สำนักหอสมุดกลาง_พระจอมเกล้าลาดกระบัง

ระบบสร้างภาพคอมพิวเตอร์แบบสามมิติโดยอาศัยข้อมูลระยะห่างกล้องกับวัตถุ

3D RECONSTRUCTION FROM RANGE DATA

T061468

นายณัฐพงษ์ ลิ้มประภากรชัย

นายสิริโรจน์ เตชะนรราช

เลขหมู่.....
เลขทะเบียน.....61468
วัน,เดือน,ปี.....18 ก.ค. 2549

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสร้างภาพคอมพิวเตอร์แบบสามมิติโดยอาศัยข้อมูลระยะห่างกล้องกับวัตถุ

3D RECONSTRUCTION FROM RANGE DATA

โดย

นายณัฐพงษ์ ลิ้มประภากรชัย

นายสิริโรจน์ เตชะนรราช

อาจารย์ที่ปรึกษา

ดร.สมศักดิ์ วลัยรัชต์

ดร. อรัญญา วลัยรัชต์

อ.สมเกียรติ วังศิริพิทักษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2547

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบสร้างภาพคอมพิวเตอร์แบบสามมิติโดยอาศัยข้อมูลระยะห่างกล้องกับวัตถุ
3D RECONSTRUCTION FROM RANGE DATA

คณะผู้จัดทำ นายณัฐพงษ์ ลิ้มประภากรชัย รหัส 44010152
นายสิริโรจน์ เตชะนรราช รหัส 44010535

.....อาจารย์ที่ปรึกษา
(ดร. สมศักดิ์ วลัยรัชต์)

.....อาจารย์ที่ปรึกษา
(ดร.อรัญญา วลัยรัชต์)

.....อาจารย์ที่ปรึกษา
(อ.สมเกียรติ วังศิริพิทักษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสร้างภาพคอมพิวเตอร์แบบสามมิติโดยอาศัยข้อมูลระยะห่างกล้องกับวัตถุ

นายณัฐพงษ์ ลิ้มประภากรชัย 44010152
นายสิริโรจน์ เตชะนรราช 44010535
ดร.สมศักดิ์ วลัยรัชต์ อาจารย์ที่ปรึกษา
ดร.อรัญญา วลัยรัชต์ อาจารย์ที่ปรึกษา
อ.สมเกียรติ วังศิริพิทักษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2547

บทคัดย่อ

ระบบสร้างภาพคอมพิวเตอร์แบบสามมิติโดยอาศัยข้อมูลระยะห่างกล้องกับวัตถุ (3D Reconstruction from Range Data) เป็นโครงงานสำหรับการสร้างโมเดลสามมิติในเครื่องคอมพิวเตอร์ จากวัตถุจริง เนื่องจากการถ่ายของวัตถุโดยใช้กล้องทั่วไปนั้น จะทำให้ได้ภาพที่เป็นลักษณะสองมิติ ซึ่งไม่สมจริงและไม่สามารถนำไปใช้งานในแอพพลิเคชัน ที่ต้องการข้อมูลของวัตถุที่เป็นสามมิติได้ ระบบสร้างภาพคอมพิวเตอร์แบบสามมิติโดยอาศัยข้อมูลระยะห่างกล้องกับวัตถุ จึงถูกสร้างขึ้นเพื่อให้สามารถสร้างภาพแบบสามมิติของวัตถุขนาดเล็ก ซึ่งเหมือนกับรูปร่างของวัตถุจริง เพื่อนำภาพแบบสามมิติที่ได้จากระบบนี้ สามารถบันทึกและนำไปใช้ในแอพพลิเคชันทั่วไปได้ โดยจะทำการเก็บภาพของวัตถุจริง ผ่านทางกล้อง เพื่อนำภาพที่ได้จากกล้องเหล่านั้นไปผ่านกระบวนการ Image Processing เพื่อให้ได้ข้อมูลที่ต้องการ แล้วจึงนำข้อมูลที่ได้มาคำนวณ เพื่อหาระยะของแต่ละส่วนของวัตถุ ซึ่งจะ สามารถคำนวณตำแหน่งต่าง ๆ ของเวอร์เท็กซ์เพื่อสร้างภาพแบบสามมิติของวัตถุขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3D Reconstruction from Range Data

Natapong Limprapakornchai	44010152
Siriroj Techanoraraj	44010535
Dr. Somsak Walairacht	Advisor
Dr. Aranya Walairacht	Advisor
Mr. Somkiat Wangsiripitak	Advisor

Academic year 2004

Abstract

3D Reconstruction from Range Data is the project that can construct 3D model in from object in computer. Because of picture that come from camera is 2 dimension picture.2 dimension picture is not realistic and cannot use with other application that need data of object in 3D format. 3D Reconstruction from Range Data was built to construct model of small object that model similar to original object.3D model in this project can save and use with other application .This project take picture of object by camera and use all these picture in image processing to get data that necessary to this project .Then use data to calculate the position of object. Finally we can get vertex data of object that use to build 3D model.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยคำแนะนำ คำปรึกษาและคอยดูแลจากหลายๆฝ่ายด้วยกัน โดยเฉพาะอาจารย์ที่ปรึกษาทั้งสามท่านที่ให้โอกาสให้ข้าพเจ้าได้ทำปริญญานิพนธ์ฉบับนี้ คอยให้ความเอาใจใส่ แนะนำและความช่วยเหลือเสมอมา คือ ดร.สมศักดิ์ วลัยรัชต์ ดร.อรัญญา วลัยรัชต์ และ อาจารย์สมเกียรติ วังศิริพิทักษ์ ซึ่งต้องขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้จัดเตรียมสิ่งอำนวยความสะดวก เพื่อให้การวิจัยและพัฒนาโปรแกรมเป็นไปได้ด้วยความสะดวกและรวดเร็ว รวมทั้งยังมีอินเตอร์เน็ตความเร็วสูงให้บริการ สำหรับการค้นคว้าความรู้ต่างๆ ซึ่งเป็นส่วนหนึ่งที่สำคัญของโครงงานนี้

ขอขอบคุณพี่ๆเพื่อนๆน้องๆในห้องมัลติมีเดียที่คอยสร้างความครึกครื้น คอยช่วยเหลือ เป็นกำลังใจให้อย่างเสมอมา และ ขอบคุณห้องมัลติมีเดียที่ให้อาศัย พักผ่อน ในยามที่อ่อนล้า ให้มีแรงทำงานต่อไป

และสุดท้ายขอขอบคุณบุคคลที่สำคัญที่สุดในชีวิตที่ทำให้ข้าพเจ้ามีวันนี้ได้ ก็คือ บิดา มารดา บุคคลในครอบครัว อันเป็นที่เคารพรัก ซึ่งคอยเลี้ยงดูสั่งสอนข้าพเจ้าเป็นอย่างดี พร้อมให้โอกาส สนับสนุนการศึกษาอย่างเต็มที่ และยังให้กำลังใจ ให้ความรัก เสมอมา ข้าพเจ้าขอกราบขอบพระคุณมา ณ ที่นี้ด้วย

ณัฐพงษ์ ลิ้มประภากรชัย
สิริโรจน์ เตชะนรราช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	**I**
บทคัดย่อภาษาอังกฤษ	**II**
กิตติกรรมประกาศ	**III**
สารบัญ	**IV**
สารบัญตาราง	**VII**
สารบัญภาพ	**VIII**
บทที่ 1 บทนำ	1
1.1 หลักการและเหตุผล	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของระบบ	2
1.4 ประโยชน์ที่คาดว่าจะได้รับจากระบบ	2
บทที่ 2 ทฤษฎีและหลักการฮาร์ดแวร์	3
2.1 ทฤษฎีและหลักการของไมโครคอนโทรลเลอร์(MCS-51)	3
2.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 อนุกรม AT89xx	4
2.1.2 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51	4
2.1.3 หน่วยความจำ (RAM) ภายในของไมโครคอนโทรลเลอร์	7
2.2 ทฤษฎีและหลักการของไดโอดส่องสว่าง (LED)	11
2.3 ทฤษฎีและหลักการของการรับส่งข้อมูลทางพอร์ตอนุกรม(Serial Prot)	11
2.3.1 วิธีการติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์	11
2.3.2 โหมดการติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์	12
2.3.3 การสื่อสารเพื่อเชื่อมต่อไมโครคอนโทรลเลอร์ในการรับส่งข้อมูลทางอนุกรม	14
2.3.4 รีจิสเตอร์ที่ใช้ในการควบคุมและรับส่งข้อมูลของพอร์ตอนุกรม	14
2.4 ทฤษฎีและหลักการของสเตปมอเตอร์	17
2.4.1 วิธีการขับสเตปปิงมอเตอร์ให้หมุนโดยการกระตุ้นเฟส	17
2.4.2 ทฤษฎีและหลักการของสเตปมอเตอร์	19
2.4.3 ข้อดีของเสตปปิงมอเตอร์เมื่อเปรียบกับมอเตอร์กระแสตรง	19
2.4.4 สเตปปิงมอเตอร์แบบยูนิโพลาร์	20
2.4.5 วิธีการตรวจสอบหาเฟสของขดลวดสเตปปิงมอเตอร์	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้าที่
บทที่ 3 ทฤษฎีและหลักการประมวลภาพและการนำภาพมาใช้	22
3.1 วิดีโอโอซีเอ็กซ์	22
3.1.1 ตัวสร้างภาพ(Image Source)	22
3.1.2 การแสดงผลวิดีโอแบบถ่ายทอดสด	23
3.1.3 การประมวลผลภาพ (Image Processing) และ การใช้งานแต่ละพิกเซล (Pixel Access)	23
3.1.4 วิดีโอโอซีเอ็กซ์ทูล (VideoOCXTools)	24
3.1.5 รูปแบบของภาพที่ใช้	24
3.1.6 ทฤษฎีและหลักการการประมวลผลภาพ (Image Processing)	24
บทที่ 4 ทฤษฎีและหลักการการทำสเตรีโอออฟซิส (Stereopsis)	31
4.1 สเตรีโอออฟซิส	31
4.1.1 การหาค่าเริ่มต้นของกล้อง (Calibration)	31
4.1.2 การคำนวนหาระยะห่างของจุด	32
บทที่ 5 ไดเร็กซ์เอ็กซ์	36
5.1 ความรู้เบื้องต้นเกี่ยวกับไดเร็กซ์เอ็กซ์	36
5.2 ชุดพัฒนาซอฟต์แวร์สำหรับไดเร็กซ์เอ็กซ์	36
5.3 คอมโพเน็นท์ไดเร็กซ์เอ็กซ์ 9	37
5.4 ไดเร็กซ์เอ็กซ์ กราฟฟิก (DirectX Graphics)	38
5.4.1 ไดเร็กซ์ทรีดี	38
5.4.2 คุณสมบัติของไดเร็กซ์ทรีดี	38
5.4.3 ไ ดเร็กซ์ทรีดีดีไวซ์(Direct3D Device)	40
5.4.4 ทฤษฎีระบบพิกัด 3 มิติ	41
5.5 กระบวนการเรนเดอร์ออบเจ็กท์ 3 มิติ	45
5.5.1 กระบวนการแปรพิกัดและให้แสง (Transformation and Lighting : T&L)	46
5.5.2 กระบวนการลงสี (Rasterization)	48
5.6 เอ็กซ์ไฟล์	48
5.7 เมช(mesh)	48
5.7.1 สถาปัตยกรรมของเมช	48
5.7.2 ฟังก์ชันของเมช	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้าที่
บทที่ 6 การออกแบบระบบ	51
6.1 การออกแบบระบบโดยรวม	51
6.2 การออกแบบฮาร์ดแวร์	53
6.2.1 ส่วนควบคุมด้วยไมโครคอนโทรลเลอร์ MCS-51	53
6.2.2 ส่วนติดต่อกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม	56
6.2.3 ส่วนหมุนแท่นวางวัตถุด้วยสเตปมอเตอร์	57
6.2.4 ส่วนแหล่งกำเนิดแถบแสง LED	59
6.3 การจัดวางตำแหน่งของฮาร์ดแวร์	60
6.3.1 การจัดวางตำแหน่งของกล้อง	61
6.4 การออกแบบแอพพลิเคชั่น(ส่วนแปลงข้อมูล)	61
6.5 การออกแบบการประมวลผลภาพ	63
6.5.1 การประมวลผลภาพด้วยการเปรียบเทียบส่วนที่แตกต่าง	63
6.5.2 การประมวลผลภาพด้วยการลบภาพ	64
6.5.3 ลดสิ่งรบกวน	64
6.5.4 การคำนวนหาตำแหน่งศูนย์กลาง	64
6.6 สมการสำหรับหาข้อมูลจุดภาพสามมิติของวัตถุ	65
6.7 การออกแบบระบบแสดงผล(ไดเร็กซ์เอ็กซ์)	69
6.7.1 เรนเดอร์(render)	69
6.7.2 โหลดเอ็กซ์ไฟล์	76
6.7.3 เซฟเอ็กซ์ไฟล์	77
บทที่ 7 การทดลอง	78
7.1 การทดลองเพื่อกำหนดค่าเทรชโฮลด์	78
7.2 การทดลองเพื่อหาเวลาที่ใช้ในการสแกนวัตถุ	80
7.3 การทดลองสแกนวัตถุที่มลักษณะพื้นผิวแบบต่างๆ	83
บทที่ 8 บทสรุป	87
8.1 สรุป	87
8.2 ปัญหาและอุปสรรค	87
8.3 แนวทางในการพัฒนาต่อ	88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2-1 เปรียบเทียบคุณสมบัติของไมโครคอนโทรลเลอร์และไมโครโปรเซสเซอร์	3
ตารางที่ 2-2 แสดงบิตต่างๆของรีจิสเตอร์ SCON (serial port control register)	14
ตารางที่ 2-3 แสดงการกำหนดบิต SM0 และ SM1 เพื่อกำหนดโหมดการทำงาน	15
ตารางที่ 2-4 แสดงรีจิสเตอร์ PCON เพื่อกำหนดอัตราการรับส่งข้อมูล	16
ตารางที่ 2-5 แบบฟูลสเตป 1 เฟส	17
ตารางที่ 2-6 แบบฟูลสเตป 2 เฟส	18
ตารางที่ 2-7 แบบฮาล์ฟสเตป 2 เฟส	18
ตารางที่ 7-1 ผลการทดลองเมื่อเปลี่ยนแปลงโหมดการทำงาน	81
ตารางที่ 7-2 ผลการทดลองเมื่อเปลี่ยนแปลงค่าเทรชโฮลด์	81
ตารางที่ 7-3 ผลการทดลองเมื่อเปลี่ยนแปลงความละเอียด	82
ตารางที่ 7-4 ผลการทดลองเมื่อเปลี่ยนแปลงจำนวนสเตป	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2-1 แสดงการทำงานภายในของขาพอร์ต 0	5
รูปที่ 2-2 แสดงการทำงานภายในของขาพอร์ต 1	5
รูปที่ 2-3 แสดงการทำงานภายในของขาพอร์ต 2	6
รูปที่ 2-4 แสดงการทำงานภายในของขาพอร์ต 3	6
รูปที่ 2-5 แสดงแผนภาพการจัดสรรหน่วยความจำ RAM ภายในของ ไมโครคอนโทรลเลอร์	8
รูปที่ 2-6 แสดงรีจิสเตอร์หน้าที่พิเศษใน ไมโครคอนโทรลเลอร์	10
รูปที่ 2-7 แสดงรูปของไดโอดเปล่งแสง	11
รูปที่ 2-8 สัญลักษณ์ของ LED	11
รูปที่ 2-9 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 0	12
รูปที่ 2-10 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 1	13
รูปที่ 2-11 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 2	13
รูปที่ 2-12 แสดงการต่อวงจรขับสเตปปิงมอเตอร์โดยใช้ไอซีสำเร็จรูป และวงจรทรานซิสเตอร์	19
รูปที่ 2-13 สเตปปิงมอเตอร์	20
รูปที่ 2-14 สเตปปิงมอเตอร์ 4 เฟส แบบยูนิโพล่าร์ (Uni-polar stepper motor)	20
รูปที่ 2-15 แสดงการใช้มิเตอร์วัดค่าความต้านทาน	21
รูปที่ 2-16 แสดงการต่อวงจรเพื่อทดสอบโดยการสวิตซ์เพื่อหาลำดับ	21
รูปที่ 3-1 แสดงระบบพิกัด Space	25
รูปที่ 3-2 ภาพแสดงกรณีที่ภาพเป็นโมโนโครม	25
รูปที่ 3-3 กระบวนการทำดิจิตอลอิมเมจ	26
รูปที่ 3-4 กระบวนการสุ่มค่า (Sampling) จากภาพแบบอนาล็อกและทำควอนไทเซชั่น (Quantization)	26
รูปที่ 3-5 ผลลัพธ์หลังผ่านกระบวนการแปลงเป็นดิจิตอลอิมเมจ	27
รูปที่ 3-6 ภาพแสดงเมตริกซ์ที่ใช้เก็บค่าเชิงตัวเลขในดิจิตอลอิมเมจ	27
รูปที่ 3-7 โมเดลสีแบบ RGB (RGB Color Model)	27
รูปที่ 3-8 แสดงโมเดลสีและการผสมสีของโมเดลสีแบบอาร์จีบี	28
รูปที่ 3-9 แสดงการประมวลผลภาพแบบลบภาพ	29
รูปที่ 3-10 แสดงการประมวลผลภาพแบบเทรชโฮลด์ด้วยค่าคงที่ = 100	30
รูปที่ 4-1 แบบจำลองกล้องรูเข็ม	31
รูปที่ 4-2 การคำนวณหาระยะบนภาพ	32
รูปที่ 4-3 มุมมองด้านบนของระบบ Active Stereo system	33
รูปที่ 4-4 มุมมองด้านข้างของระบบ Active Stereo system	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 5-1 แสดงความสัมพันธ์ระหว่างไดเร็กซ์ทรีดีกับระบบคอมพิวเตอร์ 39
รูปที่ 5-2 แสดงสถาปัตยกรรมของไดเร็กซ์ทรีดีดีไวซ์ 40
รูปที่ 5-3 แสดงระบบพิกัด 3 มิติ แบบมือซ้ายและมือขวา 41
รูปที่ 5-4 แสดงรูปลูกบาศก์ที่ประกอบกันขึ้นมาจากรูปสามเหลี่ยม 42
รูปที่ 5-5 แสดงภาพทรงกลมที่ประกอบขึ้นมาจากรูปสามเหลี่ยม 42
รูปที่ 5-6 รูปแบบลำดับเวกเตอร์ในเวอร์เท็กซ์ 43
รูปที่ 5-7 แสดงการให้แสงเงาแบบแบนราบให้กับกาน้ำชา 44
รูปที่ 5-8 แสดงการให้แสงเงาแบบเการดาให้กับกาน้ำชา 45
รูปที่ 5-9 ขั้นตอนในเรนเดอร์ไปป์ไลน์ 45
รูปที่ 5-10 แสดงความสัมพันธ์ระหว่างโมเดลในโมเดลสเปซที่มาอ้างอิงตำแหน่งใหม่ในเวิร์ลสเปซ 46
รูปที่ 5-11 แสดงความสัมพันธ์ระหว่างโมเดลในเวิร์ลสเปซกับมุมมองของกล้อง 46
รูปที่ 5-12 แสดงการทำโปรเจ็กชั่น ซึ่งจะเห็นโมเดลในมุมมองภาพ 2 มิติที่มีความลึก 47
รูปที่ 5-13 หลังจากถูกวิวพอร์ตและขริบออก 47

