุบันนั้น เป็นผลมาจากโครงการระบบคอมพิวเตอร์ยุคที่ห้า (Fifth Generation Computer Systems Project - FGCS) ซึ่งเลือกรูปแบบหนึ่งของภาษาโปรล็อกเป็นภาษาแก่น (Kernel Language) ของระบบปฏิบัติการ

ภาษาโปรล็อกมีพื้นฐานมาจากแคลคูลัสภาคแสดง (Predicate Calculus) หรือเรียกเต็ม ๆ ว่า แคลคูลัสภาคแสดงอันดับที่หนึ่ง (First-order Predicate Calculus) โดยจำกัดให้ใช้เฉพาะอนุประโยคของฮอร์น (Horn Clause) การดำเนินการของโปรแกรมโปรล็อก ก็คือการประยุกต์วิธีพิสูจน์ทฤษฎีบทโดยใช้ริโซลูชันอันดับที่หนึ่ง (First-order Resolution) แนวคิดพื้นฐานที่เกี่ยวข้องได้แก่ การทำให้เท่ากัน (Unification), การเรียกซ้ำจากส่วนท้าย (Tail Recursion), การย้อนรอย (Backtracking)

- ภาษาอ็อบเจกทีฟ-ซี (Objective-C)

ภาษาอ็อบเจกทีฟ-ซี (Objective-C Language หรือ ObjC) เป็นภาษาโปรแกรมเชิงวัตถุและมีสมบัติการสะท้อน โดยแรกเริ่ม ภาษาอ็อบเจกทีฟ-ซี พัฒนาขึ้นจากภาษาซีโดยยังคงคุณลักษณะของภาษาซีไว้ครบทุกประการ เพียงแต่เพิ่มระบบส่งข้อความ (Messaging) แบบเดียวกับภาษาสมอลทอล์คเข้าไปเท่านั้น (Objective-C Runtime) ปัจจุบันภาษาอ็อบเจกทีฟ-ซีมีคุณสมบัติอื่นๆเพิ่มเติมจากการพัฒนาภาษาอ็อบเจกทีฟ-ซี 2.0 โดยบริษัทแอปเปิล

ปัจจุบันภาษาอ็อบเจกทีฟ-ซีถูกใช้มากใน Cocoa (API) ใน Mac OS X, GNUstep (API) และ Cocotron (API) เป็นต้น ซึ่งระบบเหล่านี้ได้รับการพัฒนาขึ้นโดยมีพื้นฐานจากมาตรฐาน OpenStep (API) ใน Nextstep (Operating System) โดยมีภาษาอ็อบเจกทีฟ-ซีเป็นภาษาหลัก ปัจจุบัน Mac OS X ใช้ Cocoa เป็นเฟรมเวิร์กสำหรับสร้างโปรแกรมประยุกต์ โดยไลบรารีและ/หรือ API เหล่านี้เป็นเพียงส่วนเพิ่มขยาย (Software Extension) เท่านั้น โปรแกรมที่ใช้ภาษาอ็อบเจกทีฟ-ซีทั่วไปที่ไม่ได้ใช้ส่วนเพิ่มขยายเหล่านี้ก็ยังสามารถคอมไพล์ได้ เช่นอาจใช้แต่ gcc ซึ่งรองรับภาษาอ็อบเจกทีฟ-ซี

- ภาษารูบี้ (Ruby)

ภาษารูบี้ (Ruby Language) เป็นภาษาโปรแกรมเชิงวัตถุ ที่ได้รับอิทธิพลของโครงสร้างภาษามาจาก ภาษาเพิร์ลกับภาษาเอดา มีความสามารถในเชิงวัตถุแบบเดียวกับภาษาสมอลทอล์ค และมีความสามารถหลายอย่างจาก ภาษาไพทอน, ภาษาลิสป์, ภาษา Dylan และภาษา CLU ตัวแปลภาษารูบี้ตัวหลักเป็นซอฟต์แวร์เสรี และเป็นตัวแปลแบบอินเตอร์พรีเตอร์

2.12.3 ไบรารีโอเพนซีวี (OpenCV Library)

โอเพนซีวีเป็นไลบรารีที่มีใบอนุญาตบีเอสดี(BSD Certificated) และเป็นโอเพนซอร์ซเป็นไลบรารีที่มีอัลกอริทึมอยู่มากมาย โอเพนซีวีมีโครงสร้างแบบแยกส่วน ซึ่งหมายความว่า แพคเกจจะรวมไลบรารีต่างๆที่ใช้ร่วมกันไว้ โดยมอดูล มีดังต่อไปนี้

- core

Module สำหรับการอธิบายถึงโครงสร้างของข้อมูลขั้นพื้นฐาน รวมถึงอาเรย์หลายมิติ และฟังก์ชันพื้นฐานต่างๆที่ใช้ในมอดูลอื่นๆ

- improc

Module สำหรับการประมวลผลภาพ รวมถึงการกรองภาพ(Filtering) ทั้งแบบเชิงเส้นและไม่เป็นเชิงเส้น การเปลี่ยนแปลงทางเรขาคณิตของภาพ (การปรับขนาด, Warping, Remapping) การทำฮิสโตแกรมและอื่นๆ

- video

มอดูลเพื่อการวิเคราะห์วีดีโอ รวมถึงการประมาณการเคลื่อนที่ และขั้นตอนการตรวจจับวัตถุ

- calib3d

อัลกอริทึมทางเรขาคณิตแบบหลายมุมมอง, การสอบเทียบกล้องเพื่อการสร้างภาพสามมิติ

- features2d

การตรวจจับลักษณะเด่นของวัตถุ

- objdetect

การตรวจจับวัตถุและกรณีของกลุ่มที่กำหนดไว้ล่วงหน้า(เช่น ใบหน้า, ดวงตา, รถ หรืออื่นๆ)

- highgui

เพื่อให้ง่ายต่อการจับภาพจากวีดีโอ การแปลงสัญญาณวีดีโอและรูปภาพ

- gpu

อัลกอริทึมของจีพียู-แอกเซลเลอเรท(GPU-accelerated) จากมอดูลอื่น

- อื่นๆ

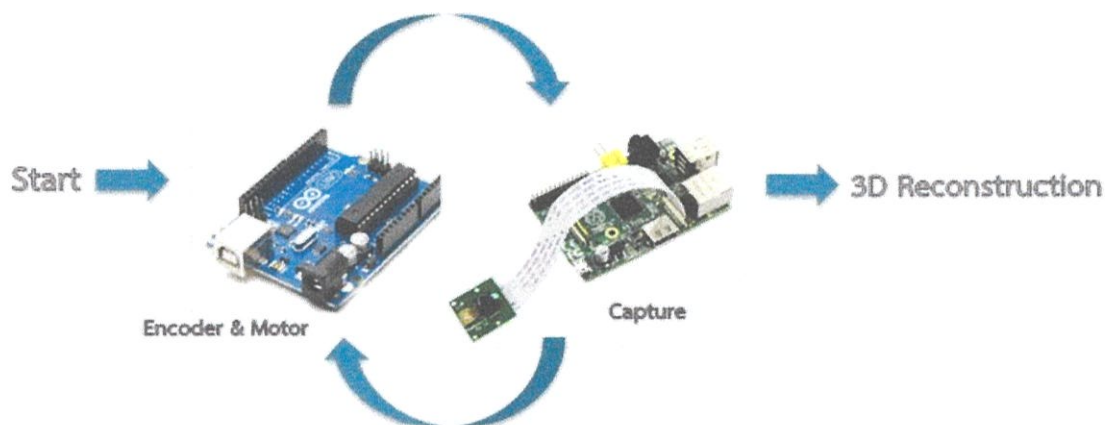
มอดูลสนับสนุนอื่นๆ เช่น แฟลน(FLANN), ไพธอน ไบนดิง(Pyton Bindings)

บทที่ 3

ระเบียบวิธีวิจัยขั้นตอนการดำเนินงานแผนการดำเนินงาน

3.1 การทำงาน

การทำงานของเครื่องสร้างแบบจำลอง 3 มิติสำหรับส่วนต่อขาของคนไข้ขาพิการ ทำโดยการใช้การทำงานของไมโครคอนโทรลเลอร์ 2 ชนิด คือ ราชเบอร์รี่พาย(Raspberry Pi) และ อาดูยโน่(Arduino) ดังรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการทำงานของเครื่องสร้างแบบจำลอง 3 มิติสำหรับส่วนต่อขาของคนไข้ขาพิการ

จากรูปเมื่อเริ่มการทำงาน ไมโครคอนโทรลเลอร์ อาดูยโน่(Arduino) จะอ่านค่าตำแหน่งของเฟืองโดยใช้เอนโคเดอร์ และส่งสัญญาณผ่านทางซีเรียลพอร์ตไปยังไมโครคอนโทรลเลอร์ตัวถัดไปคือ ราชเบอร์รี่พาย(Raspberry Pi) หลังจากนั้นก็จะทำการเก็บภาพ และส่งสัญญาณไปยัง ราชเบอร์รี่พาย(Raspberry Pi) ตัวถัดไปจนถึงตัวสุดท้ายแล้วส่งสัญญาณกลับมาให้ อาดูยโน่(Arduino) ทำการขับมอเตอร์ไปยังตำแหน่งถ่ายภาพตำแหน่งถัดไป เมื่อเก็บภาพครบตามที่ต้องการ ก็ทำการดึงข้อมูลภาพไปยังคอมพิวเตอร์เพื่อประมวลผลเป็นภาพสามมิติ

3.2 การออกแบบ

การออกแบบเครื่องสร้างแบบจำลอง 3 มิติสำหรับส่วนต่อขาของคนไข้ขาพิการ แบ่งออกเป็น 2 ส่วน คือ

1. การออกแบบด้านฮาร์ดแวร์ : ระบบการทำงานของกลไกและกล่อง
2. การออกแบบด้านซอฟต์แวร์ : ระบบการประมวลผลภาพในการเก็บภาพและสร้างแบบจำลองสามมิติ

3.1.1 การออกแบบด้านฮาร์ดแวร์

3.1.1.1 วัสดุและอุปกรณ์ที่ใช้

1. วงล้ออะลูมิเนียม ขนาดเส้นผ่านศูนย์กลาง 18 นิ้ว ความหนา 8 มิลลิเมตร จำนวน 2 อัน
2. แผ่นอะคริลิกรูปเฟือง ขนาดเส้นผ่านศูนย์กลาง 18 นิ้ว ความหนา 4.7 มิลลิเมตร 360 ซี่เฟือง
3. แผ่นอะคริลิกแกนกลางสำหรับยึดกล่อง ขนาดเส้นผ่านศูนย์กลาง นิ้ว ความหนา มิลลิเมตร
4. มอเตอร์กระแสตรง 12 โวลต์ 5000 รอบต่อนาที ทดรอบเป็น 50 รอบต่อนาที

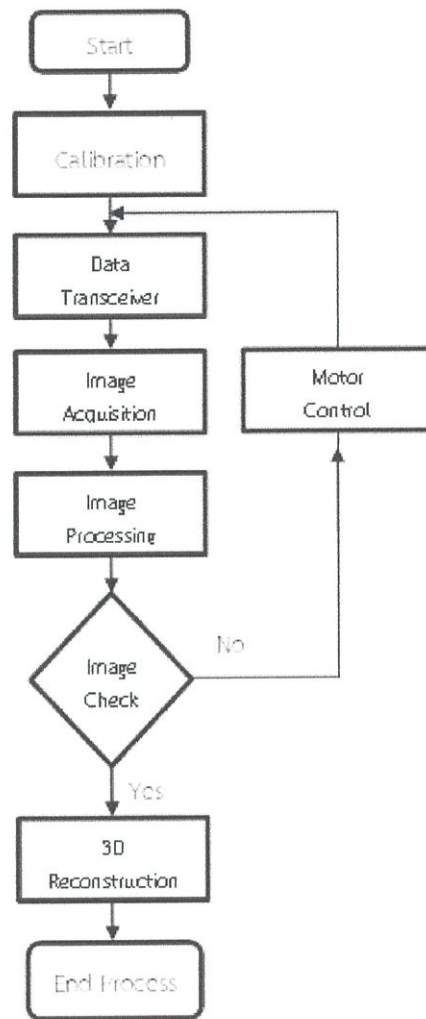
5. แผ่นไม้อัดแข็ง ความหนา 6 มิลลิเมตร ความกว้าง 30 เซนติเมตร ความยาว 45 เซนติเมตร
6. โรเตตติงดิสก์ วัสดุอะบีเอส(ABS) สีดำ ขนาดเส้นผ่านศูนย์กลาง 55 มิลลิเมตร ความหนา 2 มิลลิเมตร
7. เฟืองขับ วัสดุอะบีเอส สีดำ ขนาดเส้นผ่านศูนย์กลาง 47 มิลลิเมตร ความหนา 8 มิลลิเมตร 36 ซี่เฟือง
8. ส่วนยึดคล้องกับแผ่นอะคริลิก วัสดุพีแอลเอ(PLA) สีดำ จำนวน 10 อัน
7. แอลติอาร์ โฟโตรีซิสเตอร์เซ็นเซอร์มอดูลสำหรับอาดูยโน(KY-018) จำนวน 3 อัน
8. หัวส่งเลเซอร์เซ็นเซอร์มอดูลสำหรับอาดูยโน(AVR PIC KY-008) จำนวน 3 อัน
9. อาดูยโน อูโน อาร์พีรี จำนวน 1 อัน
10. อาดูยโนมอเตอร์ชิลด์ จำนวน 1 อัน
11. แรสบอร์รี่พายเอพลัส จำนวน 5 อัน
12. มอดูลคล้องแรสบอร์รี่พาย จำนวน 5 อัน
13. ตัวรับสัญญาณไวไฟ อีดียูพี 5 อัน

ในการสร้างโครงสร้างของเครื่องสร้างแบบจำลอง 3 มิติ ทำการยึดแผ่นอะคริลิกขนาด 18 นิ้ว ด้วยสกรูขนาด 1/4" จำนวน 12 อัน ให้อยู่ระหว่างวงล้ออะลูมิเนียมขนาด 18 นิ้ว จากนั้นติดตั้งวงล้อดังกล่าวพร้อมแผ่นอะคริลิกบนแผ่นไม้อัดแข็งโดยใช้สกรูพร้อมเหล็กฉากเป็นตัวยึดทั้งสี่มุม เพื่อยึดให้อยู่กับที่ จากนั้นทำการยึดตำแหน่งของเลเซอร์และโฟโตรีซิสเตอร์(Photoresistor) ชุดแรกที่บริเวณจุดต่ำสุดของแผ่นอะคริลิก เพื่อเป็นการสอบเทียบ(Calibration) ตำแหน่งของซี่เฟืองบนแผ่น จากนั้นทำการยึดแทนวงพร้อมมอเตอร์กระแสตรง นำเฟืองขับขนาด 47 มิลลิเมตร พร้อม โรเตตติงดิสก์(Rotating disk) มาใส่เข้ากับเพลลาของมอเตอร์ จากนั้นติดตั้งหัวเลเซอร์ และโฟโตรีซิสเตอร์อีกสองชุด เพื่อใช้สำหรับอ่านค่าที่เปลี่ยนแปลงไปของเอนโคเดอร์

ในส่วนของไมโครคอนโทรลเลอร์ นั้น แรสบอร์รี่พาย(Raspberry Pi) จะทำการติดบนแผ่นอะคริลิกพร้อมทั้งติดโมดูลคล้องเพื่อใช้ในการถ่ายภาพ ส่วน อาดูยโน(Arduino) จะติดอยู่บริเวณมอเตอร์กระแสตรง เพื่อทำการควบคุมกลไกการหมุนของเมเตอร์ และการรับ-ส่งสัญญาณระหว่าง แรสบอร์รี่พาย(Raspberry Pi) ทำให้เกิดการ ทำงานอย่างต่อเนื่อง

3.1.2 การออกแบบด้านซอฟต์แวร์

ในด้านซอฟต์แวร์จะมีการออกแบบให้ระบบการทำงานของโปรแกรมให้สามารถถ่ายภาพ และเปลี่ยนตำแหน่งของกล้องได้อย่างเป็นจังหวะ ทำงานกันอย่างเป็นระบบดังรูปที่ 3.1



รูปที่ 3.2 โฟลวชาร์ตไดอะแกรมของการออกแบบซอฟต์แวร์

จากรูปที่ 3.1 จะเห็นถึงลำดับขั้นตอนในการทำงานของโปรแกรมที่ได้ทำการออกแบบ โดยในแต่ละส่วนจะมีการออกแบบให้มันการทำงานสัมพันธ์กันระหว่าง ราชเบอร์รี่พาย, อาดยูโน, เอนโคดเดอร์ และมอเตอร์ โดยแต่ละส่วนมีรายละเอียดดังนี้

3.1.2.1 การสอบเทียบเครื่องมือก่อนใช้งาน (Calibration)

เป็นการตรวจสอบระบบให้มีความพร้อมต่อการใช้งาน โปรแกรมจะทำการทดสอบระบบ รวมถึงทำให้ตำแหน่งของกล้องอยู่ในตำแหน่งที่ถูกต้องก่อนการใช้งาน โดยทำการหมุนมอเตอร์เพื่อทดสอบการอ่านข้อมูลของเอนโคดเดอร์ และทำการปรับเทรซโซลของโพโตริซิสเตอร์ ทำให้สามารถอ่านตำแหน่งขององศาของเฟืองอะคริลิกได้ถูกต้องในแต่ละสภาพของแสงที่แตกต่างในแต่ละพื้นที่การเก็บผลข้อมูล แล้วทำให้สามารถเริ่มการทำงานของเครื่องในตำแหน่งเดิมที่ต้องการได้ และจะได้ตำแหน่งเดิมของกล้องที่ถูกต้องทุกครั้งที่การใช้งาน รวมถึงยังเป็นการให้โปรแกรมทำการทดสอบการสื่อสารระหว่าง แรชเบอร์รี่พายและอาดยูโนเพื่อเตรียมความพร้อมก่อนทำการ รับและส่งสัญญาณการถ่ายข้อมูลในการถ่ายรูป

3.1.2.2 การรับ-ส่งข้อมูล (Data Transceiver)

แรสเบอร์รี่พายและอาδυโนจะสื่อสารและส่งสัญญาณผ่านซีเรียลพอร์ต(Serial Port) ของแต่ละไมโครคอนโทรลเลอร์โดยทางแรสเบอร์รี่พาย จะใช้ไลบรารี wiringPi/serialPi ที่มีคำสั่งในการสื่อสารข้อมูลผ่านซีเรียลพอร์ต ทางอาδυโนจะใช้ไลบรารี softwareSerial ซึ่งเป็นไลบรารีสำเร็จที่ทำได้ทำให้สามารถส่งข้อมูลผ่านซีเรียลพอร์ตได้ ไมโครคอนโทรลเลอร์ทั้งสองนี้จะส่งข้อมูลเป็นไบต์ ไปยังโมดูลบนไมโครคอนโทรลเลอร์อีกฝั่ง เมื่อได้รับข้อมูลแล้วจึงสามารถที่จะไปยังการทำงานส่วนต่อไปได้ โดยการส่งข้อมูลผ่านซีเรียลพอร์ตสามารถทำได้โดยการเขียนโปรแกรมให้เหมาะสมกับภาษาที่ใช้งานของแต่ละไมโครคอนโทรลเลอร์ รวมถึงต้องมีการเชื่อมต่อสายรับ-ส่งข้อมูล(tx, rx และ ground) เพื่อให้ไมโครคอนโทรลเลอร์ทั้งสองชนิดสามารถรับส่งข้อมูลให้แกกันและกันได้