รูปที่ 6-1 ภาพแสดงขั้นตอนการทำงานของระบบ 52
รูปที่ 6-2 ภาพแสดงการติดต่อในแต่ละส่วน 53
รูปที 6-3 ภาพแสดงกระบวนการทำงานของ MCS-51 54
รูปที่ 6-4 แสดงวงจรแปลงแรงดันไฟ 12 โวลท์เป็น 5 โวลท์ 55
รูปที่ 6-5 ภาพส่วนควบคุมด้วยไมโครคอนโทรลเลอร์ MCS-51 และ
ส่วนติดต่อกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม 55
รูปที่ 6-6 รูปรายละเอียดของ MAX 232 56
รูปที่ 6-7 แสดงการต่อวงจรรับส่งข้อมูลผ่านทางพอร์ตอนุกรม 57
รูปที่ 6-8 วงจรขับสเตปมอเตอร์โดยใช้ ULN 2003 58
รูปที่ 6-9 ภาพส่วนหมุนแท่นวางวัตถุด้วยสเตปมอเตอร์ 58
รูปที่ 6-10 แสดงวงจรภายในส่วนแหล่งกำเนิดแถบแสง LED 59
รูปที่ 6-11 แสดงส่วนหน้าของส่วนแหล่งกำเนิดแถบแสง LED 59
รูปที่ 6-12 แสดงการจัดวางตำแหน่งของฮาร์ดแวร์ 60
รูปที่ 6-13 แสดงการตั้งค่าเริ่มต้นของกล้องบันทึกภาพ 61
รูปที่ 6-14 แสดงการทำงานในส่วนแอพพลิเคชัน 62
รุปที่ 6-15 แสดงถึงตำแหน่งต่างๆในระบบ 66
รูปที่ 6-16 รูปการหมุนของวัตถุ 67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 7-1 รูปผลการทดลองค่าเทรชโฮลด์ 79
รูปที่ 7-2 ผลการทดลองเมื่อนำแก้วใสมาสร้างโมเดล 83
รูปที่ 7-3 ผลการทดลองเมื่อนำแก้วมีหูจับสีน้ำเงินมาสร้างโมเดล 84
รูปที่ 7-4 ผลการทดลองเมื่อนำกล่องอะลูมิเนียมสีแดงมาสร้างโมเดล 84
รูปที่ 7-5 ผลการทดลองเมื่อนำตะกร้าจักสานมาสร้างโมเดล 85
รูปที่ 7-6 ผลการทดลองเมื่อนำแก้วมีหูจับสีขาวมาสร้างโมเดล 86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

1.1 หลักการและเหตุผล

ตั้งแต่ในอดีตมนุษย์เรานั้นต้องการสิ่งที่ดีขึ้นกว่าที่เป็นอยู่จึงทำให้เกิดการพัฒนาเกิดขึ้น ซึ่งจะเห็นได้จากการที่มีเทคโนโลยีใหม่ๆ เพื่อแต่ละจุดประสงค์ออกไป ซึ่งการบันทึกภาพให้เสมือนของจริงนั้นก็เป็นจุดประสงค์หนึ่ง ตั้งแต่ในอดีต ที่มนุษย์เราพยายามที่จะวาดภาพเพื่อเป็นการบันทึกรายละเอียดสิ่งของจนกระทั่งมีกล้องถ่ายรูป แต่เนื่องจากเป็นรูปที่มีเพียงแค่สองมิติ จึงทำให้ไม่สามารถเห็นรายละเอียดของวัตถุได้ทั้งวัตถุ จึงทำให้ต้องเปลืองทรัพยากร เช่น การที่ต้องถ่ายรูปของวัตถุในหลายๆด้าน ต่อมาจึงมีการสร้างโมเดลจำลองของวัตถุขึ้นมาด้วยวัสดุต่างๆ เช่น พลาสติก แต่ก็ยังมีความยุ่งยากในการผลิตโมเดลจำลองออกมา แต่ด้วยความก้าวหน้าของเทคโนโลยีคอมพิวเตอร์ทำให้เราสามารถสร้างแบบจำลองของวัตถุ หรือ โมเดลขึ้นมาได้โดยใช้โปรแกรมทางด้านสามมิติ เช่น ทรีดีแม็กซ์ (3Ds Max) แต่ก็ยังมีปัญหาในเรื่องความซับซ้อนยุ่งยากของการใช้โปรแกรมเหล่านี้ อีกทั้งต้องใช้ความชำนาญเพื่อที่จะสร้างโมเดลของวัตถุออกมาให้เหมือนจริง ภายหลังจึงมีความพยายามที่จะผลิตเครื่องมือออกมาเพื่อลดความยุ่งยากในการสร้างโมเดลของวัตถุ แต่ก็ยังมีปัญหาอื่นๆอีก เช่น ความยุ่งยากในการผลิตเครื่องมือ,ความเหมือนจริงของรูปร่างวัตถุที่ได้,ราคาของเครื่องมือที่ใช้ ซึ่งหากเราสามารถที่จะพัฒนาเทคโนโลยีทางด้านนี้ได้ ก็จะทำให้เกิดความสะดวกเนื่องจากเราสามารถที่จะใช้โมเดลของวัตถุ ในการนำเสนออุปกรณ์ผ่านทางอินเตอร์เน็ตโดยที่เราไม่จำเป็นต้องนำอุปกรณ์จริงๆไป

เนื่องจากในปัจจุบันเครื่องมือต่างๆนั้นมีความหลากหลายและมีราคาถูก ซึ่งทำให้เราสามารถที่จะนำอุปกรณ์เหล่านั้นมาประยุกต์ได้ แต่การจะนำอุปกรณ์ใดมาใช้นั้นจำเป็นต้องมีการค้นคว้าทดลองนำเอาหลักการและ ทฤษฎีทั้งหลายมา ประยุกต์เพื่อให้ได้เป็นเครื่องมือที่ตรงตามจุดประสงค์ คือ มีราคาที่ถูก, ได้โมเดลของวัตถุที่เหมือนจริง และ สะดวกต่อการใช้งาน จึงได้มีการจัดทำโครงงานนี้ขึ้นมา

1.2 วัตถุประสงค์

1) นำวัตถุจริงมาสร้างเป็นภาพ 3 มิติ ในคอมพิวเตอร์

2) สามารถบันทึกข้อมูลเพื่อนำไปเปิดในแอพพลิเคชันทั่วไปได้

3) ศึกษาการทำงานของการประมวลผลภาพ

4) ศึกษาทฤษฎีในการคำนวณหาตำแหน่งของ จากภาพของวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของระบบ

1) สามารถสร้างภาพสามมิติได้จากวัตถุที่มีขนาดไม่เกิน 20 x 20 x 20 เซนติเมตร

2) วัตถุที่นำมาใช้นั้นจะต้องมีรูปร่างที่ไม่ซับซ้อนจนเกินไป มีลักษณะเป็นคอนเว็กซ์ (convex) คือ ไม่มีส่วนเว้า หรือส่วนที่เป็นคอนเคฟ (concave)

3) สามารถสร้างโมเดลเฉพาะส่วนพื้นผิวภายนอกของวัตถุ ไม่สามารถส่วนภายในวัตถุซึ่งมีการบดบังเกิดขึ้นได้

4) สามารถบันทึกโมเดลของวัตถุที่สร้างนั้น เพื่อนำไปเปิดใช้ในโปรแกรมที่เกี่ยวกับการสร้างภาพสามมิติอื่นๆ เช่น ทรีดีแม็กซ์ เพื่อปรับปรุงให้ดีขึ้นได้

5) สามารถเปิดดูโมเดลที่ถูกบันทึกโดยโปรแกรมสามมิติอื่นๆได้ แต่ต้องถูกบันทึกมาในรูปแบบไดเร็กเอ็กซ์ เอ็กซ์ไฟล์ (.x File)

1.4 ประโยชน์ที่คาดว่าจะได้รับจากระบบ

1) สามารถที่จะสร้างโมเดลของวัตถุได้ง่าย เนื่องจากการสร้างโมเดลสามมิติออกมาไม่จำเป็นต้องใช้คนที่มีความชำนาญวาดภาพผ่านทางโปรแกรมแอพพลิเคชันต่างๆ เช่น ทรีดีแม็กซ์ (3ds Max)

2) ใช้เวลาน้อยในการสร้างโมเดลของวัตถุ เมื่อเทียบกับการทำโมเดลขึ้นเองโดยใช้โปรแกรมอื่นๆ เนื่องจากวิธีการใช้เครื่องมือนี้ง่าย ไม่จำเป็นต้องมีความชำนาญ

3) สามารถนำโมเดลไปใช้ในการนำเสนอได้ เนื่องจากตัวโมเดลเป็นลักษณะสามมิติ ที่มีลักษณะใกล้เคียงจากรูปร่างของวัตถุจริง ทำให้คนทั่วไปเข้าใจถึงรูปร่างของวัตถุได้ดี มากกว่าการใช้เพียงรูปภาพสองมิติ

4) ไฟล์โมเดลสามารถนำไปใช้ร่วมกับโปรแกรมอื่นได้ ทำให้มีความง่ายต่อการเปิดดู ตกแต่งแก้ไข โดยใช้ฟังก์ชันของโปรแกรมเหล่านั้น เพื่อทำให้โมเดลของวัตถุดีขึ้น เช่น โมเดลมีความละเอียดมากยิ่งขึ้น, มีความถูกต้องมากยิ่งขึ้น เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
ทฤษฎีและหลักการของฮาร์ดแวร์

2.1 ทฤษฎีและหลักการของไมโครคอนโทรลเลอร์(MCS-51)

ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งที่รวมเอาหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณเอาท์พุตหน่วยความจำ วงจรกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน

คุณสมบัติ	ไมโครโปรเซสเซอร์	ไมโครคอนโทรลเลอร์
ขนาดของหน่วยประมวลผลกลาง	ไม่น้อยกว่า 8 บิต	ส่วนใหญ่จะมีขนาด 8 บิต
หน่วยคำนวณทางคณิตศาสตร์และลอจิก	มีอยู่ภายใน	มีอยู่ภายใน
วงจรกำเนิดสัญญาณนาฬิกา	มีอยู่ภายใน	มีอยู่ภายใน
การเชื่อมต่อกับหน่วยความจำโปรแกรม	เชื่อมต่อภายนอกเท่านั้น	ใช้ได้ทั้งภายในและภายนอก
การเชื่อมต่อกับหน่วยความจำข้อมูล	เชื่อมต่อภายนอกเท่านั้น	ใช้ได้ทั้งภายในและภายนอก
การเชื่อมต่อกับพอร์ตอินพุตเอาท์พุต	เชื่อมต่อภายนอกเท่านั้น	มีอยู่ภายในและสามารถขยายได้
ไทเมอร์/เคาน์เตอร์	ไม่มีในซีพียูขนาดเล็ก	มีอย่างน้อย 1 ตัวขนาด 8-16 บิต
วอตช์ด็อก ไทเมอร์	ไม่มีในซีพียูขนาดเล็ก	มีอย่างน้อย 1 ตัว
จำนวนขาต่อใช้งาน	ไม่น้อยกว่า 40 ขา	มีตั้งแต่ 8 ขาขึ้นไป

ตารางที่ 2-1 เปรียบเทียบคุณสมบัติของไมโครคอนโทรลเลอร์และไมโครโปรเซสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 อนุกรม AT89xx

1) เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต

2) ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง

3) หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความจำแบบอีอีพรอม (EEROM) เพิ่มเติม

4) ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาท์พุต

5) มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์ (Full duplex)

6) ไทเมอร์/เคาน์เตอร์ขนาด 16 บิตอย่างน้อย 2 ตัว

7) สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท

8) สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์

9) มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ในภายชิป

10) มีวงจรสื่อสารอนุกรมแบบ SPI สำหรับในอนุกรม AT89Sxx

11) มีวอตช์ด็อกไทเมอร์ในตัว สำหรับในอนุกรม AT89Sxx

2.1.2 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

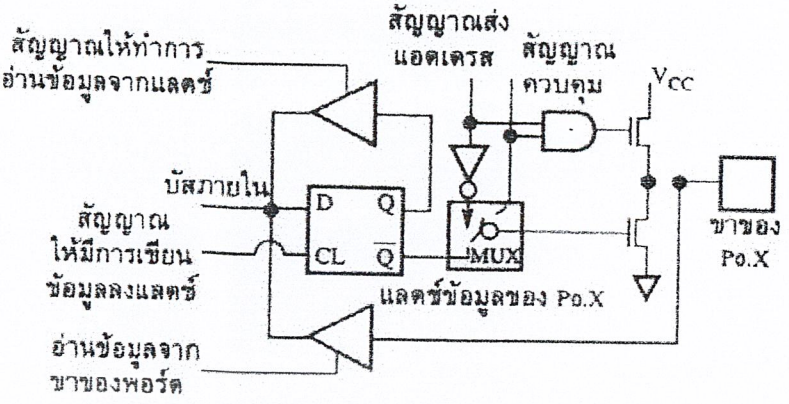
ไมโครคอนโทรลเลอร์ MCS-51 ทุกเบอร์ จะมีสถาปัตยกรรมและขาใช้งานพื้นฐานเหมือนกัน โดยมีขาดังนี้

1) ขา Vcc ใช้สำหรับต่อไฟเลี้ยง +5V

2) ขา GND เป็นขากราวด์สำหรับต่อกับกราวด์ของระบบ

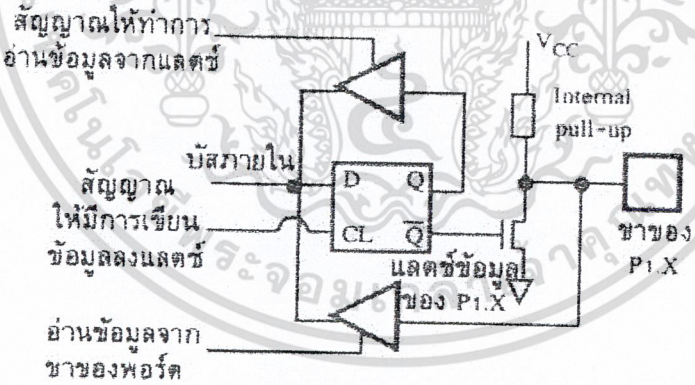
3) ขาพอร์ต 0 (P0.0-P0.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยทางการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดค่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์ (impedance) สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วยเพื่อสลับการทำงานเป็นไปได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-1 แสดงการทำงานภายในของขาพอร์ต 0

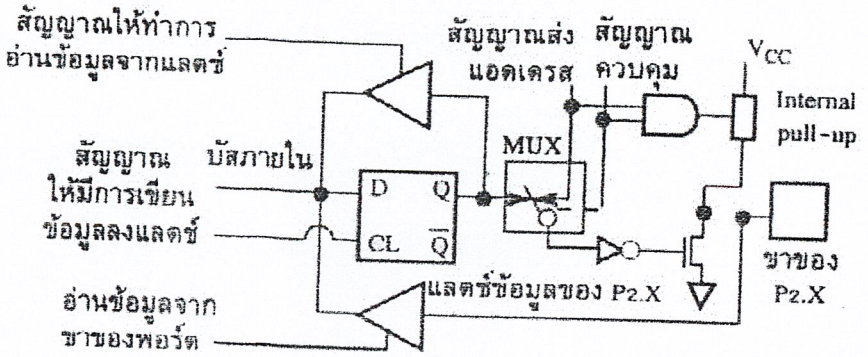
4) ขาพอร์ต 1 (P1.0-P1.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย นอกจากนั้นในอนุกรม AT89Sxx จะใช้ขา P1.0 เป็นขาอินพุตสำหรับนับค่าของไทเมอร์ 2 และ P1.1 เป็นขาอินพุตทริกเกอร์ของไทเมอร์ 2 ในขณะที่ขา P1.4 ถึง P1.7 เป็นขาสำหรับเชื่อมต่อแบบ SPI เพื่อทำการโปรแกรมข้อมูลในระบบ



รูปที่ 2-2 แสดงการทำงานภายในของขาพอร์ต 1

5) ขาพอร์ต 2 (P2.0-P2.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

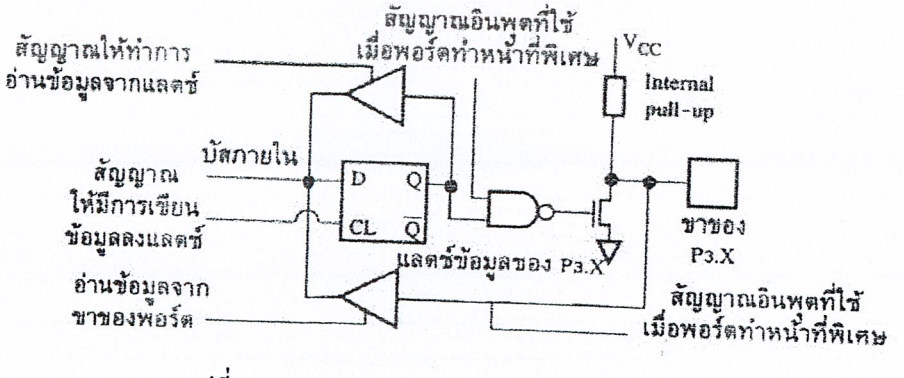
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-3 แสดงการทำงานภายในของขาพอร์ต 2

5) ขาพอร์ต 3 (P3.0-P3.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นต้นต่อไปนี้

P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD
P3.1 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD
P3.2 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 หรือขา INT0
P3.3 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา INT1
P3.4 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0 หรือขา T0
P3.5 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา T1
P3.6 ใช้เป็นขาสัญญาณ WR ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก
P3.7 ใช้เป็นขาสัญญาณ RD ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก



รูปที่ 2-4 แสดงการทำงานภายในของขาพอร์ต 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) ขารีเซต (Reset) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซตสถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชชีนไซเกิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างปกติ

7) ขา ALE/PROG (Address Latch Enable/Program pulse input) เป็นขาที่ใช้ในการควบคุมการแลตช์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีพรอม (EPROM)

8) ขา PSEN (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์ ตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละแมชชีนไซเคิล (machine cycle) แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใดๆ ออกมา

9) ขา EA/Vpp (External Access enable/Programming voltage input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น “0” เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้เป็น “ 1 ” เป็นการ เลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ ที่ขานี้ยังใช้เป็นขาอินพุตสำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม +12V

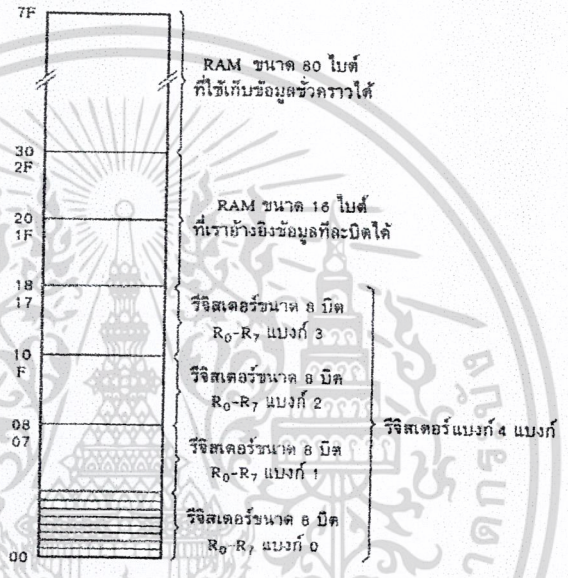
10) ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานไมโครคอนโทรลเลอร์

2.1.3 หน่วยความจำ (RAM) ภายในของ ไมโครคอนโทรลเลอร์

เพื่อให้การเขียนโปรแกรมนั้นเป็นไปได้อย่างคล่องตัว ไมโครคอนโทรลเลอร์ จึงสนับสนุนชุดคำสั่งที่ช่วยให้โปรแกรมเมอร์สามารถทำการอ้างอิงแอดเดรสของหน่วยความจำภายในได้หลายวิธี โดยเราสามารถอ้างอิงหน่วยความจำ 32 ตำแหน่งแรกเป็นรีจิสเตอร์ หรืออ้างอิงเป็นตำแหน่งหนึ่งในหน่วยความจำก็ได้ การเข้าถึงข้อมูลในตำแหน่งที่ถูกอ้างอิงเป็นรีจิสเตอร์ นั้นเราสามารถทำได้โดยใช้คำสั่งที่มีขนาดเพียงแค่ 1 ไบต์ ซึ่งการใช้คำสั่งชนิดนี้จะช่วยให้โปรแกรมเมอร์สามารถเขียนโปรแกรมได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำ ของ ไมโครคอนโทรลเลอร์ นี้จะมีพื้นที่ส่วนหนึ่งขนาด 16 ไบต์ ซึ่งเราสามารถทำการอ้างอิงข้อมูลหน่วยความจำ ส่วนนี้ทีละบิตหรือทีละไบต์ก็ได้ โดยคำสั่งทั่วไปจะสามารถทำการอ้างอิงข้อมูลทีละไบต์จากหน่วยความจำส่วนนี้ แต่ถ้าเราต้องการอ้างอิงข้อมูลทีละบิตในหน่วยความจำส่วนนี้เราจะต้องใช้คำสั่งพิเศษ คำสั่งพิเศษนี้มีประโยชน์มากเมื่อเราต้องการทำการประมวลข้อมูลที่รับมาจากอุปกรณ์ภายนอก เช่น ในงานที่มีการใช้ระบบควบคุม ในหัวข้อต่อไปเราจะพบว่าคำสั่งที่ติดต่อกับ I/O เพื่อทำการเคลื่อนย้ายข้อมูลของ ไมโครคอนโทรลเลอร์ สามารถเคลื่อนย้ายข้อมูลที่มีขนาดเป็นไบต์หรือบิตก็ได้



รูปที่ 2-5 แสดงแผนภาพการจัดสรรหน่วยความจำ RAM ภายในของ ไมโครคอนโทรลเลอร์ ซึ่งเราแบ่งเป็นส่วนที่อ้างอิงเป็น register ได้ 32 ตำแหน่ง ส่วนที่อ้างอิงข้อมูลทีละบิตได้ 128 ตำแหน่ง และส่วนที่เก็บข้อมูลชั่วคราวขนาด 80 byte

รูปที่ 2-5 แสดงแผนภาพการจัดสรรหน่วยความจำภายในช่วง 128 ตำแหน่งแรกของไมโครคอนโทรลเลอร์ จากรูปเราจะพบว่า 32 ตำแหน่งแรกในหน่วยความจำจะถูกกำหนดให้เป็นรีจิสเตอร์แบงค์ (register bank) จำนวน 4 แบงค์ (bank) ในแต่ละแบงค์ จะมี รีจิวเตอร์ 8 ตัว รีจิสเตอร์ในที่นี้เป็นตำแหน่งในหน่วยความจำสำหรับเขียนและอ่านข้อมูลขนาด 8 บิต ซึ่งเราสามารถอ้างอิงได้โดยใช้คำสั่งที่มีขนาด 1 ไบต์ และเนื่องจากว่าคำสั่งขนาด 1 ไบต์ นี้จะใช้บิตเพียงแค่ 3 บิต ในการระบุตำแหน่งของรีจิสเตอร์ ที่เราต้องการดังนั้นเราจึงใช้คำสั่งนี้อ้างรีจิสเตอร์ ที่แตกต่างกันได้เพียง 8 ตัว

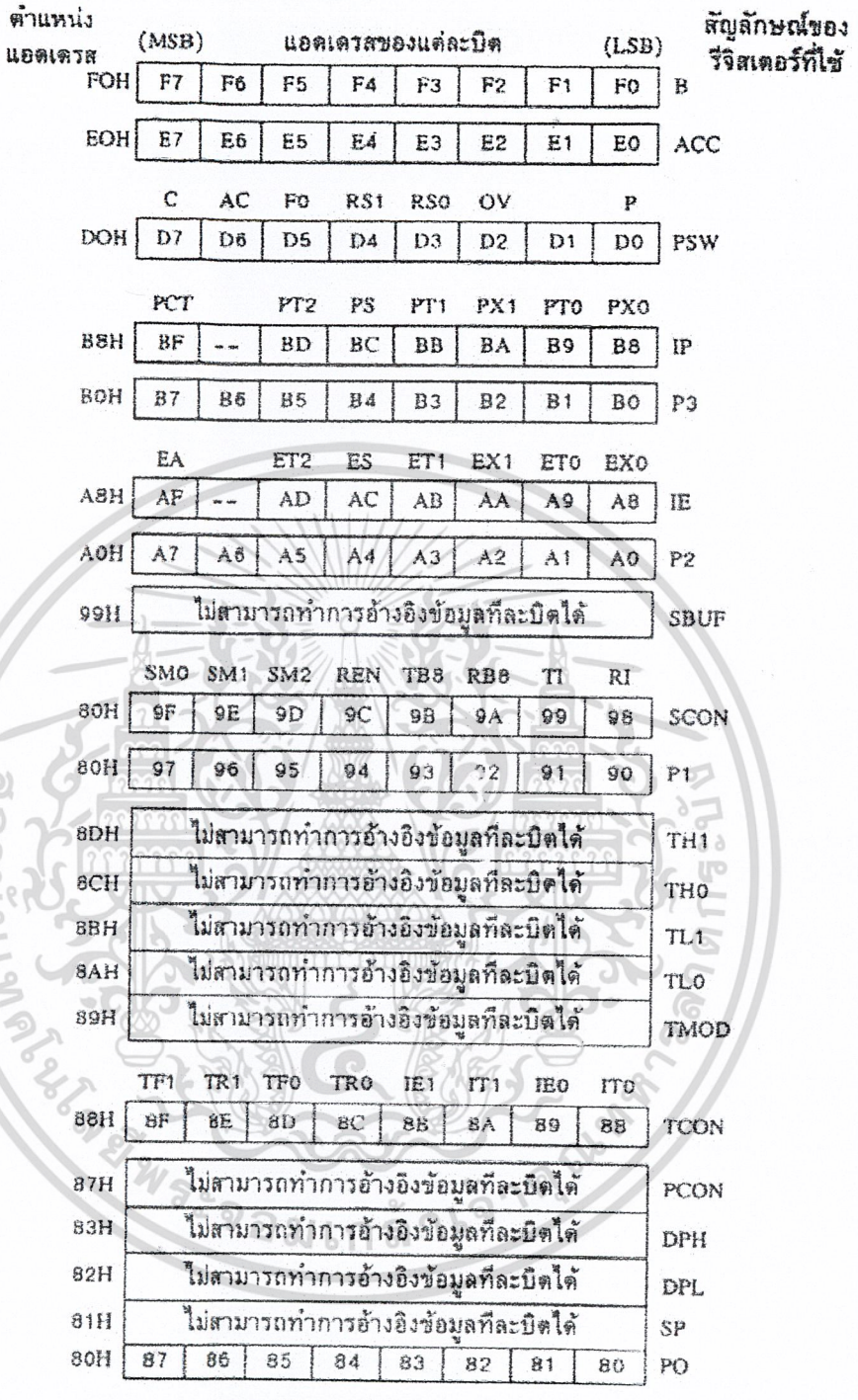
เราจะทราบได้อย่างไรว่าแบงค์รีจิสเตอร์แบงค์ใดจะถูกนำมาใช้งาน แบงค์รีจิสเตอร์ที่กำลังใช้งานอยู่ในขณะนั้นจะถูกเลือกโดยการตั้งค่าที่แฟล็กสำหรับเลือกแบงค์เมื่อแบงค์รีจิสเตอร์หนึ่งๆ ถูกเลือกแล้ว การเคลื่อนย้ายข้อมูลจะเกิดขึ้นกับรีจิสเตอร์ R0 และ R7 ของแบงก์นั้น ถ้าเราต้องการเปลี่ยนไปใช้รีจิสเตอร์ในแบงค์อื่น เราต้องเปลี่ยนค่าในแฟล็กสำหรับเลือกแบงค์ก่อนเพื่อเลือกแบงค์ที่จะทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2-5 แสดงพื้นที่ขนาด 16 ไบต์ ในหน่วยความจำตั้งแต่ตำแหน่งที่ 20H ถึง 30H ซึ่งเป็นส่วนของ RAM ที่เราสามารถอ้างอิงข้อมูลเป็นบิตได้ โดยเราสามารถใช้คำสั่งพิเศษเพื่ออ้างอิงข้อมูลในหน่วยความจำส่วนนี้ทีละบิต หรือใช้คำสั่งทั่วไปเพื่ออ้างอิงข้อมูลทีละไบต์ก็ได้ การใช้คำสั่งพิเศษในไมโครคอนโทรลเลอร์ ช่วยให้โปรแกรมเมอร์สามารถทำการคอมพลีเมนต์บิตใดก็ได้หรือทำการแอนด์บิตหนึ่งกับอีกบิตหนึ่งที่อยู่ในแอคคิวมูเลเตอร์ (accumulator) หรืออยู่ในรีจิสเตอร์สถานะเป็นต้น

พื้นที่ส่วนสุดท้ายในหน่วยความจำที่มีขนาด 80 ไบต์ นั้นมีชื่อว่า สแครชแพ็ด (scratchpad) ซึ่งโปรแกรมเมอร์สามารถนำข้อมูลที่เปลี่ยนแปลงได้ไปเก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

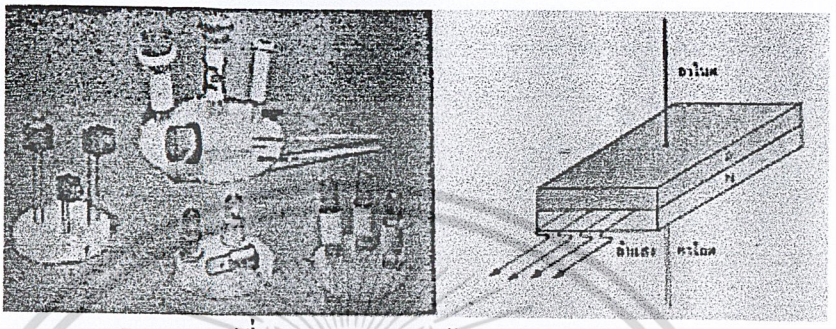


รูปที่ 2-6 แสดงรีจิสเตอร์หน้าที่พิเศษใน ไมโครคอนโทรลเลอร์ รีจิสเตอร์ขนาด 8 บิตทั้ง 21 ตัวนี้จะมีหน้าที่พิเศษเฉพาะของมัน แสดงตารางจัดสรรหน่วยความจำส่วนที่เป็นรีจิสเตอร์หน้าที่พิเศษใน ไมโครคอนโทรลเลอร์ รีจิสเตอร์แต่ละตัวจะมีหน้าที่พิเศษเพียงหน้าที่เดียวเท่านั้น ซึ่งส่วนใหญ่จะเกี่ยวกับการทำงานของ I/O

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

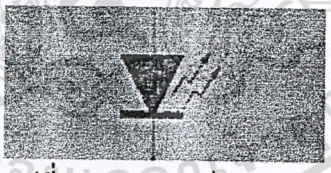
2.2 ทฤษฎีและหลักการของไดโอดส่องสว่าง (LED)

ไดโอดเปล่งแสง (Light-emitting diode) เรียกย่อ ๆ ว่า LED คือ ไดโอดซึ่งสามารถเปล่งแสงออกมาได้แสงที่เปล่งออกมาประกอบด้วยคลื่นความถี่เดียวและเฟสต่อเนื่องกัน ซึ่งต่างกับแสงธรรมดาที่ตาคนมองเห็น อันประกอบด้วยคลื่นซึ่งมีเฟสและความถี่ต่าง ๆ ดันมารวมกัน ไดโอด ซึ่งสามารถให้แสงออกมาได้ ทั้งชนิดที่เป็นสารกึ่งตัวนำของเหลวก๊าซ ในที่นี้จะกล่าวถึงชนิดที่เป็นสารกึ่งตัวนำเท่านั้น



รูปที่ 2-7 แสดงรูปของไดโอดเปล่งแสง

ไดโอดชนิดนี้เหมือนไดโอดทั่ว ๆ ไปที่ประกอบด้วยสารกึ่งตัวนำชนิด P และ N ประกบกันมีผิวข้างหนึ่งเรียบเป็นมันคล้ายกระจก เมื่อไดโอดตกไบแอสตรงจะทำให้อิเลคตรอนที่สารกึ่งตัวนำชนิด N มีพลังงานสูงขึ้นจนสามารถวิ่งข้ามรอยต่อไปรวมกับโฮลใน P ต่อให้เกิดพลังงานในรูปของประจุโฟตอน ซึ่งจะส่งแสงออกมา การประยุกต์ LED ไปใช้งานอย่างกว้างขวางส่วนมากใช้ในภาคแสดงผล (display unit) LED โดยทั่วไปมี 2 ชนิดใหญ่ ๆ คือ LED ชนิดที่ตาคนเห็นได้กับชนิดที่ตาคนมองไม่เห็นต้องใช้ทรานซิสเตอร์มาเป็นตัวรับแสงแทนตาคน



รูปที่ 2-8 สัญลักษณ์ของ LED

2.3 ทฤษฎีและหลักการของการรับส่งของข้อมูลทางพอร์ตอนุกรม (Serial Port)

2.3.1 วิธีการติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

พอร์ตสื่อสารอนุกรมของไมโครคอนโทรลเลอร์ จะมีโครงสร้างเป็นแบบฟูลดูเพล็ก ซึ่งรับและส่งข้อมูลในเวลาเดียวกันได้ โดยจะมีรีจิสเตอร์ SBUF (Serial Data Buffer) เป็นบัฟเฟอร์สำหรับการรับส่งข้อมูลอนุกรม โดยเริ่มต้นเมื่อมีการเขียนข้อมูลเก็บไว้ในรีจิสเตอร์ SBUF หลังจากนั้นข้อมูลจะถูกจัดการโดยวิธีทางฮาร์ดแวร์ในการเลื่อนบิต เพื่อส่งสัญญาณออกไปภายนอก หลังจากมีการส่งข้อมูลออกไปจนครบแล้ว จึงจะทำการเซตบิตโดยกำหนดค่าของแฟล็ก TI ในรีจิสเตอร์ SCON ให้เป็นสถานะ "1" เพื่อแจ้งว่ารีจิสเตอร์ SBUF ว่างแล้ว และพร้อมที่จะส่งข้อมูลไบต์ต่อไปได้