3.1.2.3 การได้มาของข้อมูลภาพ (Image Acquisition)

การมาของภาพในโปรแกรมนี้ จะใช้กล้องโมดูลกล้องแรสเบอร์รี่พาย และหลักการได้มาของข้อมูลภาพ(Image Acquisition) เพื่อให้ได้ภาพที่สามารถนำไปใช้งานได้ ส่วนในทางโปรแกรมของ แรสเบอร์รี่พายจะใช้ไลบรารีโอเพนซีวี เพื่อให้สามารถเข้าถึง,ควบคุม และตั้งค่าต่างๆ ของกล้องผ่านโปรแกรมได้ง่าย ซึ่งในการตั้งค่าต่างๆ ของกล้องเช่น ความละเอียด ขนาดของเฟรมภาพ เฟรมเรท และโทนสีของภาพ เป็นต้น ล้วนมีความสำคัญส่งผลให้กล้องทำงานได้อย่างปกติ หากมีการตั้งค่าที่ไม่สัมพันธ์กับการถ่าย จะทำให้กล้องเกิดการค้างส่งผลให้โปรแกรมไม่สามารถทำงานต่อไป

3.1.2.4 การประมวลผลข้อมูลภาพ (Image Processing)

การประมวลผลภาพจะไลบรารีโอเพนซีวี ในการเขียนโปรแกรม โดยโปรแกรมที่ออกแบบจะทำการลบพื้นหลังของภาพที่ถ่ายได้ด้วยหลักการลบภาพพื้นหลัง(Background Subtraction) เพื่อให้ได้ภาพวัตถุที่ต้องเพียงอย่างเดียว พร้อมทั้งยังได้ทำการปรับเทรซโซลของภาพเพื่อให้ภาพที่ได้จากการลบภาพพื้นหลังมีความชัดเจนและลบสิ่งต่างๆที่ไม่จำเป็นในภาพออกเพื่อป้องกันไม่ให้เกิดปัญหาในกระบวนการทำแบบจำลองโมเดลสามมิติ

3.1.2.5 การตรวจสอบจำนวนภาพ (Image Check)

ทำการตรวจสอบจำนวนภาพที่ถ่าย ในขั้นตอนการเขียนโปรแกรมจะมีการตั้งจำนวนภาพที่ต้องการที่จะถ่ายไว้ รวมถึงมีการระบุงบุงศในการถ่ายภาพ ในการถ่ายภาพแต่ละครั้งจะมีการเปลี่ยนองศาการถ่ายภาพโดยมอเตอร์ หากภาพยังไม่ครบตามจำนวนองศาที่ได้กำหนดไว้ จะทำการส่งสัญญาณกลับไปยัง อาδυโน (Arduino) เพื่อทำการควบคุมมอเตอร์กระแสตรง ให้ตำแหน่งของกล้องอยู่ในองศาถัดไป หากถ่ายภาพครบจำนวนองศาที่ตั้งไว้แล้ว ก็สามารถนำรูปที่ได้ไปประมวลผลในโปรแกรมปรับแต่งภาพและจำลองโมเดลสามมิติ

3.1.2.6 การควบคุมมอเตอร์ (Motor Control)

การควบคุมมอเตอร์กระแสตรงจะใช้หลักการของเอนโคดเดอร์ ในการวัดองศาที่เปลี่ยนไปของแผ่นอะคริลิก โดยใช้โพโตรีซิสเตอร์, เลเซอร์ และดิสก์เอนโคดเดอร์ ที่ออกแบบมา เริ่มต้นเมื่ออาδυโนได้รับสัญญาณจากแรสเบอร์รี่พาย ก็จะทำการหมุนมอเตอร์กระแสตรง เมื่อองศาที่ทำการวัดมีการเปลี่ยนไปตาม

ตำแหน่งองศาที่ต้องการ ก็จะทำให้การหยุดหมุนมอเตอร์กระแสดตรง พร้อมทั้งทำการส่งสัญญาณกลับไปยังแรสเบอร์รี่พาย เพื่อทำถ่ายรูปในองศาใหม่จนครบจำนวนองศาที่ได้ตั้งค่าไว้ในตอนต้น

3.1.2.7 การสร้างภาพสามมิติ (3D Reconstruction)

ข้อมูลของภาพที่ถ่ายมาได้ และได้ผ่านกระบวนการลบภาพพื้นหลัง จะถูกบันทึกอยู่ในเมมโมรี่การ์ดของไมโครคอนโทรลเลอร์แรสเบอร์รี่พาย ข้อมูลภาพเหล่านี้สามารถนำมาใช้งานได้ด้วยใช้โปรแกรมไฟล์ซิลล่า โดยการระบุไอพีแอดเดรสของโมดูลรับสัญญาณไร้สายที่ติดอยู่กับไมโครคอนโทรลเลอร์แต่ละตัว ทำให้สามารถเข้าถึงข้อมูลที่อยู่ในเมมโมรี่การ์ดของไมโครคอนโทรลเลอร์ได้ เมื่อดึงข้อมูลภาพของไมโครคอนโทรลเลอร์แต่ละตัวมาได้แล้ว หลังจากนั้นภาพที่ได้จะสามารถนำไปประมวลผลและสร้างเป็นแบบจำลองโมเดลสามมิติได้ในคอมพิวเตอร์ โดยใช้โปรแกรมอิมเมจเจ (ImageJ) เพื่อทำการปรับข้อมูลภาพอีกรอบหนึ่ง เพื่อให้ได้ข้อมูลภาพที่พร้อมเข้าสู่โปรแกรมสร้างภาพสามมิติ(Biocross 3D Reconstruction)

3.3 เครื่องมือที่ใช้ในการทดลอง

1. เครื่องเก็บภาพแบบจำลอง 3 มิติ
2. คอมพิวเตอร์สำหรับการประมวลผล, การวิเคราะห์ และการแสดงผลแบบจำลอง 3 มิติ
3. โปรแกรม ImageJ ใช้สำหรับปรับข้อมูลภาพ
4. โปรแกรม FileZilla ใช้สำหรับดึงข้อมูลภาพ
5. โปรแกรม Biocross 3D Reconstruction ใช้สำหรับการสร้างภาพ 3 มิติจากข้อมูลภาพที่มี
6. ไมโครคอนโทรลเลอร์อาดูยโน ใช้ควบคุมมอเตอร์เพื่อขับเฟืองอะคริลิกและเอนโคเดอร์เพื่อนับตำแหน่งองศาของเฟืองอะคริลิกที่เปลี่ยนไป
7. ไมโครคอนโทรลเลอร์แรสเบอร์รี่พาย ใช้สำหรับการเก็บข้อมูลภาพถ่ายสำหรับการสร้างภาพสามมิติ

3.4 ขั้นตอนการดำเนินงาน

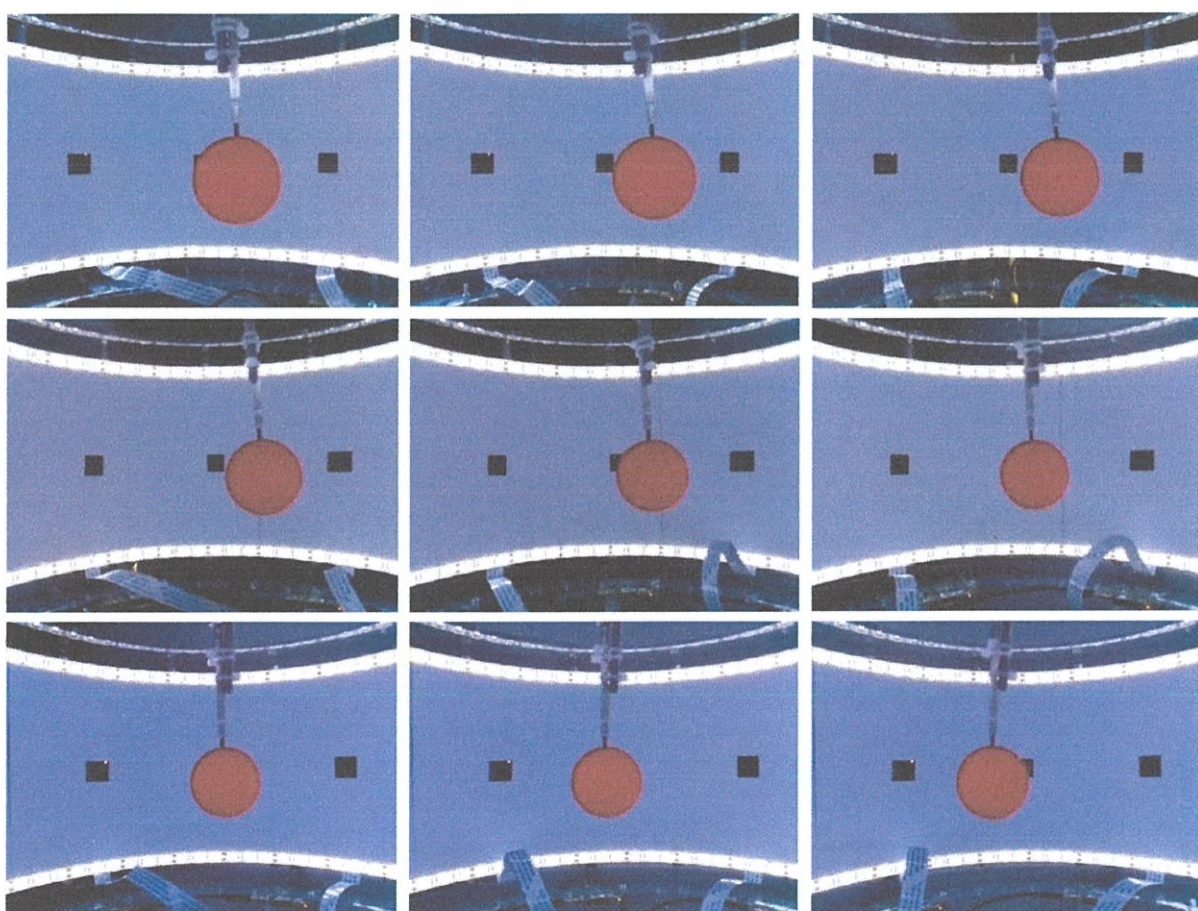
- 3.4.1 การทดลองเก็บข้อมูล ตรวจสอบการทำงานของระบบที่สร้างขึ้น
 - 3.4.1.1 การทดลองการทำงานด้านกลไกของเครื่องมือ ทดสอบความเสถียรและความสิ้นไหล
 - 3.4.1.2 การทดลองการสื่อสารระหว่างไมโครคอนโทรลเลอร์ และการเก็บภาพถ่ายด้วยกล้อง
- 3.4.2 การจัดเก็บผลจากวัตถุหรืออุปกรณ์จำลองเลียนแบบคนไข้แขนขาพิการ เพื่อทำการวิเคราะห์
- 3.4.3 การวิเคราะห์ผลเชิงสถิติ โดยการนำผลการวิเคราะห์จากระบบที่สร้างขึ้น