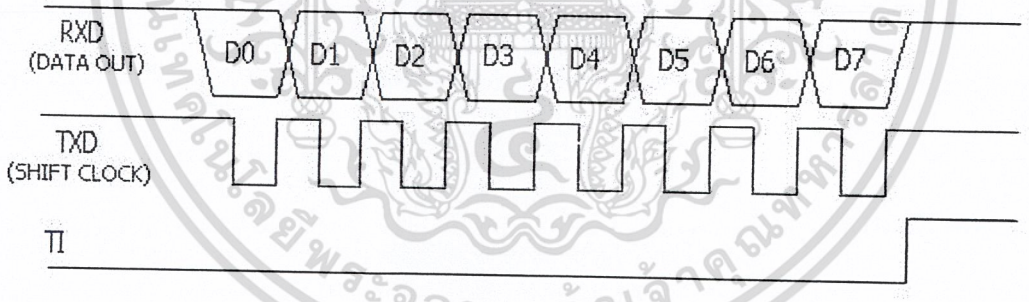
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับข้อมูลจากพอร์ตอนุกรมจะต้องเริ่มต้น โดยการกำหนดค่าของบิต REN ที่อยู่ในใน รีจิสเตอร์ SCON ให้มีค่าเป็นสถานะ “1” หลังจากนั้นเมื่อมีการรับข้อมูลเข้ามาจากภายนอก ก็จะทำการเลื่อนข้อมูลไปโดยอัตโนมัติ และเมื่อบิตสุดท้ายถูกเลื่อนบิตเข้ามาเรียบร้อยแล้ว ข้อมูลจะถูกย้ายมาเก็บไว้ที่รีจิสเตอร์ SBUF และจะทำการเซตที่บิต RI ให้เป็นสถานะ"1" ซึ่งส่งผลให้เกิดการอินเตอร์รัปท์โปรแกรมขึ้น

2.3.2 โหมดการการติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

การสื่อสารอนุกรมของไมโครคอนโทรลเลอร์จะแบ่งออกได้เป็น 4 โหมดด้วยกัน และในแต่ละโหมดจะสามารถสรุปหน้าที่ได้ดังนี้

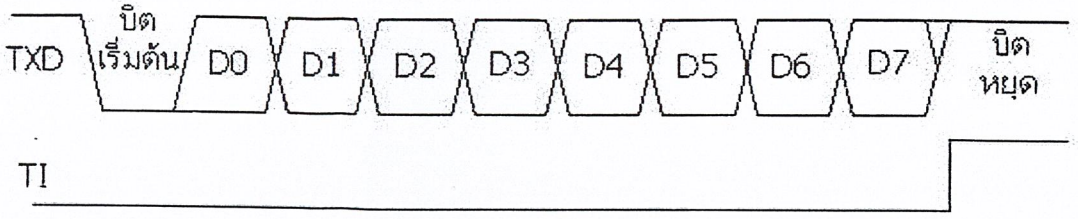
โหมด 0 : จะเป็นการรับส่งข้อมูลขนาด 8 บิตแบบอนุกรม การส่งข้อมูลจะเลื่อนออกไปทีละบิต โดยจะใช้งานของขา RxD เพียงขาเดียว และจะไม่มีการส่งบิตเริ่มต้น (Start bit) ส่วนขา TxD จะใช้เป็นขาของสัญญาณนาฬิกาในการให้จังหวะการเลื่อน ข้อมูลกับวงจรภายนอก (Shift clock) อัตราการรับส่งข้อมูล (Baud rate) จะเป็น 1/12 เท่าของสัญญาณนาฬิกา การรับและส่งข้อมูลจะเริ่มจากบิตต่ำ (LSB) ก่อน ใช้สำหรับเป็นชิพท์รีจิสเตอร์ (Shift Register) จุดประสงค์เพื่อใช้ในการขยายพอร์ตอินพุต หรือพอร์ตเอาต์พุตให้มีจำนวนมาก แต่ในโหมด 0 เรามักจะไม่ค่อยนิยมนำมาใช้งาน เพราะไอซี ไมโครคอนโทรลเลอร์ในปัจจุบันที่เราใช้อยู่นั้น มีจำนวนพอร์ตที่มากพอ และมีไอซีเบอร์อื่นๆ ในตระกูลเดียวกันให้เลือกจำนวนพอร์ตใช้งานมากมายอยู่แล้ว



รูปที่ 2-9 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 0

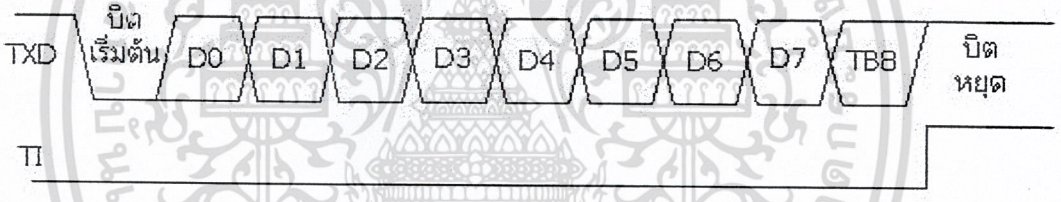
โหมด 1 : จะเป็นการรับและส่งข้อมูลขนาด 10 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) สามารถใช้ในการติดต่อสื่อสารอนุกรมกับมาตรฐานของ RS-232C ของไมโครคอมพิวเตอร์ได้ ซึ่งข้อมูลอนุกรม 10 บิต จะเข้ามาทางขา RXD และ ส่งข้อมูลออกแบบอนุกรมทางขา TXD โดยจะประกอบด้วย 1 บิตแรกเป็นบิตเริ่มต้น (Start bit ค่า 0) 8 บิตต่อมาจะเป็นบิตของข้อมูล (การรับ/ส่งจะเริ่มจากบิตต่ำก่อน) และ บิตหยุดอีก 1 บิต (Stop bit ค่า 1) ส่วนทางด้านรับข้อมูลจะนำค่าบิตหยุด (Stop bit) ที่รับเข้ามาได้นำไปเก็บไว้ในบิต RB8 ที่อยู่ในรีจิสเตอร์ SCON และความเร็วของการส่งข้อมูลในโหมด 1 จะขึ้นอยู่กับบิต SMOD ที่อยู่ในรีจิสเตอร์ PCON และอัตราโอเวอร์โฟลว์ของไทเมอร์ 1 ซึ่งอัตราการรับส่งข้อมูลในโหมดนี้สามารถกำหนดได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-10 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 1

โหมด 2 : จะเป็นการรับและส่งข้อมูลขนาด 11 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) ข้อมูลแบบอนุกรมจะถูกรับเข้ามาทางขา RXD และส่งข้อมูลออกไปทางขา TXD ซึ่งข้อมูล 11 บิตประกอบด้วย บิตแรกจะเป็นบิตเริ่มต้น (Start bit ค่า 0) 9 บิตต่อมาจะเป็นบิตของข้อมูล และบิตสุดท้ายจะเป็นบิตหยุด 1 บิต (Stop bit ค่า 1) สำหรับข้อมูลในบิตที่ 9 จะกำหนดไว้ใน TB8 ที่อยู่ใน รีจิสเตอร์ SCON ซึ่งสามารถกำหนดเป็น 1 หรือ 0 ก็ได้ นิยมนำมาใช้ในการส่งบิตเพื่อตรวจสอบการส่งข้อมูล (Parity bit)



รูปที่ 2-11 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 2

โหมด 3: เป็นการรับส่งข้อมูลแบบ 11 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) เหมือนกับโหมด 2 แต่ในโหมด 3 สามารถกำหนดอัตราความเร็วใน การรับและส่งข้อมูลได้ตามต้องการ

*** UART (Universal Asynchronous Receiver Transmitters) : เป็นการส่งข้อมูลแบบอนุกรม โดยขึ้นอยู่กับความพร้อมของทางด้านส่ง และด้านรับ เป็นการส่งข้อมูลโดยทำการเพิ่มเติมข้อมูลบางอย่างเข้าไป(Start bit ,Stop bit ,Parity bit) เพื่อให้การรับ และการส่งข้อมูลสามารถจะทำงานให้มีความถูกต้องของข้อมูลมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 การสื่อสารเพื่อเชื่อมต่อไมโครคอนโทรลเลอร์ในการรับและส่งข้อมูลทางอนุกรม

มีอยู่ด้วยกัน 2 ระบบคือ

1) Single Processor System คือ ระบบการสื่อสารโดยใช้ไมโครคอนโทรลเลอร์ 2 ตัวเชื่อมต่อหากัน

2) Multiprocessors System คือ ระบบการสื่อสารแบบมัลติโปรเซสเซอร์ โดยใช้ไมโครคอนโทรลเลอร์ 1 ตัวเป็นตัวแม่ (Master) และสามารถที่จะเชื่อมต่อกับไมโครคอนโทรลเลอร์ที่เป็นตัวลูก (Slave) ได้อีกเป็นจำนวนหลายๆ ตัว

2.3.4 รีจิสเตอร์ที่ใช้ในการควบคุม และรับส่งข้อมูลของพอร์ตอนุกรม

รีจิสเตอร์ที่ใช้งานในการติต่อสื่อสารทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ จะประกอบด้วยรีจิสเตอร์ SCON ที่ทำหน้าที่ควบคุมการทำงาน รีจิสเตอร์ SBUF จะใช้ในการเก็บข้อมูลที่จะรับหรือส่ง และรีจิสเตอร์ PCON ซึ่งจะใช้ ในการกำหนดอัตรารับส่ง โดยรีจิสเตอร์ แต่ละตัวจะมีหน้าที่และการทำงานในแต่ละบิตดังต่อไปนี้

รีจิสเตอร์ SCON (Serial Port Control Register) เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ในส่วนของรีจิสเตอร์พิเศษ (Special Function Register) ในตำแหน่งแอดเดรสที่ 98H และสามารถเข้าถึงข้อมูลแบบไบต์ และแบบบิตได้ โดยจะทำหน้าที่ควบคุมการทำงานของพอร์ตอนุกรม การเลือกโหมดการทำงาน และเก็บข้อมูลในบิตที่ 9 (ซึ่งโดยปกติข้อมูลจะมี 8 บิต อยู่ในรีจิสเตอร์ SBUF) ของการรับข้อมูล (RB8) และส่งข้อมูล (TB8) รายละเอียดของแต่ละบิตมีดังต่อไปนี้

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SM0	SM1	SM2	REN	TB8	RB8	T1	R1

ตารางที่ 2-2 แสดงบิตต่างๆของรีจิสเตอร์ SCON (serial port control register)

SM0, SM1 : (Serial port mode bit 0-1) เป็นบิตที่ใช้ในการกำหนดโหมดการทำงานของพอร์ตอนุกรมจำนวน 4 โหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SM0	SM1	โหมด การทำงาน อัตรารับ-ส่ง
0	0	0 Shift register f osc /12
0	1	1 8 bit UART Variable
1	0	2 9 bit UART fosc/32 หรือ fosc/64
1	1	3 9 bit UART Variable

ตารางที่ 2-3 แสดงการกำหนดบิต SM0 และ SM1 เพื่อกำหนดโหมดการทำงาน

SM2 : เป็นบิตที่ทำหน้าที่ควบคุมการทำงานและเลือกลักษณะการเชื่อมต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์ แบบ Single Processor System หรือ Multi processors System โดยกำหนดให้

SM2 =1 เป็นการเลือกแบบ Multi processors System คือระบบการสื่อสารแบบใช้ซีพียูหลายๆตัว ร่วมกันทำงาน จะใช้งานในโหมด 2 หรือโหมด 3

SM2 =0 เป็นการเลือกแบบ Single Processor System โดยสามารถใช้ได้กับทุกโหมด (การใช้งานในโหมด 0 ต้องกำหนดให้ SM2 = 0)

ในกรณีที่เลือกให้ SM2 = 1 แบบ Multi processors System ถ้าข้อมูลที่รับเข้ามาบิตที่ 9 (อยู่ในบิต RB8) มีค่าเป็น "1" ทำให้แฟล็กอินเตอร์รัปต์ทางด้านรับ จะถูกเซตให้เป็น 1 (RI = 1) แต่ถ้าข้อมูลในบิตที่ 9 รับเข้ามามีค่าเป็น "0" จะทำให้แฟล็กอินเตอร์รัปต์ทางด้านรับเป็น 0 (RI = 0)

การทำงาน ในโหมด 1 ถ้าให้ SM2 = 1 แฟลกอินเตอร์รัปต์ทางด้านรับ (แฟล็ก RI) จะไม่ถูกเซต หากข้อมูลที่รับเข้ามาไม่มีบิตหยุด (Stop bit)

REN : (Enable Serial Reception)เป็นบิตที่ควบคุมการรับข้อมูลของพอร์ตอนุกรม กำหนดสถานะของบิตได้โดยซอฟต์แวร์

1 = ให้มีการรับข้อมูล 0 = ไม่ให้มีการรับข้อมูล

TB8 : (Transmit bit D8) เป็นบิตของข้อมูลบิตที่ 9 ในการส่งข้อมูลใช้งานโหมด 2 และโหมด 3 กำหนดสถานะของบิตได้โดยซอฟต์แวร์

RB8 : (Receive bit D8) เป็นบิตของข้อมูลบิตที่ 9 ในการรับข้อมูล โดยใช้งานโหมด 2 และโหมด 3 หากใช้งานในโหมด 1 ถ้ากำหนดให้ SM2 = 0 บิตนี้จะเป็นค่าของ Stop Bit ที่รับเข้ามา สำหรับโหมด 0 จะไม่ใช้งานบิตนี้

TI : (Transmit Interrupt Flag)เป็นบิตที่ใช้งานในการอินเตอร์รัปท์ด้านส่งข้อมูล และจะถูกเซตทางฮาร์ดแวร์เมื่อมีการส่งข้อมูลเสร็จสิ้นลงในบิตที่ 8 ของโหมด 0 (Shift register) หรือเมื่อเริ่มต้นส่ง Stop bit ในโหมด 1,2 หรือ 3 และจะต้องเคลียร์บิตนี้ด้วยซอฟต์แวร์ทุกครั้ง เมื่อโปรแกรมตอบสนองการอินเตอร์รัปท์ ของการส่งข้อมูลเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RI : (Receive Interrupt Flag) เป็นบิตที่ใช้งานในการอินเตอร์รัปท์ทางด้านรับข้อมูล จะถูกเซตทางฮาร์ดแวร์ เมื่อมีการรับข้อมูลเสร็จสิ้นลงในบิตที่ 8 ในโหมด 0 (Shift register) และจะต้องเคลียร์บิตนี้ด้วยซอฟต์แวร์ทุกครั้ง เมื่อโปรแกรมตอบสนองการอินเตอร์รัปท์ของการรับข้อมูลเรียบร้อยแล้ว หรืออาจกล่าวได้ว่าถ้าบิต RI ถูกเซต เมื่อใด หมายถึงข้อมูลได้เข้ามาเก็บไว้ที่รีจิสเตอร์ SBUF จนครบทั้ง 8 บิตแล้ว สามารถที่จะอ่านข้อมูลจากรีจิสเตอร์ SBUF ได้

รีจิสเตอร์ SBUF (serial data buffer register) เป็นรีจิสเตอร์ขนาด 8 บิตหรือ 1 ไบต์มีแอดเดรสอยู่ตำแหน่งที่ 99H และเข้าถึงข้อมูลแบบไบต์ได้อย่างเดียว จะทำหน้าที่รับและส่งข้อมูลออกไปยังพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ ในการอ่านค่าข้อมูลจากภายนอกที่รับเข้ามาทางพอร์ตอนุกรมจะต้องอ่านค่าจากรีจิสเตอร์ SBUF ซึ่งเป็นบัปเฟอร์ (Buffer) ในการเก็บข้อมูลที่รับเข้ามาได้จากภายนอก และในทำนองเดียวกันขณะที่ต้องการส่งข้อมูล เราก็จะนำเอาค่าข้อมูลที่ส่งออกไปไว้ที่ในรีจิสเตอร์ SBUF ก่อนและหลังจากนั้นจึงจะส่งออกไป โดยจะใช้คำสั่งการโอนย้ายข้อมูลแบบไบต์เช่น MOV SBUF,#20H หรือ MOV SBUF,@R1 ก็ได้

การรับข้อมูลในโหมด 0 จะเริ่มต้นรับ เมื่อค่าของบิต RI = 0 และ REN = 1 ส่วนในโหมดอื่นๆ การรับข้อมูลจะเริ่มต้นเมื่อกำหนดบิต REN = 1 และมี Start bit เข้ามาที่ขา RXD

รีจิสเตอร์ PCON (Power Control) เป็นรีจิสเตอร์ขนาด 1 ไบต์มีแอดเดรสอยู่ตำแหน่งที่ 87H เข้าถึงข้อมูลได้แบบไบต์อย่างเดียวเท่านั้น โดยจะประกอบด้วยบิตดังต่อไปนี้

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMOD	-	-	-	GF1	GF0	PD	IDL

ตารางที่ 2-4 แสดงรีจิสเตอร์ PCON เพื่อกำหนดอัตราการรับส่งข้อมูล

PC0N.7 SMOD : ในกรณีที่ใช้ไทม์เมอร์1 เป็นตัวกำหนดอัตรารับส่ง (Baud rate) และหากกำหนดให้บิตนี้มีค่าเป็น "0" ในการใช้งานกับพอร์ตสื่อสารอนุกรมโหมด 1 ,2 และโหมด 3 ค่าอัตรารับส่ง (Baud rate) จะเพิ่มขึ้นเป็นสองเท่า

รีจิสเตอร์ PCON ไม่สามารถอ้างตำแหน่งแบบบิตได้ แต่จะใช้คำสั่งทางลอจิกของการ OR เช่น ORL PCON, #80H จะเป็นการเซตบิตที่ 7 ของรีจิสเตอร์ PCON และการกำหนดให้บิตมีสถานะเป็น "0" หรือเคลียร์บิตจะใช้การ AND เช่น ANL PCON, #01111111B จะเป็นการเคลียร์บิตที่ 7 ของรีจิสเตอร์ PCON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.4 ทฤษฎีและหลักการของสเตปมอเตอร์

2.4.1 วิธีการขับสเตปปิงมอเตอร์ให้หมุนโดยการกระตุ้นเฟส

ในการควบคุมสเต็ปปิงมอเตอร์เพื่อที่จะให้ทำการหมุน มีวิธีการควบคุมกระแสไฟที่จ่ายให้กับขดลวดสเตเตอร์ (Stator) ในแต่ละเฟสของสเตปปิงมอเตอร์ อย่างเป็นลำดับที่แน่นอน โดยถ้าหากเราต้องการให้กระแสไหลในเฟสใดๆ ก็จะทำให้สถานะของเฟสนั้นๆเป็นสถานะลอจิก "1" และในการกระตุ้นเฟสของของสเตปปิงมีอยู่ด้วยกัน 2 แบบคือ

2.4.1.1 การกระตุ้นเฟส แบบฟูลสเตปมอเตอร์ (Full Step Motor) ยังสามารถแบ่งการกระตุ้นเฟสออกได้เป็นอีก 2 วิธีด้วยกันคือ

1) การกระตุ้นเฟสแบบฟูลสเตป 1 เฟส (Single-Phase Driver) หรือแบบเวฟ แสดงดังตารางที่ 2-5 จะเป็นการป้อนกระแสไฟให้กับขดลวด ของสเตปปิ้งมอเตอร์ทีละขด โดยจะป้อนกระแสเรียงตามลำดับกันไป ดังนั้นกระแส ที่ไหลในขดลวด จะทำการไหลในทิศทางเดียวกันทุกขด ลักษณะเช่นนี้จึงทำให้แรงขับของสเตปปิงมอเตอร์มีน้อย

สเตปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

ตารางที่ 2-5 แบบฟูลสเตป 1 เฟส

2) การกระตุ้นเฟสแบบฟูลสเตป 2 เฟส (Two-Phase Driver) แสดงดังตารางที่ 2-6 เป็นการป้อนกระแสให้กับขดลวด 2 ขด ของสเตปปิงมอเตอร์พร้อมๆกันไป และจะกระตุ้นเรียงถัดกับไปเช่นเดียวกับแบบหนึ่งเฟส ดังนั้นการกระตุ้นแบบนี้จึงต้องใช้กำลังไฟมากขึ้น และจะทำให้มีแรงบิดของมอเตอร์มากกว่าการกระตุ้นแบบ 1 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
61468

สเตปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

ตารางที่ 2-6 แบบฟูลสเตป 2 เฟส

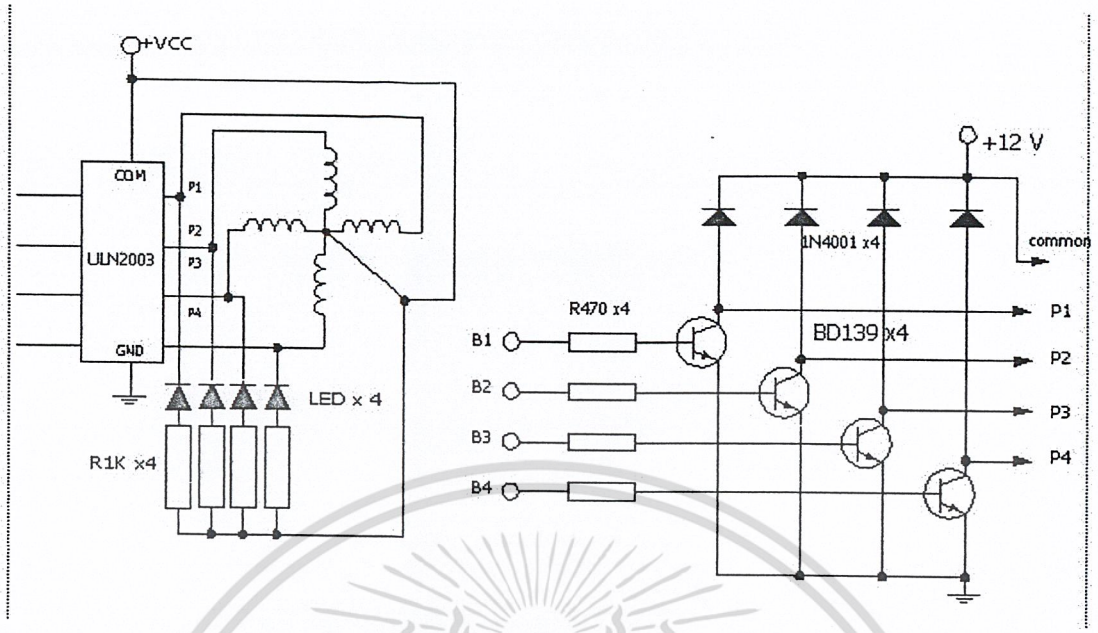
2.4.1.2 การกระตุ้นเฟส แบบฮาล์ฟสเตป (Half Step Motor) หรือ one-two phase Driver คือการกระตุ้นเฟสแบบ ฟูล สเตป 1 เฟส และ 2 เฟส เรียงลำดับกันไป แสดงดังตารางที่ 2-7 แรงบิดที่ได้จากการกระตุ้นเฟสแบบนี้จะมีเพิ่มมากขึ้น เพราะช่วงของสเตปมีระยะสั้นลง ในการกระตุ้นแบบนี้ เราจะต้องมีการกระตุ้นที่เฟสถึง 2 ครั้ง จึงจะได้ระยะของ สเตปเท่ากับการกระตุ้นเพียงครั้งเดียว ของแบบฟูลสเตป 2 แบบแรก ความละเอียดของการหมุนตำแหน่งองศาต่อสเตป ก็เป็นสองเท่าของแบบแรก ความถูกต้องของตำแหน่งที่กำหนดจึงมีมากขึ้น

สเตปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

ตารางที่ 2-7 แบบฮาล์ฟสเตป 2 เฟส

วงจรที่ใช้ในการขับสเตปปิงมอเตอร์โดยใช้ไอซีสำเร็จรูป และวงจรจากทรานซิสเตอร์ แสดงได้ในรูปที่ 2-12 โดย ไอซีสำเร็จรูปเบอร์ ULN2003 จะมีคุณสมบัติเป็นไอซีไดรเวอร์กระแสสูงแบบคอลเล็คเตอร์เปิด สามารถเลือกแรงดันได้กว้าง 5-30 โวลท์ จ่ายกระแสได้สูงถึง 500 มิลลิแอมป์ต่อขา และมีไดโอดที่ป้องกันกระแสย้อนกลับอยู่ภายในไอซี ส่วนแอลอีดีที่ต่อในวงจรเราจะต่อไว้เพื่อแสดงการกระตุ้นแต่ละเฟส ของแต่ละแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-12 แสดงการต่อวงจรขับสเตปปิงมอเตอร์โดยใช้ไอซีสำเร็จรูป และวงจรทรานซิสเตอร์

2.4.2 ทฤษฎีและหลักการของสเตปมอเตอร์

สเตปปิงมอเตอร์เป็นอุปกรณ์เอาต์พุตอย่างหนึ่ง ซึ่งสามารถนำไอซีไมโครคอนโทรลเลอร์ มาทำการควบคุมได้สะดวก และเป็นมอเตอร์ที่เหมาะสมสำหรับใช้ในงานควบคุมการหมุน ที่ต้องการตำแหน่งและทิศทางที่แน่นอน การทำงานของ สเตปปิงมอเตอร์จะขับเคลื่อนทีละขั้น(Step) ขั้นละ 0.9, 1.8, 5, 7.5, 15 หรือ 50 องศา ซึ่งขึ้นอยู่กับคุณสมบัติแต่ละชนิดของสเตปปิงมอเตอร์ตัวนั้นๆ สเตปปิงมอเตอร์จะแตกต่างจากมอเตอร์กระแสตรงทั่วไป (DC MOTOR) โดยการทำงานของมอเตอร์กระแสตรงจะหมุนไปแบบต่อเนื่อง ไม่สามารถหมุนเป็นแบบสเตปๆ ได้ดังนั้นในการนำไปกำหนดตำแหน่งจึงควบคุมได้ยากกว่า แต่ในส่วนใหญ่เราจะใช้สเตปปิงมอเตอร์มาทำการการควบคุมโดยใช้วิธีในระบบดิจิตอล เช่น พรินเตอร์ (Printer) พล็อตเตอร์ (X-Y Plotter) ดิสก์ไดร์ฟ (Disk drive) ฯลฯ

2.4.3 ข้อดีของสเตปปิงมอเตอร์เมื่อเปรียบกับมอเตอร์กระแสตรง (DC MOTOR)

1) การควบคุมไม่ต้องอาศัยตัวตรวจจับการหมุน

2) ไม่ต้องใช้แปรงถ่าน ดังนั้นจึงทำให้ไม่มีส่วนที่จะต้องสึกหรอ และ ปัญหาของการสปาร์ค (ที่เกิดจากหน้าสัมผัสของแปรงถ่านแหวนตัวนำในโรเตอร์) ที่ทำให้เกิดสัญญาณรบกวน

3) การควบคุมโดยทางวงจรดิจิตอลหรือไมโครโพรเซสเซอร์ ทำได้ง่าย และสะดวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

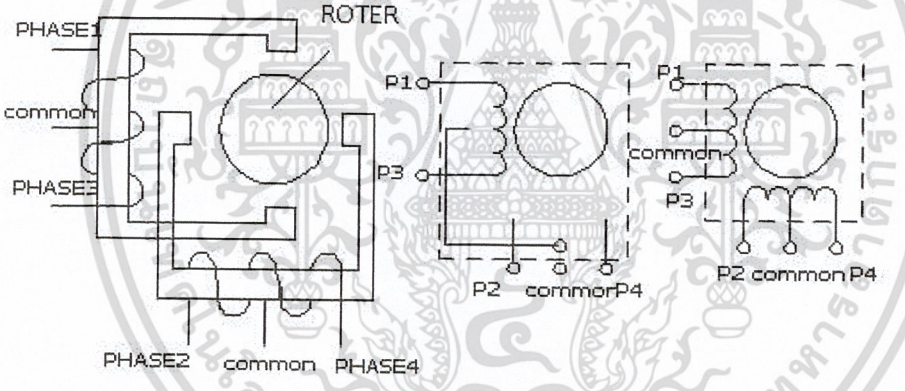
รูปที่ 2-13 สเตปปิงมอเตอร์

2.4.4 สเตปปิงมอเตอร์แบบยูนิโพลาร์ (Uni.-polar stepper motor)

ซึ่งโครงสร้างของสเตปปิงมอเตอร์แบบนี้จะมีส่วนประกอบที่สำคัญ 2 ส่วนด้วยกันคือ

1) ส่วนที่ทำการหมุน (Rotor) จะเป็นแม่เหล็กถาวรหรืออื่นๆ

2) ส่วนที่อยู่กับที่ (Stator) เป็นขดลวดที่พันไว้จำนวนหลายๆขด



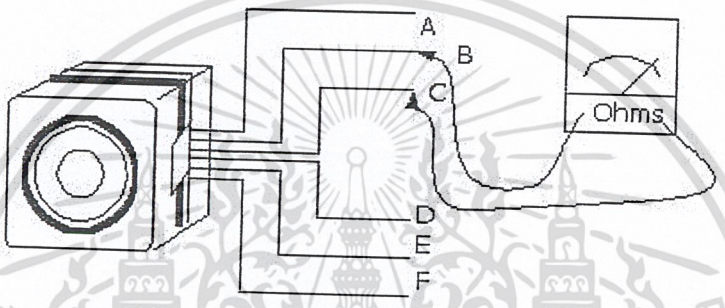
รูปที่ 2-14 สเตปปิงมอเตอร์ 4 เฟส แบบยูนิโพล่าร์ (Uni-polar stepper motor)

2.4.5 วิธีการตรวจสอบหาเฟสของขดลวดสเตปปิงมอเตอร์

ในขั้นตอนที่ 1 ให้สังเกตว่าสเต็ปปิ้งมอเตอร์ที่นำมาทดลองที่เป็นแบบยูนิโพล่าร์ จะมีจำนวนสาย 5 เส้นหรือ 6 เส้น (ถอดจากดิสก์ไดร์ฟเก่าขนาด 5 นิ้วนำมาใช้งานได้)

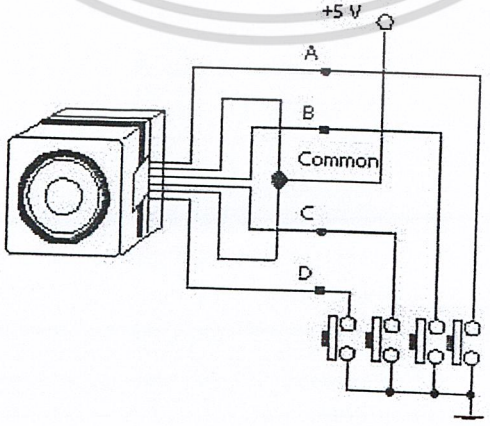
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขั้นตอนที่ 2 ใช้มิเตอร์วัดค่าความต้านทานของเส้นลวดในแต่ละขดดังรูป 2-15 ขั้นตอนการวัดให้หาสายที่ต่อเป็นจุดร่วมเสียก่อน(common) โดยให้ใช้ มัลติมิเตอร์ตั้งค่าไว้สำหรับการวัดค่าความต้านทาน แต่ละเส้น สังเกตที่ค่าความต้านทาน ถ้าหากเราไม่ได้วัดระหว่าง จุดต่อร่วม(common) กับสายแต่ละเส้น ค่าความต้านทานจะมีค่าเป็น 2 เท่าของการวัดระหว่างจุดต่อร่วมกับสายที่ใช้งาน ตัวอย่างเช่น ถ้าให้จุด B เป็นจุดร่วม หากวัดระหว่างที่จุด A กับจุด B จะมีค่าเท่ากับ 60 โอมห์ แต่ถ้าวัดระหว่างที่จุด A และจุด C ซึ่งไม่ใช่จุดร่วมก็จะได้ค่าเท่ากับ 120 โอมห์ หากเป็นแบบที่มีสาย 6 เส้นก็จะมีจุดร่วมสองจุด เพราะมีขดลวดคนละชุดกัน และสายที่เป็นจุดร่วมส่วนใหญ่จะมีสีเหมือนกัน ทำนองเดียวกันหากเป็นแบบที่มีสาย 5 เส้นก็จะมีจุดร่วมเพียงจุดเดียวเท่านั้น



รูปที่ 2-15 แสดงการใช้มิเตอร์วัดค่าความต้านทาน

ในขั้นตอนที่ 3 หากเป็นแบบที่มีสาย 6 เส้นก็ให้ทำการต่อจุดร่วมเข้าด้วยกันจะได้เป็น 5 เส้น แล้วต่อวงจรตาม รูปหลังจากนั้นให้ทดลองกดสวิตซ์ ที่ต่อเข้ากับแต่ละจุดโดยเริ่มที่ จุด A จุด B จุด C และจุด D แล้วให้สังเกตการหมุนของสเตปปิงมอเตอร์ว่าหมุนได้ต่อเนื่องหรือไม่ หากมีการกระโดดข้ามสเตปก็ให้ทดลองโดยเรียงลำดับการกดสวิตซ์ใหม่ จนหาลำดับของสายได้ถูกต้องคือมอเตอร์เดินตามที่ละสเตป อย่างเป็นลำดับ



รูปที่ 2-16 แสดงการต่อวงจรเพื่อทดสอบโดยการสวิตซ์เพื่อหาลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่3
ทฤษฎีและหลักการการประมวลภาพและการนำภาพมาใช้

3.1 วิดีโอโอซีเอ็กซ์(VideoOCX)

วิดีโอโอซีเอ็กซ์ เป็นเครื่องมือในการทำการประมวลผลภาพซึ่งมีลักษณะเป็นแอคทีพเอ็กซ์ (ActiveX) สามารถนำไปประยุกต์ใช้ได้กับหลายๆ ภาษาในการเขียนโปรแกรม เช่น วิช่วลซี (Visual C) และ วิช่วลเบสิก (Visual Basic)

เบื้องต้นการใช้งานวิดีโอโอซีเอ็กซ์จะต้องทำการรวม วิดีโอโอซีเอ็กซ์ เข้าไปในโปรเจค(Project) ที่จะทำการพัฒนา ซึ่งจะแตกต่างกันตามภาษาที่ใช้ในการพัฒนา โดยจะขึ้นอยู่กับการเพิ่มส่วนที่เป็นแอคทีฟเอ็กซ์เข้าไปในโปรเจค ในบางภาษาเช่น วิช่วลซีพลัสพลัส (Visual C++) จำเป็นต้องกำหนดให้บางตัวแปรทำหน้าที่เป็นตัวควบคุม

ในส่วนของการแจ้งข้อผิดพลาด โอซีเอ็กซ์สามารถแจ้งเตือนข้อความได้หลากหลายเพื่อช่วยให้เข้าปัญหาที่เกิดขึ้น สามารถตั้งค่าฟังก์ชัน ErrorMessages ให้เป็นค่า TRUE แล้วโอซีเอ็กซ์จะแสดงข้อผิดพลาดโดยอัตโนมัติ ถ้าบางกระบวนการเกิดการผิดพลาดสามารถที่จะเรียกฟังก์ชัน GetLastErrorString หรือ GetLastErrorNum เป็นรับค่าข้อความแจ้งเตือนที่สามารถจะนำไปคำนวณและแสดงผลได้อีกด้วย