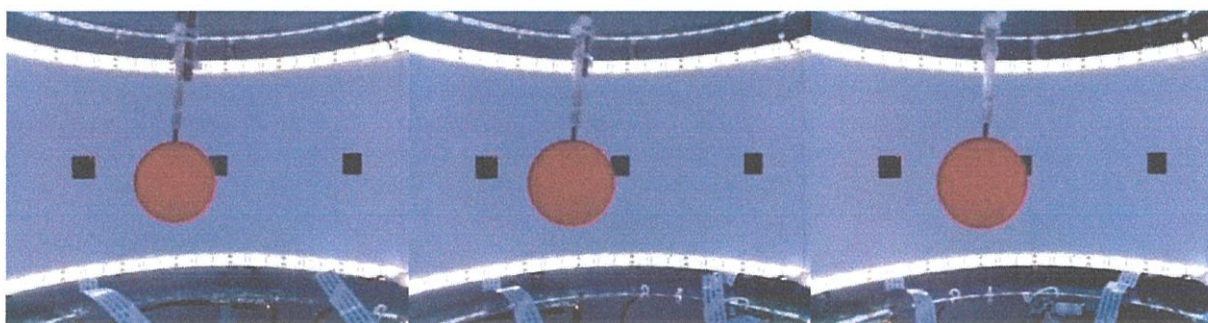
บทที่ 4

การทดลองและผลการทดลอง

ในบทนี้จะแสดงถึงการทดลองและผลการทดลองของเครื่องสร้างแบบจำลอง 3 มิติสำหรับส่วนต่อขาของคนไข้พิการ โดยการทดลองจะทำการโดยทำการเก็บผลของวัตถุหลายรูปทรง แล้วทำการประมวลผลภาพที่ได้จากเครื่องมือนำมาสร้างเป็นภาพ 3 มิติในคอมพิวเตอร์ แล้วนำผลที่ได้จากการสร้างภาพ 3 มิติ มาเปรียบเทียบกับวัตถุจริงที่ใช้ในการถ่าย

4.1 ผลการทดลองเก็บภาพลูกบิงปอง

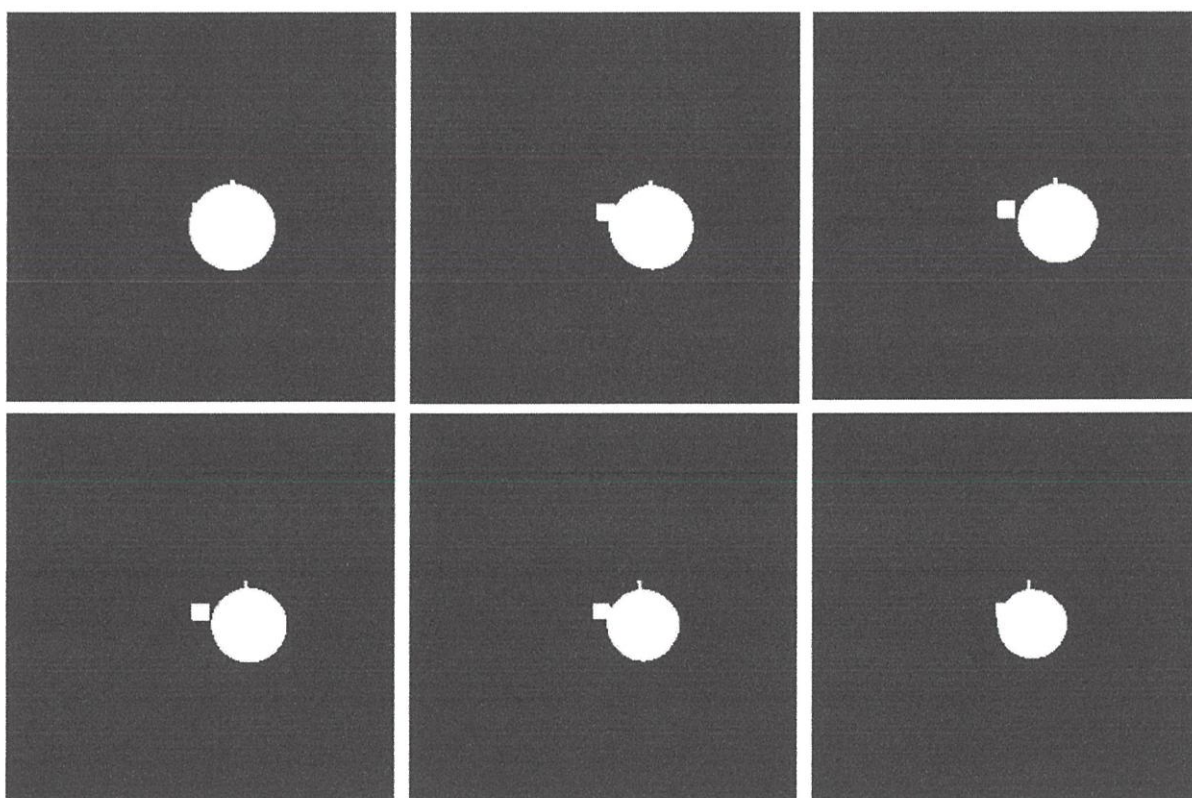


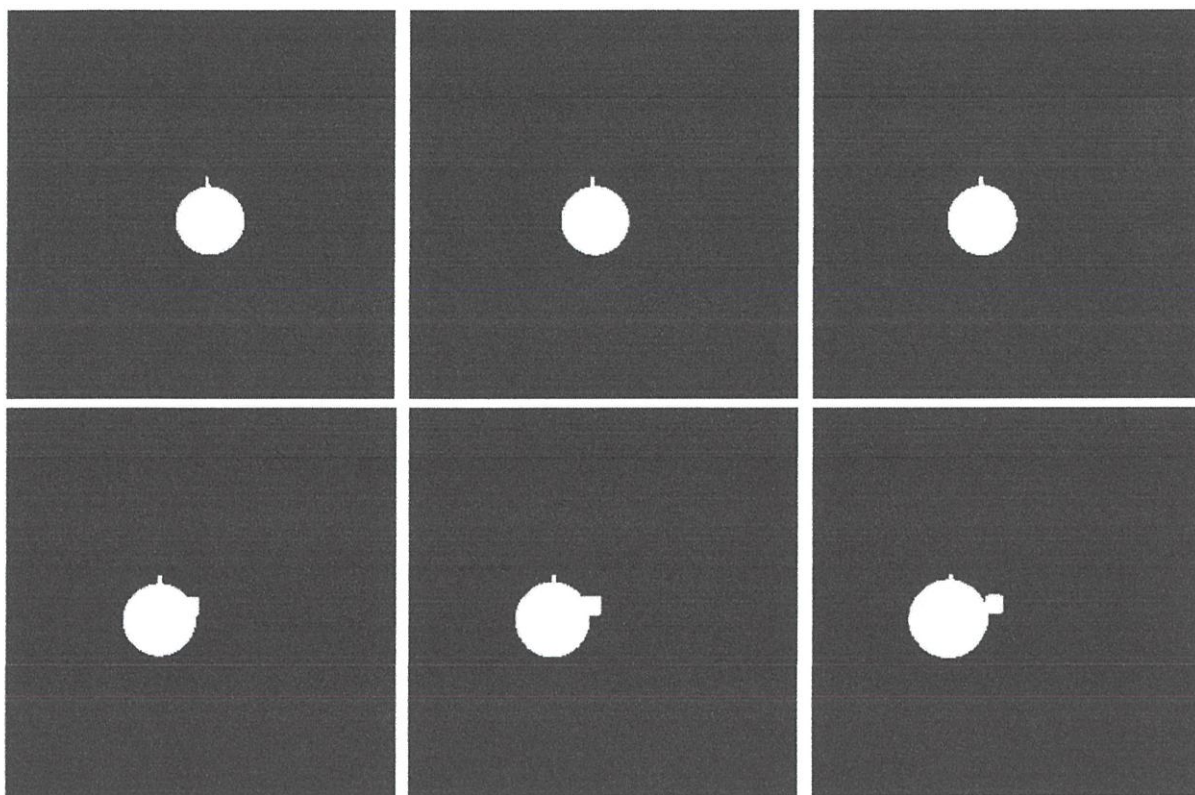


รูปที่ 4.1 การเก็บภาพจากกล้องแรสเบอร์รี่พาย

ภาพจากรูปที่ 4.1 เป็นตัวอย่างภาพ 12 ภาพ จากทั้งหมด 90 ภาพ เป็นภาพถ่ายที่ได้จากกล้องทั้ง 5 ตัว ที่มีระยะห่างกัน 36 องศา