สามารถเลือกไดร์เวอร์ (Driver) ของกล้องที่ต้องการให้เป็นตัวจับภาพได้ถ้าหากมีไดร์เวอร์มากกว่า 1 ตัวทำงานอยู่บนเครื่องคอมพิวเตอร์โดยการตั้งค่าในพร็อพเพอตี้ (Property) หรือเรียกฟังก์ชัน ShowDriverDlg เพื่อแสดงไดอาล็อก (Dialog) ที่สามารถเลือกไดร์เวอร์จากรายการได้

ในการตั้งค่าเริ่มต้นของ โอซีเอ็กซ์และการเชื่อมต่อกับตัวสร้างภาพ (วิดีโอ หรือ เอวีไอ) ต้องทำการเรียกฟังก์ชัน Init ซึ่งต้องแน่ใจว่าขณะนั้นไม่มีโปรแกรมใช้งานตัวกล้องหรือไฟล์วิดีโอนั้นอยู่ เนื่องจากจะมีเพียงโปรแกรมเดียวที่สามารถใช้งานตัวกล้องหรือไฟล์วิดีโอนั้นได้ จากนั้นทำการตั้งค่าการแสดงผลให้เป็น TRUE เพื่อแสดงผลของวิดีโอที่ถ่ายภาพอยู่ขณะนั้น

ในการปิดการใช้งานโอซีเอ็กซ์และตัดการทำงานจากตัวสร้างภาพให้ทำการเรียกฟังก์ชัน Close

3.1.1 ตัวสร้างภาพ(Image Source)

วิดีโอโอซีเอ็กซ์รองรับการใช้งานตัวสร้างภาพในสองลักษณะคือ อุปกรณ์กล้องวิดีโอ (Video Devices) สามารถใช้งาน VFW (VFW compatible) ได้ และ ไฟล์วิดีโอแบบ เอวีไอ

สามารถเปลี่ยนการทำงานอยู่ทั้งสองแบบได้โดยการตั้งค่าฟังก์ชันโหมดโดย 0 หมายถึง วิดีโอ และ 1 หมายถึงเอวีไอ ก่อนทำการเรียกฟังก์ชัน Init ซึ่งทั้งสองชนิดสามารถเรียกฟังก์ชัน Capture เพื่อทำการจับภาพอัตโนมัติ ซึ่งวิธีนี้มีประโยชน์มากในการทำการประมวลผลภาพซึ่งจะพูดต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การแสดงผลวิดีโอแบบถ่ายทอดสด

สามารถแสดงผลได้ในสองลักษณะ โอเวอร์เลย์ (Overlay) และพรีวิว (Preview)

ในโหมดของโอเวอร์เลย์ตัวถ่ายภาพวิดีโอจะทำการคัดลอกทุกภาพของวิดีโอลงในหน่วยความจำของการ์ดแสดงผลโดยตรง (Graphic card) โดยไม่ไปยุ่งเกี่ยวกับซีพียู (CPU) เลย ทำให้ภาพที่ได้ไม่กระตุก แต่ไม่ทุกไดร์เวอร์ของกล้องวิดีโอที่จะรองรับการทำงานแบบนี้ ในการเปิดการใช้งานโอเวอร์เลย์ให้ใช้ฟังก์ชัน SetOverlay หลังจากทำการเลือกไดร์เวอร์และตั้งค่าเริ่มต้นแล้ว

ส่วนการแสดงผลแบบพรีวิว โดยทั่วไปแล้วโหมดนี้จะทำสามารถใช้งานได้กับทุกไดร์เวอร์ของกล้องวิดีโอ ในโหมดนี้กล้องวิดีโอจะจับภาพและส่งเข้าไปเก็บในหน่วยความจำของเครื่องคอมพิวเตอร์และจึงนำมาแสดงผล ซึ่งจะใช้ทรัพยากรของซีพียูมากกว่าแบบโอเวอร์เลย์ แต่จะปลอดภัยมากกว่าถ้าหากตัวฮาร์ดแวร์ (Hardware)

ไม่รองรับการทำงานแบบโอเวอร์เลย์ ในการเปิดการใช้งานแบบพรีวิวให้ใช้ฟังก์ชัน SetPreview หลังจากที่ทำการเลือกไดร์เวอร์และทำการตั้งค่าเริ่มต้นแล้ว

3.1.3 การประมวลผลภาพ (Image Processing) และการใช้งานแต่ละพิกเซล (Pixel Access)

ในการเข้าถึงข้อมูลของภาพโดยใช้การจับภาพของวิดีโอโอซีเอ็กซ์ ต้องทำเตรียมคอนเทนเนอร์ (Container) สำหรับภาพ โดยทำการเรียกฟังก์ชัน CreateGrayImageHandle หรือฟังก์ชัน CreateColorImageHandle หลังจากทำการตั้งค่าเริ่มต้นแล้ว ตัวควบคุม (Handle) จะถูกส่งค่าหลังจากผ่านกระบวนการย่อยในฟังก์ชัน Capture

ในวิช่วลซีพลัสพลัส มีสองทางในการทำงานกับภาพ คือส่งค่าตัวควบคุมไปยังฟังก์ชันที่แตกต่างกันในวิดีโอโอซีเอ็กซ์ หรือ วิดีโอโอซีเอ็กซ์ทูล (VideoOCXTools) เพื่อเก็บภาพหรือประมวลผลภาพ อีกทางหนึ่งคือใช้ฟังก์ชัน GetDataPointer เป็นรับค่าตัวชี้ (Pointer) ที่ชี้ไปยังข้อมูลในภาพโดยตรง ตัวอย่างการเขียนโปรแกรม ในภาษาวิช่วลซีพลัสพลัส

```
long *array = (long*) m_VideoOCX.GetDataPointer(image handle)
```

จะได้อาร์เรย์ (array) เป็นตัวชี้ซึ่งชี้ไปยังข้อมูลในตัวควบคุม image handle ตัวชี้จะสามารถเข้าถึงข้อมูลแต่ละพิกเซลของภาพได้โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 วิดีโอโอซีเอ็กซ์ทูล (VideoOCXTools)

วิดีโอโอซีเอ็กซ์ทูล ส่วนนี้แยกออกมาเป็นส่วนควบคุมแบบแอคทีฟเอ็กซ์จากวิดีโอโอซีเอ็กซ์ ใช้เพื่อการควบคุมการใช้งานฟังก์ชันในการประมวลผลภาพ เช่น การค้นหาขอบของภาพ หรือ การวางข้อความลงบนภาพ

3.1.5 รูปแบบของภาพที่ใช้

รูปแบบของภาพที่ใช้เมื่อต้องการเข้าถึงข้อมูลของภาพโดยตรง (ในซีพลัสพลัส) โดยใช้ตัวชี้ ชี้ไปยังข้อมูลในภาพ

ทุกภาพที่เป็นภาพสีจะเก็บไว้ใน ตัวควบคุมภาพสี (ColorImageHandle) ภายใน อาร์เรย์ที่เก็บข้อมูลขนาดไบต์ โดยเก็บค่าสีแบบ อาร์จีบี 24 บิต (24 Bit RGB) และเพื่อคุณภาพทำให้การเรียงของสีเป็น น้ำเงิน เขียว และ แดง ตามลำดับโดยเป็น 1 ไบต์ต่อ 1 สี

ภาพขาวดำ (Grayscale) จะเก็บไว้ในตัวควบคุมภาพขาวดำ (GrayImageHandle) ซึ่งถูกแสดงโดยอาร์เรย์ที่เก็บข้อมูลแบบไบต์ โดยทุกค่า Gray Value จะเก็บอยู่ในช่วงตั้งแต่ 0 (สีดำ) ถึง 255 (สีขาว)

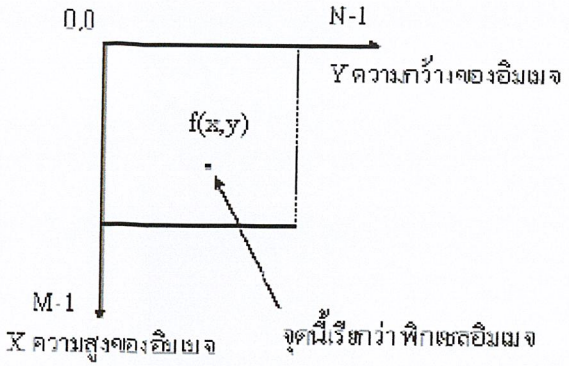
3.1.6 ทฤษฎีและหลักการการประมวลผลภาพ (Image Processing)

การประมวลผลภาพแบบดิจิตอล (Digital Image Processing)

โดยปกติแล้วตาของคนเราจะรับภาพแบบอนาล็อก ซึ่งสามารถอธิบายได้ด้วยตัวแปรทางคณิตศาสตร์แบบต่อเนื่อง แต่เนื่องจากเครื่องคอมพิวเตอร์มีการประมวลผลในลักษณะที่เป็นเลขฐานสอง ซึ่งเป็นหลักในการคำนวณ ดังนั้นการนำภาพเข้าสู่คอมพิวเตอร์จึงเป็นแบบ ดิจิตอล

ดิจิตอลอิมเมจ คือ ภาพที่ได้จากการสุ่มค่าในระบบพิกัด และการทำควอนไทเซชั่น (Quantization) ของค่าระดับความสว่าง (Brightness Value) หรือ ความเข้ม (Intensity) ระบบพิกัด Space นี้จะใช้กับการแสดงดิจิตอลอิมเมจซึ่งจะมีขนาดกว้างและสูงของภาพแสดงในแกน Y และ X ส่วนจุดใดๆ ที่วางบนระนาบ XY จะเป็นฟังก์ชัน f(x, y) และเรียกว่าพิกเซล (Pixel) ที่แสดงถึงค่าระดับความเข้ม ซึ่งจะเป็นจำนวนที่นับได้จำกัด (Finite Number) แบบไม่ต่อเนื่อง หรือเรียกว่า Discrete Quantity ซึ่งเป็นค่าที่ได้จากการทำควอนไทเซชั่นโดยจะใช้การแปลงจากอนาล็อก (Analog) เป็นดิจิตอล (Digital) ในกรณีของโมโนโครม (Monochrome) หรือที่รู้จักว่าภาพขาวดำ(gray-scale หรือ black and white image) ค่าของระดับความสว่างของแต่ละพิกเซลจะถูกแทนค่าด้วยตัวเลข โดยที่ค่าอยู่ระหว่าง 0-255 ซึ่งค่า 0 แทนสีดำ ค่า 255 แทนสีขาว ส่วนค่าระหว่าง 0-255 แทน ความแตกต่างของสีเทา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-1 แสดงระบบพิกัด Space

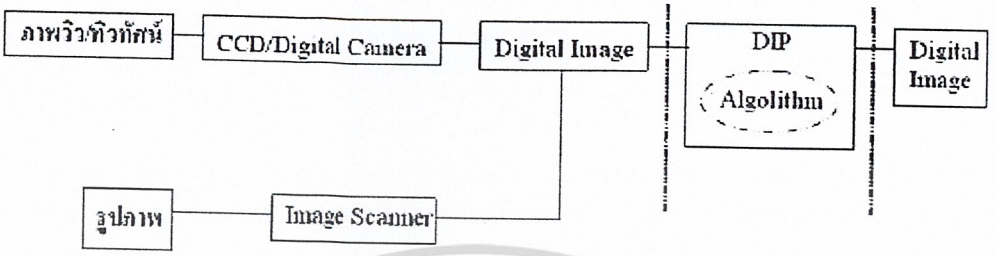
รูปที่ 3-2 ภาพแสดงกรณีที่ภาพเป็นโมโนโครม

3.1.6.1 กระบวนการทำดิจิตอลอิมเมจ

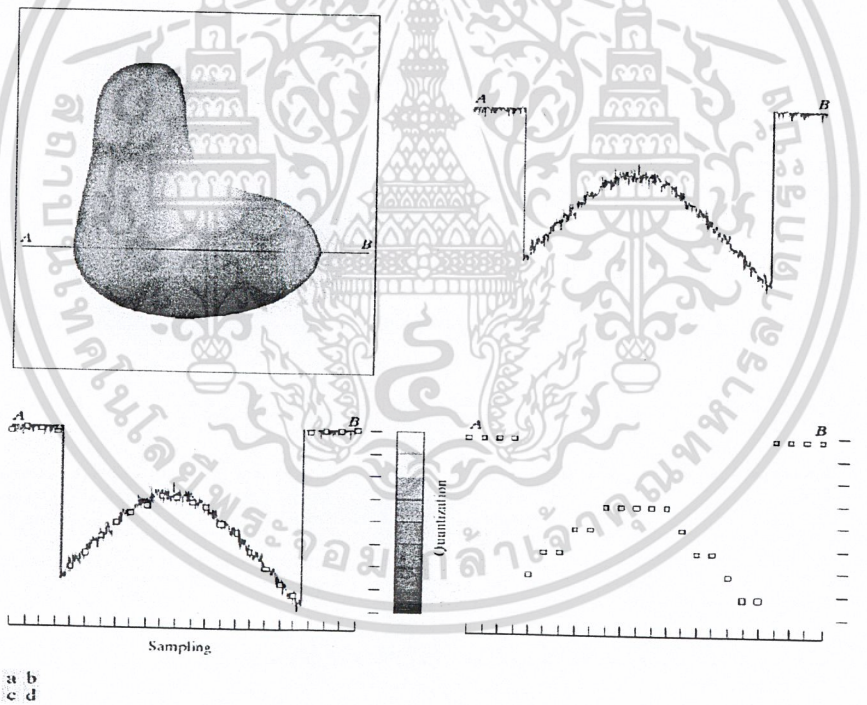
มีสองเหตุผลใหญ่ๆ ที่ต้องทำกระบวนการอิมเมจดิจิตอล คือเพื่อปรับปรุงอิมเมจดิจิตอลให้มองเห็นได้ง่ายขึ้น และเพื่อปรับปรุงอิมเมจให้หุ่นยนต์ตีความหมาย หรือเข้าใจจดจำรูปร่างลักษณะได้อย่างแม่นยำ ตัวอย่างเช่น การจดจำตัวอักษรหรือ Optical Character Recognition (OCR) ที่สามารถจดจำตัวอักขระได้ถึง 99.9% การปรับปรุงอิมเมจให้ใช้พื้นที่เก็บน้อยลง การตรวจสอบลายพิมพ์มือของแต่ละบุคคล เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำกระบวนการอิมเมจดิจิตอล คือการนำอิมเมจดิจิตอล เข้ามาทำกระบวนการโดยการใส่ฟังก์ชันอัลกอริทึมต่างๆ เข้าไป ก็จะได้เอาท์พุตเป็นอิมเมจดิจิตอล ที่ตรงตามแนวความคิดของการทำกระบวนการอิมเมจดิจิตอล เราสามารถแสดงภาพตั้งแต่การนำภาพวิว ทิวทัศน์ จนถึงเอาท์พุต ดังรูป

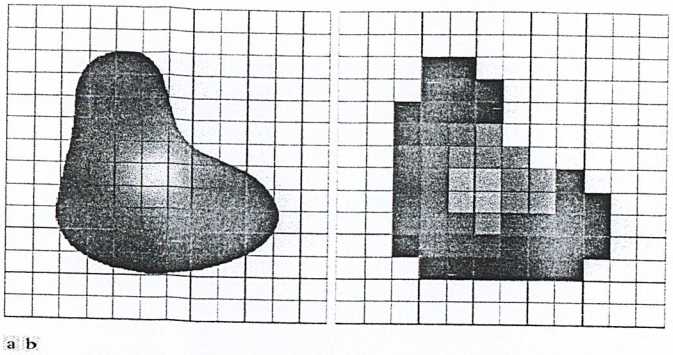


รูปที่ 3-3 กระบวนการทำดิจิตอลอิมเมจ



รูปที่ 3-4 กระบวนการสุ่มค่า (Sampling) จากภาพแบบอนาล็อกและทำควอนไทเซชั่น (Quantization)

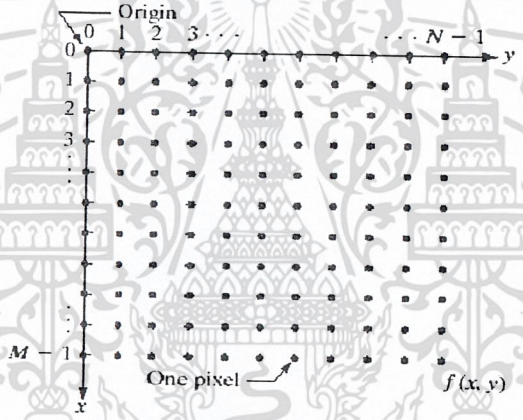
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



a b

รูปที่ 3-5 ผลลัพธ์หลังผ่านกระบวนการแปลงเป็นดิจิตอลอิมเมจ

ผลลัพธ์หลังจากผ่านกระบวนการแปลงเป็นดิจิตอลอิมเมจจะถูกเก็บไว้ในรูปของ เมตริกซ์ โดยมีแกนเป็น XY



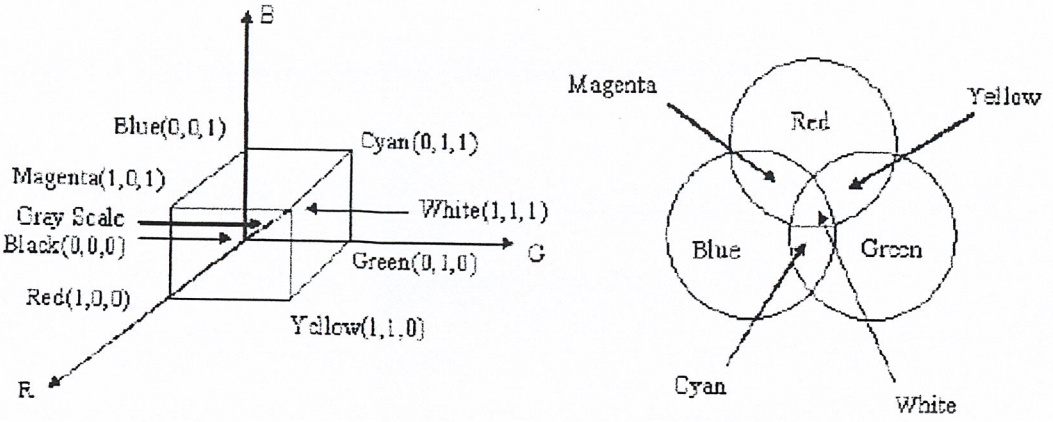
รูปที่ 3-6 ภาพแสดงเมตริกซ์ที่ใช้เก็บค่าเชิงตัวเลขในดิจิตอลอิมเมจ

$$f(x,y) = \begin{pmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{pmatrix}$$