4.2 ผลการทดลองภาพที่ถูกปรับเทอร์ชโซ

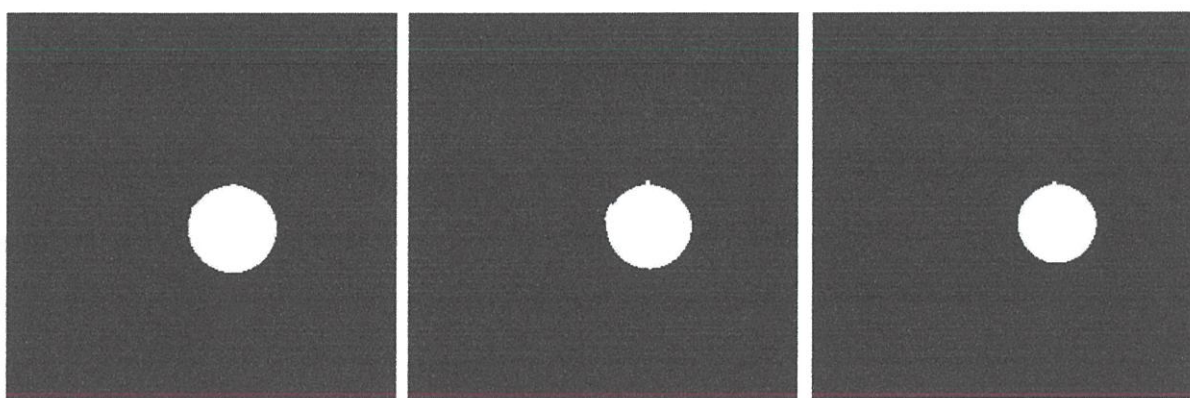


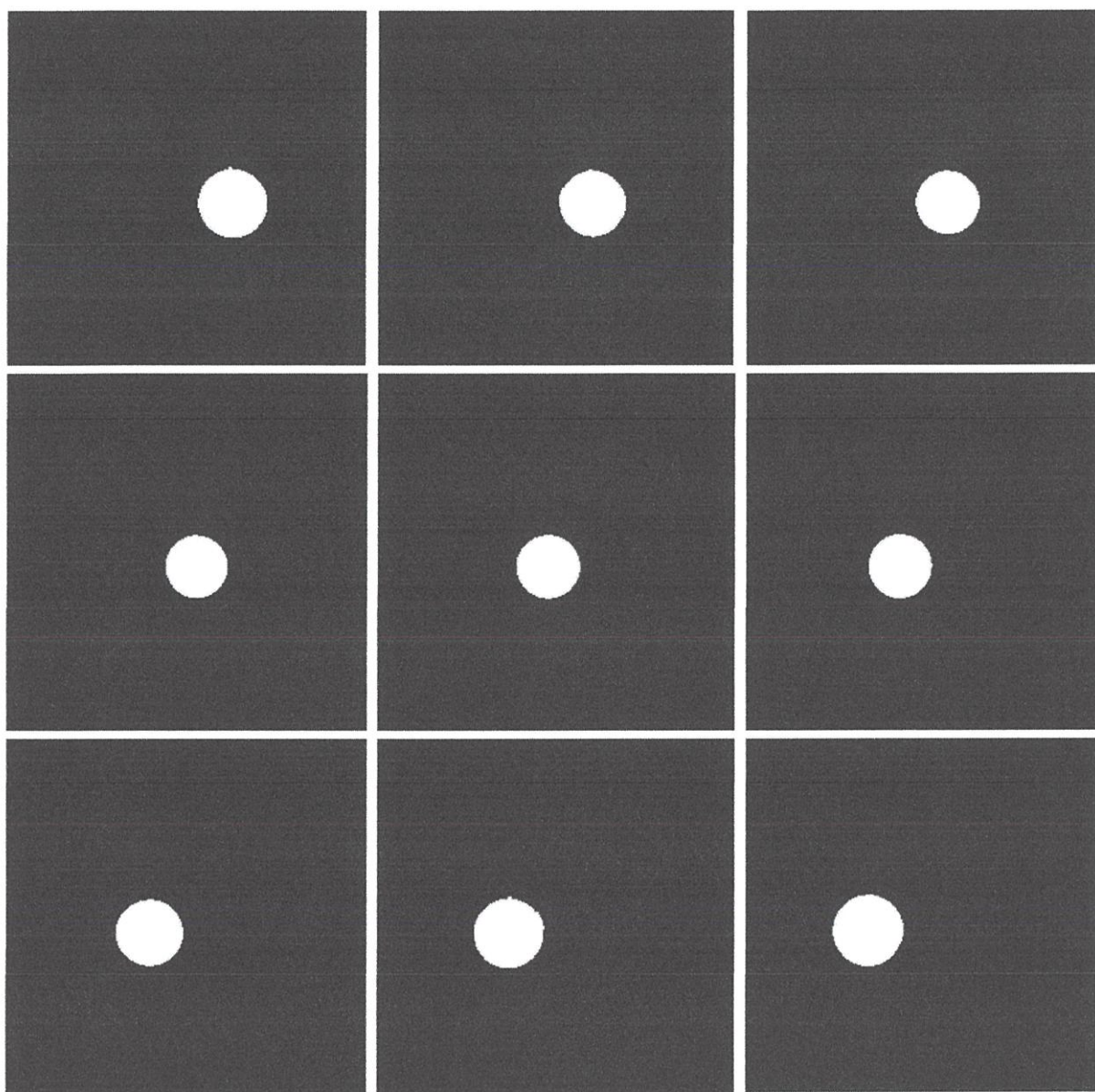


รูปที่ 4.2 ภาพที่ทำการปรับทreshold

ภาพขาวดำ 15 ภาพ จากทั้งหมด 90 ภาพ เป็นภาพที่ปรับค่าแสงให้เห็นเฉพาะแต่วัตถุที่สนใจ เพื่อเตรียมสำหรับการสร้างภาพต่อไปต่อไป โดยวัตถุที่สนใจในที่นี้คือลูกปิงปอง

4.3 ผลการทดลองภาพหลังจากทำการลบภาพพื้นหลัง

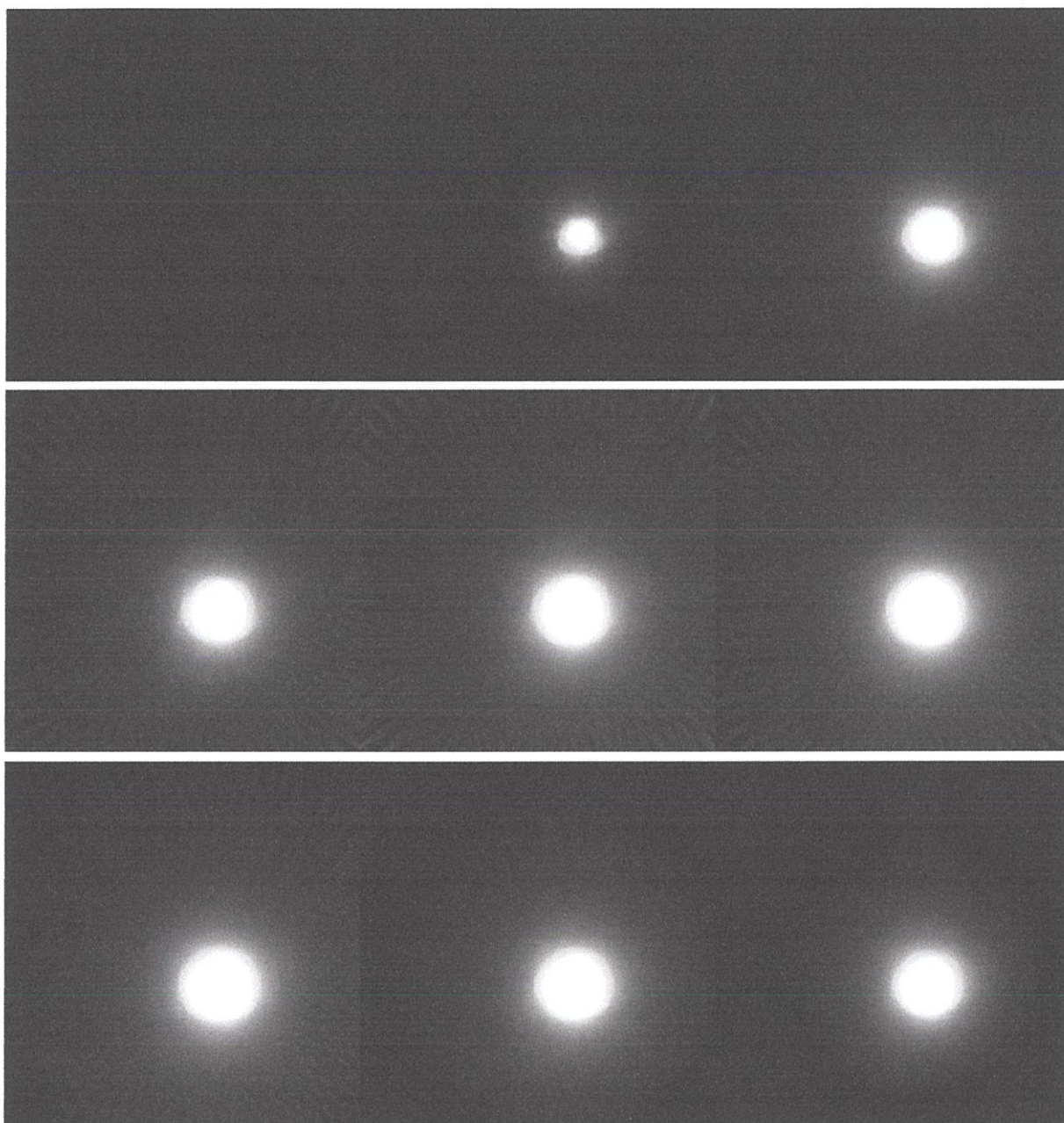




รูปที่ 4.3 ภาพที่ทำการลบพื้นหลัง

เป็นการทำให้ภาพเหลือแต่จุดที่เราสนใจ โดยการลบภาพพื้นหลัง (Background Subtraction) เตรียมความพร้อมที่จะทำการรีคอนสตรัคชันภาพเพื่อทำภาพสามมิติ

4.4 ผลการทดลองการสร้างภาพตัดขวาง

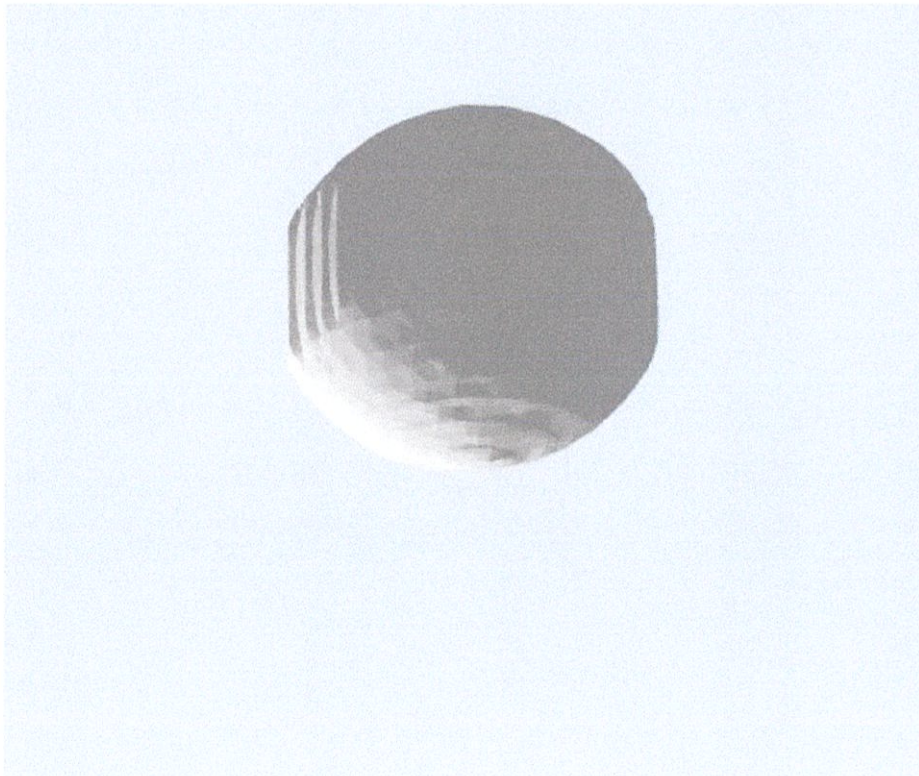




รูปที่ 4.4 ภาพที่ได้หลังจากการทำกรตัดขวาง

ภาพรีคอนสตรักชัน เป็นภาพที่เกิดจากการนำเอาภาพที่เตรียมไว้มาทำเป็นภาพตัดขวาง จากนั้นโปรเจคชันภาพทั้งหมดกลับมา ได้ภาพออกมาเป็นโมเดลสามมิติ

4.5 ผลการทดลองโมเดลสามมิติ



รูปที่ 4.5 ภาพสามมิติจากลูกปิงปอง

เป็นการนำภาพตัดขวางที่ได้มาสร้างเป็นแบบจำลองสามมิติ

4.6 ผลเปรียบเทียบความคลาดเคลื่อนของวัตถุจริงกับวัตถุที่ได้จากการสร้างภาพสามมิติ

จากการเก็บภาพด้วยเครื่องมือที่ได้ออกแบบมา ทำให้ได้ภาพวัตถุและภาพจำลองสามมิติดังผลการทดลองที่กล่าวมาข้างต้น จากนั้นกลุ่มของเราจึงทำการเก็บภาพของวัตถุในรูปแบบอื่นๆ ได้แก่ วัตถุทรงสี่เหลี่ยม และด้ามจับแรสเบอร์รี่พาย โดยสำหรับลูกปิงปอง ได้ทำการวัดความคลาดเคลื่อนโดยการแทนที่ของน้ำ ส่วนวัตถุทรงสี่เหลี่ยม และด้ามจับแรสเบอร์รี่พายได้ใช้เวอร์เนียร์คาลิเปอร์ในการวัด โดยผลความคลาดเคลื่อนแสดงในตารางดังนี้

ตารางที่ 4.1 แสดงผลความคลาดเคลื่อนของวัตถุตัวอย่างได้แก่ ลูกปิงปอง วัตถุทรงสี่เหลี่ยม และด้ามจับแรสเบอร์รี่พาย โดยจะเป็นการเปรียบเทียบโดยการวัดปริมาตรของวัตถุจริงกับวัตถุที่ได้จากการสร้างภาพจำลองสามมิติ

ชนิดของวัตถุ	วัตถุจริง	วัตถุจากการสร้างภาพสามมิติ	ความแม่นยำ
ลูกปิงปอง	เส้นผ่านศูนย์กลาง : 40 มม. ปริมาตร : 10,666.67 π มม. ³	เส้นผ่านศูนย์กลาง : 38 มม.* ปริมาตร : 10,254 π มม. ³ *	เส้นผ่านศูนย์กลาง : 95% ความแม่นยำเชิงปริมาตร : 96.13% *ไม่เป็นทรงกลม
วัตถุทรงสี่เหลี่ยม	ปริมาตร : 25*25*25 มม. ³	ปริมาตร : 23.4*24.1*25.6 มม. ³	ความแม่นยำเชิงปริมาตร : 92.39%
ด้ามจับแรสเบอร์รี่	ปริมาตร : 3.45 ซม. ³	ปริมาตร : 3.37 ซม. ³	ความแม่นยำเชิงปริมาตร : 97.68%
* ทำการวัดด้วยปริมาตรโดยการแทนที่น้ำ รูปร่างของแบบจำลองสามมิติที่ได้มีมิติและรูปร่างต่างกับต้นแบบ			

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

จากผลการสร้างแบบจำลองสามมิติในตารางที่ 4.1 แสดงให้เห็นว่า ผลแบบจำลองสามมิติที่ได้นั้น มีมิติต่างๆ ที่ต่างจากวัตถุจริงอยู่ ค่าความคลาดเคลื่อนทั้งในมิติแกน x , แกน y และแกน z รวมทั้งรูปร่างที่ผิดจากวัตถุจริง อาจเนื่องมาจากตำแหน่งของกล้องที่ยึดติดอยู่บนโมดูล ค่าตัวแปรภายในของกล้อง การส่ายของแกนการหมุน และตัวแปรภายนอกต่างๆ เช่น แสง เงา ทำให้การเรียงของภาพผิดพลาดไป

หลังจากลองปรับแก้ตำแหน่งทางกลศาสตร์แล้ว ผลการทดลองแบบจำลองสามมิตียังคงมีความผิดพลาดอยู่ รวมทั้งลองปรับแก้ในด้านของซอฟต์แวร์ที่นำมาใช้ เพื่อช่วยในการเรียงตัวของภาพถ่าย(Raw Data) ก่อนนำมาสร้างแบบจำลอง ไม่สามารถแก้ปัญหาความผิดพลาดอันเกิดจากการเรียงตัวของภาพที่ไม่สมบูรณ์ได้ เนื่องจากมุมที่ใช้ถ่ายภาพที่ได้จากกล้องนั้น ไม่สามารถทำการแก้ไขได้ เพราะภาพดังกล่าวจะถูกนำมาโปรเจกชัน(Projection) การแก้ไขการเรียงตัวของภาพ ในซอฟต์แวร์จึงไม่สามารถทำได้

ข้อเสนอแนะควรทำการเช็คทุกๆตัวแปรที่เกี่ยวข้อง เนื่องจากค่าความผิดพลาดเพียงนิดเดียว สามารถทำให้แบบจำลองสามมิติที่ได้ มีความคลาดเคลื่อน และต้องทำการตรวจวัดตำแหน่งและขนาดต่างๆทุกครั้ง พร้อมทั้งทำการตรวจเทียบ ภาพจากกล้องแต่ละกล้องให้มีการเรียงตัวของภาพ ตำแหน่ง และองศาที่ตรงกันเปรียบเทียบการใช้กล้องตัวเดียว การแก้ไขด้านซอฟต์แวร์ไม่สามารถทำได้ จึงต้องใช้กลศาสตร์มาแก้ปัญหา

เอกสารอ้างอิง

- [1] P. Debevec, C. Taylor, and J. Malik : “Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In Computer Graphics”, SIGGRAPH 96 Proceedings, August 1996, pages 11–20.
- [2] B. Mok Oh, M. Chen, J. Dorsey, and F. Durand : “Image-based modeling and photo editing. In Computer Graphics”, SIGGRAPH 2001 Proceedings, August 2001, pages 433–442.
- [3] R. Tsai : “A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf cameras and lenses”, IEEE Journal of Robotics and Automation, August 1987, pages 323–344.
- [4] Van Lunteren A, van Lunteren-Gerritsen GH, Stassen HG, Zuithoff MJ : “A field evaluation of armprostheses for unilateral amputees”, Prosthet Orthot Int, 1983, pages 141-51.
- [5] Castleman, K.R. : “Digital Image Processing”, Second ed., 1996, Englewood Cliffs, New Jersey, Prentice-Hall.
- [6] Young, I.T. : “Quantitative Microscopy”, IEEE Engineering in Medicine and Biology, 1996, 15(1): pages 59-66.