รูปที่ 3-7โมเดลสีแบบ RGB (RGB Color Model)

โมเดลสีหรือ Color Space ประกอบด้วย 3 แม่สีหลัก ได้แก่ สีแดง เขียว และน้ำเงิน ถ้านำแต่ละแม่สีมาพล็อตกราฟในระบบพิกัด โมเดลสี โดยแต่ละสีมีค่า 0 ถึง 1 (0 แสดงถึงความมืด และ 1 แสดงถึงความสว่าง) จะได้ภาพการผสมสีทางแสงหรือการบวกแม่สีเข้าด้วยกัน (Additive Primary Color)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-8 แสดงโมเดลสีและการผสมสีของโมเดลสีแบบอาร์จีบี

ถ้าแต่ละแม่สีเป็นขนาด 8 บิต รวมทั้งหมด เท่ากับ 24 บิต ซึ่งสามารถสร้างสีใหม่ได้ถึง 256*256*256 เท่ากับ 16777216 สี ในที่นี้จะใช้พิกเซลอิมเมจที่มีแต่ละแม่สีเท่ากับ 8 บิต หรือ เรียกว่ามีความลึกเท่ากับ 24 บิตเป็นหลัก ถ้าพิกเซลเป็นภาพสีขนาด 24 บิต จะอ่านค่าอิมเมจดิจิตอลในรูปแบบเมตริกซ์ 2 มิติ ขนาด M*N เหมือนกับในสมการ แต่ค่า f(x, y) จะอยู่ในช่วงที่ประกอบด้วย

R ระดับ 0 จนถึง 255 (0 < R < 255)

G ระดับ 0 จะถึง 255 (0 < G < 255)

B ระดับ จ จนถึง 255 (0 < B < 255)

ในบางครั้งถ้าต้องการแปลงโมเดลสีให้เป็นขาวดำซึ่งก็คือ Gray Scale จะใช้สมการ

$$\text{Gray Scale} = 0.299*R + 0.587*G + 0.144*B$$

แต่เราสามารถใช้อีกสมการ โดยการหาค่าเฉลี่ยทั้งสามสีดังนี้

$$\text{Gray Scale} = (R + G + B)/3$$

การประมวลผลภาพโดยใช้ดิจิตอลอิมเมจสามารถใช้ระบบโมเดลสีมาทำการวิเคราะห์และประมวลผลต่อไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.6.2 กระบวนการลบภาพ

เป็นวิธีการที่ใช้เพื่อการลบส่วนที่ไม่ต้องการออกไปจากภาพโดยการเปรียบเทียบความแตกต่างระหว่างสองภาพ ซึ่งทั้งสองภาพนี้จะต้องลักษณะที่ใกล้เคียงกัน โดยในการทำงานจะเริ่มจากการเข้าถึงข้อมูลในระดับพิกเซลจากนั้นจะนำค่าของสีในแต่ละพิกเซลมาเทียบความแตกต่างกัน

รูปที่ 3-9 แสดงการประมวลผลภาพแบบลบภาพ

ขั้นตอนการทำงานจะทำการนำภาพสองภาพมาเปรียบเทียบกันโดยใช้ค่าสีในแต่ละพิกเซล ด้วยวิธีการลบกัน ผลลัพธ์ที่จะได้จะเหลือเพียงวัตถุที่ไม่มีในอีกภาพหนึ่งแต่วิธีจะไม่สามารถคงสีเดิมของวัตถุไว้ได้ ต้องทำการวนกลับไปตรวจสอบตำแหน่งเก่าในภาพเดิมเพื่อหาค่าสีที่ถูกต้องมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.6.3 กระบวนการทำเทรชโฮลด์(Threshold)

วิธีการทำเทรชโฮลด์เป็นการลดสิ่งรบกวน (Noise) ที่เกิดขึ้นกับภาพและกำจัดสิ่งที่ไปต้องการก่อนการนำไปประมวลผลภาพโดย จะทำการตั้งค่าหนึ่งๆ เพื่อใช้ในการปรับค่าของระดับสีในภาพ ซึ่งโดยทั่วไปภาพขาวดำ จะมีค่าเกรย์เลเวล (Gray-level) ตั้งแต่ 0-255 การกำหนดค่าคงที่จะใช้เพื่อเปรียบเทียบค่าของเกรย์เลเวลในแต่ละพิกเซล ถ้าน้อยกว่าจะปรับให้เป็น 0 ถ้ามากกว่าจะปรับให้เป็น 255 เทรชโฮลด์มี 2 ลักษณะคือ

การกำหนดค่าล่วงหน้า (Preassigned Threshold Value) เป็นวิธีที่ง่ายที่สุดโดยจะเลือกค่าคงที่ขึ้นมาค่าหนึ่งโดยทั่วไปมักเป็นค่ากลาง ของภาพ

การเทรชโฮลด์จากค่ากลาง (Mid – Range Threshold Value) แตกต่างจากวิธีแรกตรงที่จะทำการคำนวณอัตโนมัติโดยผู้ใช้ไม่จำเป็นต้องกำหนด โดยทั่วจะคำนวณโดยการหาค่ากลางหรือค่าเฉลี่ย

รูปที่ 3-10 แสดงการประมวลผลภาพแบบเทรชโฮลด์ด้วยค่าคงที่ = 100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่4
ทฤษฎีและหลักการการทำสเตรีโอออฟซิส (Stereopsis)

4.1 สเตรีโอออฟซิส

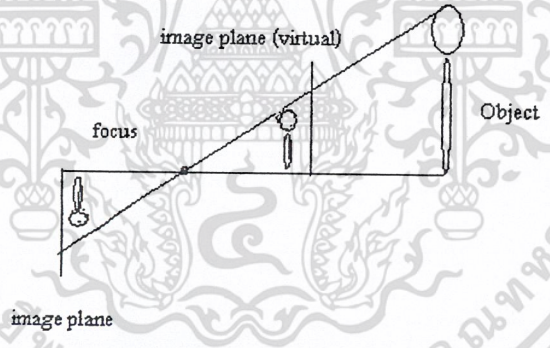
คือ วิธีการหาระยะทางของจุดที่สนใจกับกล้อง ซึ่งมีวิธีการดังนี้

4.1.1 การหาค่าเริ่มต้นของกล้อง(Calibration)

คือ การคำนวนหาค่าที่เป็นคุณสมบัติของกล้อง เช่น ลักษณะของเลนส์ โฟกัส, ความเบี่ยงเบน หรือ ลักษณะของฉากรับภาพ ว่ามีความกว้างเท่าไรเป็นต้น โดยใช้หลักการดังนี้

4.1.1.1 แบบจำลองกล้องรูเข็ม

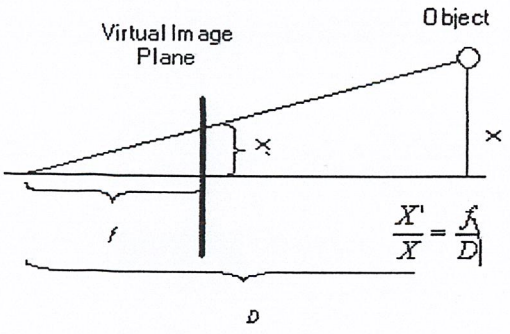
กล้องโดยในปัจจุบันโดยทั่วไปนั้นมีปัจจัยหลายอย่าง ที่เป็นส่วนทำให้เกิดภาพในกล้องต่างๆ ได้แก่ วัตถุ,ฉากรับภาพ(Image plane), โฟกัส และ เลนส์ ซึ่งจะนำไปคำนวนเพื่อหาค่าต่างๆ ที่เกี่ยวกับการเกิดภาพได้ ซึ่งถ้าหากที่จะพิจารณาถึงความสัมพันธ์ระหว่างวัตถุ และภาพที่เกิด ว่ามีขนาดเป็นอย่างไร ห่างกันเท่าไร เราจะสามารถพิจารณาจากแบบจำลองของกล้องรูเข็มได้



รูปที่ 4-1 แบบจำลองกล้องรูเข็ม

จากแบบจำลองนี้สามารถที่จะคำนวนหาตำแหน่งจริงของวัตถุ หรือ ตำแหน่งของวัตถุบนภาพได้ง่ายขึ้น โดยที่ไม่จำเป็นต้องรู้ค่าที่เกี่ยวกับเลนส์บางอย่างได้ เช่น ค่าความเบี่ยงเบนของเลนส์ ซึ่งจะทำให้การคำนวนตำแหน่งจริงของวัตถุหรือตำแหน่งวัตถุบนภาพได้ง่ายขึ้น ถ้าหากต้องการหาตำแหน่งของวัตถุที่ได้จากภาพของกล้องนั้น จำเป็นที่จะต้องรู้ถึงค่าโฟกัส และขนาดของฉากรับภาพ (Image plane) ทั้งในแนวตั้งและแนวนอน ซึ่งจะคำนวนได้จากการใช้ทฤษฎีสามเหลี่ยมคล้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-2 การคำนวณหาระยะบนภาพ

4.1.1.2 วิธีการคำนวนอัตราส่วนการเกิดภาพของกล้อง

วิธีการคำนวนอัตราส่วนการเกิดภาพของกล้องนั้นสามารถทำได้ง่ายๆ โดยใช้วิธีให้กล้องถ่ายภาพไปยังฉากฉากหนึ่งแล้วเราจึงทำการวัดว่าบริเวณที่กล้องเรานั้นถ่ายได้มีขนาดเท่าไร (สมมุติว่าความสูงจากจุดศูนย์กลางภาพไปยังขอบด้านบนมีความยาว x) แล้วทำการวัดค่าจากจุดหน้ากล้องไปถึงยังฉากที่กล้องถ่าย (ให้เป็นตัวแปร D) เมื่อเราได้ค่าทั้งสองแล้ว หากเราทำการกำหนดค่า ของโฟกัส หรือ ขนาดรับภาพขึ้นมาเองแล้ว เราจะสามารถหาค่าอีกค่าหนึ่งได้ทันที ดังรูปที่ 4-2 ซึ่งค่าที่เราได้มานั้นเป็นเพียงอัตราส่วนของการเกิดภาพของกล้อง เท่านั้น ไม่ใช่ค่าโฟกัสที่แท้จริงของเลนส์ และไม่ใช่ค่าขนาดฉากรับภาพจริงด้วย ซึ่งถ้าหากเราต้องการค่าพวกนี้จริงๆแล้ว ก็จะต้องรู้ค่าจริงค่าใดค่าหนึ่งก่อน

4.1.2 การคำนวนหาตำแหน่งจริงของวัตถุ

หลังจากที่ทำการหาค่าเริ่มต้นของกล้องได้แล้วนั้น ต่อมาก็จะทำการคำนวนระยะห่างของจุดที่เราสนใจกับกล้อง ซึ่งวิธีการคำนวนนั้นแบ่งออกเป็นสองวิธี

4.1.2.1 Passive Stereo System

คือการใช้กล้องตั้งแต่สองตัวขึ้นไปเพื่อหาข้อมูลสามมิติของวัตถุ มีหลักสำคัญดังนี้

1) กล้องทั้งหมดจะต้องถ่ายภาพของวัตถุชิ้นเดียวกัน

2) กล้องทั้งหมดต้องถ่ายวัตถุในเวลาเดียวกัน

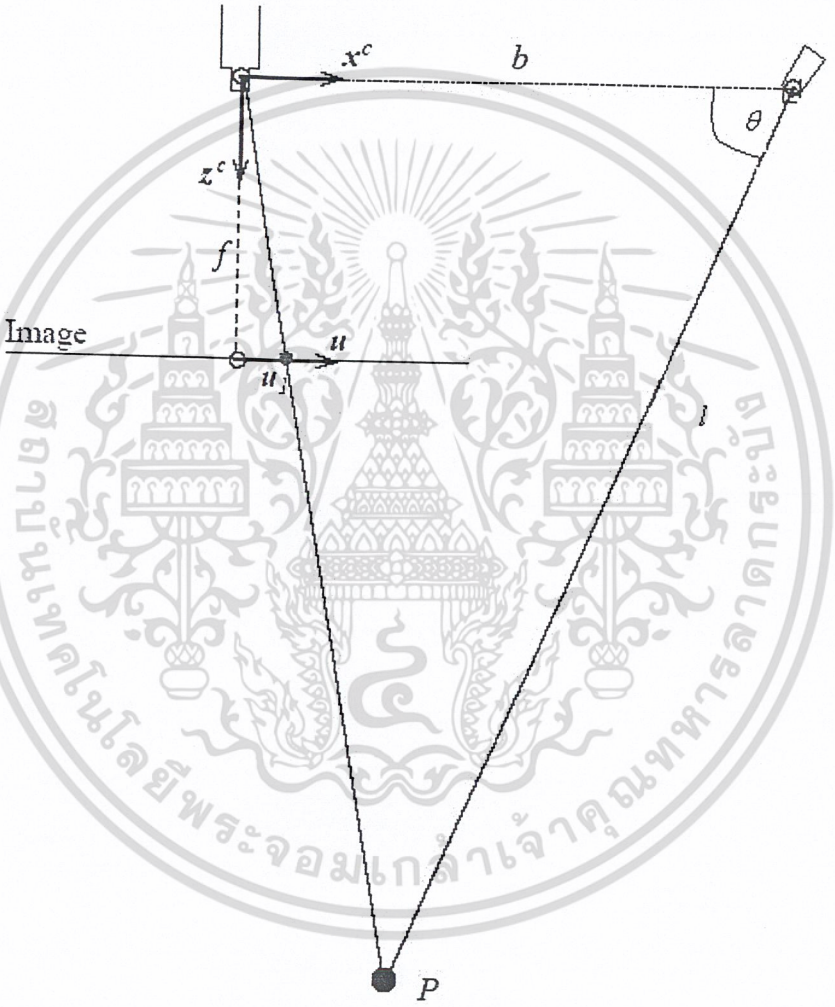
3) กล้องทั้งหมดจะต้องถ่ายวัตถุจากตำแหน่งที่ต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2.2 Active Stereo System

ในวิธีการแบบแอคทีฟนั้นจะคล้ายกับแบบพาสซีพ แต่จะเปลี่ยนจากกล้องตัวหนึ่งเป็นแหล่งกำเนิดแสง เช่น ไดโอด ซึ่งแสงลงบนวัตถุอย่างเป็นแบบแผนแทนกล้องตัวหนึ่ง เพื่อที่จะเป็นการชี้เฉพาะถึงจุดที่ต้องการหาข้อมูลสามมิติของวัตถุ

โดยจะคำนวนตำแหน่งจริง ได้โดยจะเริ่มจากคำนวน ตำแหน่งของจุดบนวัตถุบนระนาบ x,z (มุมมองด้านบน top view)



รูปที่ 4-3 มุมมองด้านบนของระบบ Active Stereo system

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 4-3

ด้วยคุณสมบัติของสามเหลี่ยมคล้าย จะให้

$$x^c = z^c \cdot \frac{u_1}{f} \qquad \text{----- (1)}$$

ระยะห่างจากแหล่งกำเนิดแสงถึงจุดบนวัตถุ จะให้

$$l\cos\theta = x^c - b \qquad \text{----- (2)}$$

$$l\sin\theta = z^c \qquad \text{----- (3)}$$

จาก (2),(3) จะให้

$$x^c = z^c \cot\theta + b \qquad \text{------ (4)}$$

จาก(1),(4) จะให้

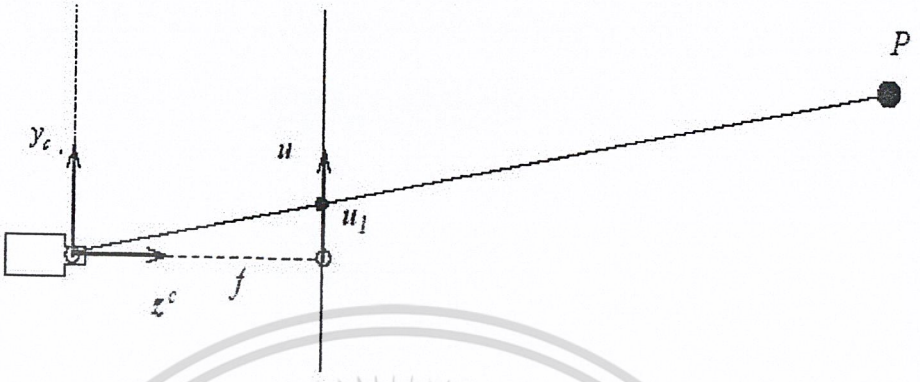
$$z^c = b \Big/ \left(\frac{u_1}{f} - \cot\theta \right) \qquad \text{----- (5)}$$

โดยที่

l	คือ ระยะทางของแหล่งกำเนิดแสงที่ฉายลงกระทบวัตถุ
θ	คือ มุมของแหล่งกำเนิดแสงกับระนาบของกล้อง
x^c, z^c	คือ ระยะของจุด P จนถึงกล้องบนระนาบ x,z
b	คือ ระยะทางของแหล่งกำเนิดแสงจนถึงจุด x^c
f	คือ ระยะของโฟกัส
u_1	คือ ระยะของรูปที่เกิดบนภาพจนถึงจุดกึ่งกลางภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเราหาพิกัดในระนาบ x, z ได้แล้ว ก็จะนำมาหาตำแหน่งของจุดบนวัตถุบนระนาบ y,z (ดูจากมุมมองด้านข้าง)