ภาคผนวก

โปรแกรมทดสอบการทำงานของเอนโคเดอร์

```

#include <SoftwareSerial.h>

//INPUTENCODERPIN
#define LaserencoderA A0
#define LaserencoderB A1
int encoder0Pos = 0;
int encoder0PinALast = LOW;
int n = LOW;
int m ;
int previousPos = 0;
int sensorValueA = 0;
int sensorValueB = 0;

//reverse
int encoder0Pos2 = 0;
int encoder0PinALast2 = LOW;
int n2 = LOW;
int m2 ;
int previousPos2 = 0;

//MOTORCONNECTEDMOTORSHIELD
int E1 = 3; //PWM Speed control
int M1 = 4; //HIGH FW & LOW BW

//button
char keyboard = ' ';
char reader = ' ';

//Counter && Trigger Part
int counterTrigger = 0;
bool rePosition = true;
bool machineDone = true;

bool communicationLoop = true;
bool subtractionLoop = true;

int digitalTrigger = 1;

```

```
int stepCounter = 0;
int anglePerPi = 36;
int angleCapture = 4;
//void rePosition();
bool encoderRead = false;
int a =0;

SoftwareSerial serialOne(8, 9);

void setup() {
  // initialize the digital pin as an output.
  pinMode(M1, OUTPUT);

  Serial.begin(9600);
  delay(1000);
  Serial.println("Start Engine");

  serialOne.begin(9600);
  serialOne.flush();
}

void loop()
{

  while(!encoderRead){
    unsigned long startTimer = millis();
    anglePerPi = 36;

    int sensorValueA = analogRead(A0);
    if (sensorValueA < 75)
      {
        n = 1;
      }
    else
      {
        n = 0;
      }
  }
}
```

```
int sensorValueB = analogRead(A1);

if (sensorValueB < 135)
{
    m = 1;
}
else
{
    m = 0;
}

if ((encoder0PinALast == LOW) && (n == HIGH))
{
    if (m == LOW)
    {
        encoder0Pos++;
    }
    else
    {
        encoder0Pos--;
        if(encoder0Pos == -1){
            encoder0Pos =+ 2;
        }
    }
}

encoder0PinALast = n;

if(encoder0Pos < anglePerPi)
{
    if(encoder0Pos - previousPos < 1)
    {
        digitalWrite(M1,HIGH);
        analogWrite(E1,255);
    }
    else if(encoder0Pos - previousPos >= 1)
    {
        digitalWrite(M1,HIGH);
        analogWrite(E1,0);
    }
}
```

```

    }

else if(encoder0Pos >= anglePerPi)
{
    digitalWrite(M1,HIGH);
    analogWrite(E1,0);
    Serial.print("Timeeeeeeeeeeeeeeeeeee : ");
    Serial.println( millis() - startTimer);
    encoderRead = true;
    encoder0Pos = 0;
    Serial.println("Out encoderRead");
    delay(1000);
    rePosition = false;

}

} // encoderRead

while(!rePosition){

    //int z = Serial.read();
    sensorValueA = analogRead(A0);
    if (sensorValueA < 40)
    {
        n = 1;
    }
    else
    {
        n = 0;
    }
    sensorValueB = analogRead(A1);

    if (sensorValueB < 170)
    {
        m = 1;
    }
    else
    {

```

```

    m = 0;
}

if ((encoder0PinALast == LOW) && (n == HIGH))
{
    if (m == LOW)
    {
        encoder0Pos--;
        if(encoder0Pos == -1){
            encoder0Pos =+ 2;
        }
    }
    else
    {
        encoder0Pos++;
    }
}

encoder0PinALast = n;

if(encoder0Pos < 36)
{
    if(encoder0Pos-previousPos < 1)
    {
        digitalWrite(M1,LOW);
        analogWrite(E1,255);
    }
    else if(encoder0Pos-previousPos>=1)
    {
        digitalWrite(M1,LOW);
        analogWrite(E1,0);
    }
    previousPos = encoder0Pos;
}
else if(encoder0Pos>= 36)
{
    digitalWrite(M1,LOW);
    analogWrite(E1,0);
    digitalTrigger = 1; //switch to capobject
}

```

```

    }
}
    digitalTrigger = 0;

}
}

```

โปรแกรมการอ่านเอนโคเดอร์และส่งสัญญาณเพื่อทำการถ่ายภาพของอาตุยโน

```

#include <SoftwareSerial.h>
//INPUTENCODERPIN
#define LaserencoderA A0
#define LaserencoderB A1
int encoder0Pos = 0;
int encoder0PinALast = LOW;
int n = LOW;
int m ;
int previousPos = 0;
int sensorValueA = 0;
int sensorValueB = 0;

//Encoder Threshold Calibration
int sensorMinA = 1023;
int sensorMinB = 1023;
int sensorMaxA = 0;
int sensorMaxB = 0;
int calibrationA = 1;
int calibrationB = 1;
int thresholdA;
int thresholdB;

//MOTORCONNECTEDMOTORSHIELD
int E1 = 3; //PWM Speed control
int M1 = 4; //HIGH FW & LOW BW

//button
char keyboard = ' ';

```

```
char reader = ' ';

//Counter && Trigger Part
int counterTrigger = 0;
bool rePosition = true;
bool machineDone = true;
bool firstCap = false;
bool communicationLoop = true;
bool subtractionLoop = true;

int digitalTrigger = 1;

int stepCounter = 0;
int anglePerPi = 36;
int angleCapture = 4;
//void rePosition();
bool encoderRead = false;
//

SoftwareSerial serialOne(8, 9);

void setup(){

  // initialize the digital pin as an output.
  pinMode(M1, OUTPUT);
  Serial.begin(9600);
  delay(1000);

  //LED Start Cal.
  pinMode(13, OUTPUT);
  digitalWrite(13, HIGH);

  if( calibrationA == 0){
    //forward motor for 2 sec
    digitalWrite(M1,HIGH);
    analogWrite(E1,255);

    Serial.println("Calibration Sensor A");
```

```

Serial.println("Forward A");

unsigned long startTimer1 = millis();

while(millis() - startTimer1 < 2000){ // for sensor A

    sensorValueA = analogRead(A0);

    //Record the maximum sensor value
    if(sensorValueA < sensorMaxA){
        sensorMaxA = sensorValueA;
    }
    //Record the minimum sensor value
    if(sensorValueA > sensorMinA){
        sensorMinA = sensorValueA;
    }
}

thresholdA = sensorMinA + 50;
Serial.print("Threshold A :");
Serial.println(thresholdA);

digitalWrite(M1,HIGH);
analogWrite(E1,0);

unsigned long startTimer2 = millis();
Serial.println("Reverse");
while(millis() - startTimer2 <2000){
    digitalWrite(M1,LOW); //reverse
    analogWrite(E1,255);
}

digitalWrite(M1,LOW);
analogWrite(E1,0);
calibrationA = 1;

}

if( calibrationB == 0){

```

```
//forward motor for 2 sec
digitalWrite(M1,HIGH);
analogWrite(E1,255);

Serial.println("Calibration Sensor B");
Serial.println("Forward B");

unsigned long startTimer3 = millis();
while(millis() - startTimer3 < 2000){ // for sensor A
  sensorValueB = analogRead(A1);

  //Record the maximum sensor value
  if(sensorValueB < sensorMaxB){
    sensorMaxB = sensorValueB;
  }
  //Record the minimum sensor value
  if(sensorValueB < sensorMinB){
    sensorMinB = sensorValueB;
  }
}

thresholdB = sensorMinB + 50;

Serial.println("Threshold B :");
Serial.println(thresholdB);

digitalWrite(M1,HIGH);
analogWrite(E1,0);

unsigned long startTimer4 = millis();
Serial.println("Reverse 2");

while(millis() - startTimer4 < 2000){
  digitalWrite(M1,LOW); //reverse
  analogWrite(E1,255);
}

digitalWrite(M1,LOW);
```

```
    analogWrite(E1,0);
    calibrationB = 1;

}

    serialOne.begin(9600);
    serialOne.flush();

    Serial.println("Type 'S' to Start Program");
}

void loop()
{
    keyboard = Serial.read();
    if(keyboard == 'S'){
        digitalTrigger = 1;

        while(!encoderRead){

            if(!firstCap){ //First Capture
                if(digitalTrigger == 1){

                    while(counterTrigger == 0){
                        if(!serialOne.write('L')) Serial.println("Sending Failed");
                        else{
                            counterTrigger = 1;
                        }
                    }
                }

                while(counterTrigger == 1){
                    serialOne.flush();

                    if(serialOne.available()){
                        reader = serialOne.read();

                        if(reader == 'L'){
                            counterTrigger = 0;
                            firstCap = true;
                        }
                    }
                }
            }
        }
    }
}
```

```
        }
    }
}
}
anglePerPi = 36;
int sensorValueA = analogRead(A0);
if (sensorValueA < 75)
{
    n = 1;
}
else
{
    n = 0;
}
int sensorValueB = analogRead(A1);

if (sensorValueB < 135)
{
    m = 1;
}
else
{
    m = 0;
}

if ((encoder0PinALast == LOW) && (n == HIGH))
{
    if (m == LOW)
    {
        encoder0Pos++;
    }
    else
    {
        encoder0Pos--;
        if(encoder0Pos == -1){
            encoder0Pos =+ 2;
        }
    }
}
```

```
}

encoder0PinALast = n;

if(encoder0Pos < anglePerPi)
{
  if(encoder0Pos - previousPos < 1)
  {
    digitalWrite(M1,HIGH);
    analogWrite(E1,255);
  }
  else if(encoder0Pos - previousPos >= 1)
  {
    digitalWrite(M1,HIGH);
    analogWrite(E1,0);
    stepCounter++;

    if(stepCounter == angleCapture){

      if(digitalTrigger == 1){
        subtractionLoop = false;
      }

    }

    if(!subtractionLoop){

      while(counterTrigger == 0){
        if(!serialOne.write('L')) Serial.println("Sending Failed");
        else{
          counterTrigger = 1;
        }
      }

      while(counterTrigger == 1){
        serialOne.flush();

        if(serialOne.available()){
```

```

        reader = serialOne.read();

        if(reader == 'L'){
            Serial.println((char)reader);
            counterTrigger = 0;
            stepCounter = 0;
            subtractionLoop = true;
        }
    }
}

previousPos = encoder0Pos;

}

else if(encoder0Pos >= anglePerPi)
{
    digitalWrite(M1,HIGH);
    analogWrite(E1,0);
    encoderRead = true;
    encoder0Pos = 0;
    delay(1000);
    rePosition = false;

}

} // encoderRead

while(!rePosition){

    sensorValueA = analogRead(A0);
    if (sensorValueA < 40)
    {
        n = 1;
    }
    else

```

```
{
  n = 0;
}
sensorValueB = analogRead(A1);

if (sensorValueB < 170)
{
  m = 1;
}
else
{
  m = 0;
}

if ((encoder0PinALast == LOW) && (n == HIGH))
{
  if (m == LOW)
  {
    encoder0Pos--;
    if(encoder0Pos == -1){
      encoder0Pos =+ 2;
    }
  }
  else
  {
    encoder0Pos++;
  }
}

encoder0PinALast = n;

if(encoder0Pos < 36)
{
  if(encoder0Pos-previousPos < 1)
  {
    digitalWrite(M1,LOW);
    analogWrite(E1,255);
  }
  else if(encoder0Pos-previousPos>=1)
```



```
        if(serialOne.available()){
            reader = serialOne.read();

            if(reader == 'H'){
                counterTrigger = 0;
                firstCap = true;
            }
        }
    }
}
anglePerPi = 36;
int sensorValueA = analogRead(A0);
if (sensorValueA < 75)
{
    n = 1;
}
else
{
    n = 0;
}
int sensorValueB = analogRead(A1);

if (sensorValueB < 135)
{
    m = 1;
}
else
{
    m = 0;
}

if ((encoder0PinALast == LOW) && (n == HIGH))
{
    if (m == LOW)
    {
        encoder0Pos++;
    }
    else
```

```
{
  encoder0Pos--;
  if(encoder0Pos == -1){
    encoder0Pos =+ 2;
  }
}

encoder0PinALast = n;

if(encoder0Pos < anglePerPi)
{
  if(encoder0Pos - previousPos < 1)
  {
    digitalWrite(M1,HIGH);
    analogWrite(E1,255);
  }
  else if(encoder0Pos - previousPos >= 1)
  {
    digitalWrite(M1,HIGH);
    analogWrite(E1,0);
    stepCounter++;

    if(stepCounter == angleCapture){

      if(digitalTrigger == 2){
        Serial.println("Capture Object");
        communicationLoop = false;
      }
    }
  }

  if(!communicationLoop){

    while(counterTrigger == 0){
      if(!serialOne.write('H')) Serial.println("Sending Failed");
      else{
        Serial.print("Sending Success");
      }
    }
  }
}
```

```

        counterTrigger = 1;
    }
}

while(counterTrigger == 1){
    serialOne.flush();

    if(serialOne.available()){
        reader = serialOne.read();

        if(reader == 'H'){
            counterTrigger = 0;
            stepCounter = 0;
            communicationLoop = true;
        }
    }
}

previousPos = encoder0Pos;

}

else if(encoder0Pos >= anglePerPi)
{
    digitalWrite(M1,HIGH);
    analogWrite(E1,0);
    encoderRead = true;
    encoder0Pos = 0;
    Serial.println("Out encoderRead");
    delay(1000);
    rePosition = false;

}

}

while(!rePosition){

```

```
sensorValueA = analogRead(A0);
if (sensorValueA < 40)
{
  n = 1;
}
else
{
  n = 0;
}
sensorValueB = analogRead(A1);

if (sensorValueB < 170)
{
  m = 1;
}
else
{
  m = 0;
}

if ((encoder0PinALast == LOW) && (n == HIGH))
{
  if (m == LOW)
  {
    encoder0Pos--;
    if(encoder0Pos == -1){
      encoder0Pos =+ 2;
    }
  }
  else
  {
    encoder0Pos++;
  }
}

encoder0PinALast = n;

if(encoder0Pos < 36)
```

```

{
  if(encoder0Pos-previousPos < 1)
  {
    digitalWrite(M1,LOW);
    analogWrite(E1,255);
  }
  else if(encoder0Pos-previousPos>=1)
  {
    digitalWrite(M1,LOW);
    analogWrite(E1,0);
  }
  previousPos = encoder0Pos;
}
else if(encoder0Pos>= 36)
{
  digitalWrite(M1,LOW);
  analogWrite(E1,0);
  digitalTrigger = 1; //switch to capobject
  keyboard = ' ';
  encoder0Pos = 0;
  rePosition = true;
  encoderRead = false;
}
}

digitalTrigger = 0;
firstCap = false;
}
}

```

โปรแกรมถ่ายภาพเมื่อได้รับคำสั่งถ่ายของแรสเบอร์รี่พาย

```

#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <wiringPi.h>
#include <wiringSerial.h>
#include <iostream>
#include </home/pi/raspicam-0.1.1/src/raspicam_cv.h>
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"

```

```
#include <ctime>
#include <sstream>
#include <cstdlib>
#include <time.h>

#define PIN7 4
#define PIN11 17
#define PIN12 18
#define outPin 24
#define LED 0

char High = 'H';

using namespace cv;
using namespace std;

int shot = 18;
int subCapture = 1;

//Image Sequence1
int imagenum=1;
char buff1[100];

//IMS 2
int subtractt=1;
char buffSubtract[100];

int fd;
//bool sendBack();

int cam = 1;
int shotCount = 0;
int subShotCount = 0;
int fileFromCam = 1;
char ct = ' ';

int main(int argc,char **argv)
{
```

```

//setupcam
time_t timer_begin,timer_end;
raspicam::RaspiCam_Cv Camera;
const char* pzOriginalImage = "Original Image";

int nCount= 100;
int iLowH = 0;
int iHighH = 10;

int iLowS = 137;
int iHighS = 255;

int iLowV = 99;
int iHighV = 255;

Mat imgHSV;
Mat imgThresholded;
int c;
cv::Mat image;
cv::Mat subImage;

Mat image1;
Mat image2;

CvScalar result;
Camera.set( CV_CAP_PROP_FORMAT, CV_8UC3 );
Camera.set( CV_CAP_PROP_FRAME_WIDTH,320);
Camera.set( CV_CAP_PROP_FRAME_HEIGHT,240);

clock_t t;

if ((fd = serialOpen("/dev/ttyAMA0", 9600)) < 0)
{
    fprintf (stderr, "Unable to open serial device: %s\n", strerror (errno)) ;
    return 1 ;
}

if (wiringPiSetupGpio () == -1)
{

```

```

fprintf (stdout, "Unable to start wiringPi: %s\n", strerror (errno)) ;
return 1 ;
}

if (!Camera.open()) {cerr<<"Error opening camera"<<endl;return -1;} //prob
printf("Ready to Capture \n");

bool done = false;
int digitalTrigger = 0;
int subtractionImage = 0;
//delay(10000);
serialFlush(fd);

while(true){
delay(500);
while(!done){
printf("Start Progame \n");
while(subtractionImage < 18 && digitalTrigger <18 ){ // big loop depend on frame need

while(serialDataAvail(fd) > 0){

while(subtractionImage < 18){

if(serialGetchar(fd) == 'L'){ // get HIGH from ardu

Camera.grab();
Camera.retrieve(subImage);
sprintf(buffSubtract,"subtractionImage%d.jpg",subtractt++);
cv::imwrite(buffSubtract, subImage);
printf("Subtraction Image has been Captured \n");

printf("send to next raspberry \n ");
delay(10);
serialPutchar(fd,'L');
fflush(stdout);
subtractionImage++;

}

}
}
}
}

```

```

while(digitalTrigger < 18){

    if(serialGetchar(fd) == 'H'){ // get HIGH from ardu

        Camera.grab();
        Camera.retrieve(image);
        sprintf(buff1,"image%d.jpg",imagenum++);
        cv::imwrite(buff1, image);
        printf("Image has been Captured \n");

        printf("send to next raspberry \n ");
        delay(10);
        serialPutchar(fd,'H');
        fflush(stdout);
        digitalTrigger++;
    }
}
}
}
}

```

```

shotCount = 0;
subShotCount = 0;
subtractionImage = 0;
digitalTrigger = 0;
done = true;

}

```

```

done = false;
printf("subtract and dropbox done \n\n");

}
}

```