รูปที่ 4-4 มุมมองด้านข้างของระบบ Active Stereo system

จากรูป 4-4

ด้วยคุณสมบัติของสามเหลี่ยมคล้าย

$$y_c = \frac{u}{f} \cdot z^c \qquad \text{---- (6)}$$

เมื่อนำค่า z^c ที่ได้จาก (5) มาคำนวนใน (6) ก็จะได้ได้ค่าของระยะห่างของจุดที่เราสนใจกับกล้อง ซึ่งจะนำไปใช้ในการหาจุดบนวัตถุเพื่อสร้างเป็นโมเดลสามมิติต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
ไดเร็กซ์เอ็กซ์ (DirectX)

5.1 ความรู้เบื้องต้นเกี่ยวกับไดเร็กซ์เอ็กซ์

ไมโครซอฟท์ ไดเร็กซ์เอ็กซ์ (Microsoft DirectX) เป็นชุดของแอพพลิเคชั่นโปรแกรมมิ่งอินเตอร์เฟส (Application Programming Interface – API) ได้รับการพัฒนาและออกแบบเพื่อใช้จัดเตรียมอินเตอร์เฟส สำหรับควบคุมฮาร์ดแวร์ทางด้านมัลติมีเดีย บนระบบปฏิบัติการ ไมโครซอฟท์ วินโดว์ (Microsoft Windows) ได้อย่างมีประสิทธิภาพ และเป็นเครื่องมือให้โปรแกรมเมอร์ใช้คำสั่งควบคุมฮาร์ดแวร์ได้อย่างใกล้ชิด โดยไม่ต้องยุ่งยากกับการสร้างส่วนโปรแกรมติดต่อในระดับล่าง ซึ่งเครื่องมือที่ใช้ติดต่อจะแตกต่างกันไปตามประเภทของอุปกรณ์ ทำให้โปรแกรมเมอร์สร้างซอฟต์แวร์เพื่อทำงานทางด้านมัลติมีเดียได้เป็นอย่างดี เนื่องจากการติดต่อในแต่ละคอมโพเน็นท์จะเป็นของอุปกรณ์ประเภทหนึ่งๆ เป็นอิสระในการเขียนโปรแกรม

API ของ ไดเร็กซ์เอ็กซ์นั้น สร้างขึ้นมาบนรากฐานของ HAL (Hardware Abstraction Layer) สามารถซ่อนลักษณะของดีไวซ์ (Device) ที่เกี่ยวพันอยู่กับฮาร์ดแวร์ และออกแบบมาเพื่อให้สามารถรองรับความขยายตัวของฮาร์ดแวร์ใหม่ๆ ที่จะออกมาในอนาคต ดังนั้นจึงสามารถรองรับความสามารถของฮาร์ดแวร์ที่มีความสามารถเร่งความเร็วใหม่ๆ ที่ปัจจุบันยังไม่มี ด้วยความสามารถในการจำลองการทำงานผ่าน HEL (Hardware Emulation Layer) หรือเลือกที่จะหลีกเลี่ยงไม่ใช้ความสามารถนี้ ถ้า HEL ไม่สนับสนุน ทำให้โปรแกรมเมอร์พัฒนางานทางด้านมัลติมีเดียง่ายขึ้น ไม่จำเป็นต้องทดสอบกับอุปกรณ์ทุกตัวในท้องตลาด เพียงแต่ทดสอบกับไดเร็กซ์เอ็กซ์ก็เพียงพอแล้ว สำหรับผู้สร้างอุปกรณ์ฮาร์ดแวร์ ก็ต้องสร้างไดร์เวอร์ของอุปกรณ์นั้นๆ เพื่อให้สามารถทำงานเข้ากับไดเร็กซ์เอ็กซ์ได้

5.2 ชุดพัฒนาซอฟต์แวร์สำหรับไดเร็กซ์เอ็กซ์ (DirectX software Development Kit: DirectX SDK)

ชุดพัฒนาซอฟต์แวร์สำหรับไดเร็กซ์เอ็กซ์ นี้ประกอบด้วยชุดของไลบรารี่, ไฟล์ DLL, ไฟล์ header รวมไปถึงเอกสาร และตัวอย่างโปรแกรม ซึ่งช่วยในการพัฒนาซอฟต์แวร์ทางด้านมัลติมีเดีย ซึ่งการที่จะใช้ไดเร็กซ์เอ็กซ์ได้นั้น ก็ต้องทำการลงชุดพัฒนาซอฟต์แวร์สำหรับไดเร็กซ์เอ็กซ์นี้ก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 คอมโพเน็นท์ในไดเร็กซ์เอ็กซ์ 9

ไดเร็กซ์เอ็กซ์ 9 (DirectX 9) จะประกอบด้วย Component 6 ตัว คือ

5.3.1 DirectX Graphics

DirectX Graphics เป็นคอมโพเน็นท์ใหม่ที่รวมคอมโพเน็นท์เดิมที่ชื่อ DirectDraw กับ Direct3D ไปเป็นอินเตอร์เฟสอันเดียวคือ Direct3D Interface ซึ่งอยู่ในคอมโพเน็นท์ DirectX Graphics นี้ ใช้จัดการทางด้านกราฟฟิกการแสดงรูปภาพ 2 มิติ และวัตถุ 3 มิติ ซึ่งช่วยจัดการการใช้อุปกรณ์แสดงผล 3 มิติ ในรูปแบบที่ไม่ขึ้นกับชนิดของอุปกรณ์ที่จะใช้ (Device-Independent)

5.3.2 DirectX Audio

DirectX Audio เป็นคอมโพเน็นท์ใหม่ที่รวมคอมโพเน็นท์เดิมที่ชื่อ DirectSound กับ DirectMusic มาเป็นอินเตอร์เฟสในคอมโพเน็นท์ Direct X Audio นี้ ใช้จัดการทางด้านเสียง สนับสนุนการเร่งความเร็วทางด้านฮาร์ดแวร์ (Hardware acceleration) ในการสร้างเสียงแบบไดนามิก (Dynamic soundtrack) และเอฟเฟ็คเสียง 3 มิติขั้นสูง (Advanced 3D position effect)

5.3.3 DirectInput

DirectInput ใช้จัดการทางด้านการรับข้อมูลจากอุปกรณ์อินพุต เช่นจาก คีย์บอร์ด, เมาส์, จอยสติ๊ก เป็นต้น เพื่อเข้าถึงข้อมูลด้วยการติดต่อโดยตรงกับฮาร์ดแวร์ด้วยฮาร์ดแวร์ไดร์เวอร์ (Hardware Driver) ซึ่งจะทำงานได้เร็วกว่าการติดต่อด้วยเมสเสจของวินโดว์ นอกจากนี้ยังสนับสนุนการทำงานกับอุปกรณ์แบบแรงตอบสนอง (Force Feedback) อีกด้วย

5.3.4 DirectPlay

DirectPlay เป็นมีเดียที่เป็นอิสระบนเน็ตเวิร์ก ซึ่งหมายถึง สามารถรันได้บนเครือข่ายแบบ TCP/IP, เครือข่ายแบบ IPX หรือแม้แต่การต่อตรงจากโมเด็ม และการเป็นอิสระทางเน็ตเวิร์กนี้จะสามารถถูกพัฒนาต่อไปได้ในอนาคต นั่นคือ สามารถรองรับอุปกรณ์และโปรโตคอลใหม่ๆ เช่น IPV6, multicast โดยปกติ DirectPlay ใช้จัดการการทำงานด้านเน็ตเวิร์ค โดยไม่จำเป็นต้องไปยุ่งกับการทำงานในระดับล่าง มีหน้าที่การทำงานสำหรับการส่งข้อมูลทีเดียวให้หลายๆเครื่องเป็นกรุ๊ปได้ สามารถเขียนโปรแกรมติดต่อทางด้านเน็ตเวิร์ค ทั้งในการติดต่อผ่านเน็ตเวิร์คระหว่างโปรแกรมแบบมัลติเพลเยอร์ (Multiplayer) แบบเพียร์ทูเพียร์ (Peer-to-Peer) และแบบไคลเอ็นท์/เซิร์ฟเวอร์ (Client/Server)

5.3.5 DirectShow

DirectShow ใช้สำหรับการทำงานด้านมัลติมีเดียเป็นสถาปัตยกรรมทางด้านมีเดียสตรีมมิ่ง (media-streaming) ของไมโครซอฟท์ วินโดว์ ที่มีความสามารถในการแคปเจอร์ (Capture) สูง สามารถบรรจุข้อมูลที่เป็นวีดีโอ และการบีบอัดข้อมูลไฟล์ (Audio data compressed) ได้หลายรูปแบบ เช่น การเล่นไฟล์วีดีโอ ประเภท MPEG, Audio-Video (AVI), MPEG-1 Layer3 (MP3) และไฟล์นามสกุล .wav เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.6 DirectSetup

DirectSetup ใช้สำหรับการจัดการทางด้านการติดตั้งองค์ประกอบที่จะต้องใช้ในงานไดเร็กซ์เอ็กซ์บนเครื่องคอมพิวเตอร์ของผู้ใช้อย่างอัตโนมัติ หากเครื่องคอมพิวเตอร์นั้นยังไม่ได้ทำการติดตั้ง

5.4 ไดเร็กซ์เอ็กซ์ กราฟฟิก (DirectX Graphics)

ไดเร็กซ์เอ็กซ์ กราฟฟิก (DirectX Graphics) เป็นคอมโพเน็นท์ใหม่ ในไดเร็กซ์เอ็กซ์ เวอร์ชั่น 8 ซึ่งรวมเอาการทำงานของ คอมโพเน็นท์เดิม คือ ไดเร็กซ์ดรอว์ (DirectDraw) กับ ไดเร็กซ์ทรีดี (Direct3D) มาเป็นอินเตอร์เฟสเดียวคือ ไดเร็กซ์ทรีดี ซึ่งจะรวมการทำงานในการแสดงภาพ 2 มิติ และ 3 มิติ ไว้ในอินเตอร์เฟสนี้

5.4.1 ไดเร็กซ์ทรีดี

ไดเร็กซ์ทรีดี นี้เป็นแอพพลิเคชั่นโปรแกรมมิ่งอินเทอร์เฟสระดับล่าง ซึ่งเป็นเครื่องมือสำหรับโปรแกรมเมอร์ที่ต้องการสร้างเกมส์หรือโปรแกรมมัลติมีเดียที่แสดงผล 3 มิติ ที่มีประสิทธิภาพสูง บนระบบปฏิบัติการวินโดว์ ซึ่งการใช้เอพีไอไดเร็กซ์ทรีดีนี้ ทำให้โปรแกรมสามารถติดต่อโดยตรงกับฮาร์ดแวร์เร่งความเร็วในระดับล่าง จึงมีความยืดหยุ่นในการใช้งานสูง

5.4.2 คุณสมบัติของไดเร็กซ์ทรีดี

1) สนับสนุนการใช้งาน Depth Buffer (โดยใช้ z-buffer หรือ w-buffer)

2) สามารถเรนเดอร์ภาพได้ทั้งในแบบ Flat Shading และ Gouraud Shading

3) สามารถสร้างแหล่งกำหนดแสงได้หลายชนิดและหลายแหล่ง

4) รองรับการใช้วัสดุแมตทีเรียล (Material) และพื้นผิวเท็กซ์เจอร์ (texture) รวมไปถึงการทำมิพแมพ (mipmap)

5) สนับสนุนซอฟต์แวร์จำลองการใช้งานของฮาร์ดแวร์ (Software Emulation Driver) ที่มีประสิทธิภาพสูง

6) มีการทำงานโดยไม่ขึ้นกับชนิดของฮาร์ดแวร์

7) รองรับการแปลงพิกัดวัตถุ (Transformation) และการตัดภาพ (Clipping)

8) สนับสนุนคำสั่งพิเศษที่มีอยู่ในซีพียู สามารถรองรับสถาปัตยกรรมแบบ Intel MMX, Intel Streaming Single-Instruction, Multiple-Data (SIMD) Extensions (SSE) และสถาปัตยกรรมทรีดีของ AMD (AMD's 3D Architecture)

9) สนับสนุน Hardware Abstraction Layer (HAL)

10) สนับสนุนการทำ page flipping ด้วย back buffer จำนวนมากในโปรแกรมที่แสดงผลแบบเต็มจอภาพ (full screen)

11) สนับสนุนการตัดภาพ (Clipping) ในโปรแกรมประยุกต์ที่ทำงานในวินโดว์โหมด และแบบโหมดเต็มจอภาพ

12) รองรับการทำ 3D - z buffer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

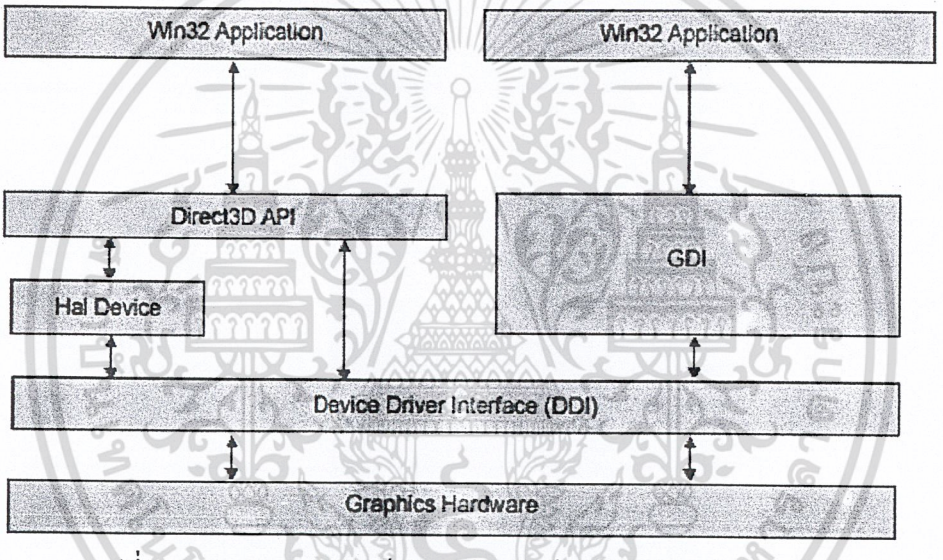
13) สามารถใช้งาน image-stretching hardware ได้

14) สามารถเข้าถึงหน่วยความจำของอุปกรณ์แสดงผล (display-device memory) แบบ Standard และ Enhance ได้ในขณะเดียวกัน

นอกจากนี้ยังมีคุณสมบัติอื่นๆ เช่น การใช้งานฮาร์ดแวร์แต่เพียงผู้เดียว (Exclusive hardware access) และการเปลี่ยนความละเอียดในการแสดงผล (Resolution switching)

ด้วยคุณสมบัติข้างต้นเหล่านี้ การพัฒนาโปรแกรมด้วยไดเร็กซ์ทรีดี จึงมีประสิทธิภาพสูงกว่าการใช้ กราฟฟิกดีไวซ์อินเตอร์เฟส (GDI) ของวินโดว์และการเขียนโปรแกรมบนระบบปฏิบัติการดอส

ตามแผนผังข้างล่างนี้ แสดงถึงความสัมพันธ์ระหว่างอินเตอร์เฟสไดเร็กซ์ทรีดี, กราฟฟิกดีไวซ์อินเตอร์เฟส (Graphics Device Interface : GDI), ฮาร์ดแวร์แอ็บสแตร็กชันเลเยอร์ (Hardware abstraction Layer : HAL) และ ฮาร์ดแวร์



รูปที่ 5-1 แสดงความสัมพันธ์ระหว่างไดเร็กซ์ทรีดีกับระบบคอมพิวเตอร์

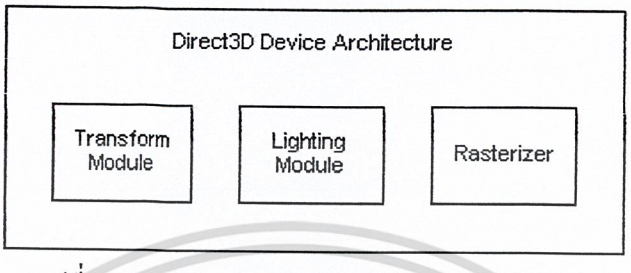
เมื่อนำมาเปรียบเทียบระหว่างไดเร็กซ์ทรีดี กับ กราฟฟิกดีไวซ์อินเตอร์เฟส (GDI) ทั้งคู่จะมีการเข้าถึงกราฟฟิกฮาร์ดแวร์ (Graphics Hardware) ผ่านดีไวซ์ไดรเวอร์ (Device Driver Interface : DDI) สิ่งที่ต่างกันไป คือ ไดเร็กซ์ทรีดีมีข้อดีกว่าของหน้าที่การทำงานกับฮาร์ดแวร์เมื่อมีการเลือกใช้ HAL ซึ่งดีไวซ์ HAL นี้จัดการความเร็วทางด้านฮาร์ดแวร์ บนพื้นฐานของชุดการทำงานที่สนับสนุนกราฟฟิกดีไวซ์

หากฮาร์ดแวร์ที่มีอยู่ในเครื่องคอมพิวเตอร์ใดไม่สนับสนุนการเร่งความเร็วในส่วนใดของไปป์ไลน์ก็ตาม ไดเร็กซ์ทรีดีจะใช้ซอฟต์แวร์ที่มีประสิทธิภาพสูงในการทำงานในส่วนนั้นแทน แต่การทำงานของซอฟต์แวร์ ก็ยังมีประสิทธิภาพต่ำกว่าใช้ฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.3 ไดเร็กซ์ทรีดีดีไวซ์(Direct3D Device)

เมื่อพูดตามหลักสถาปัตยกรรม ไดเร็กซ์ทรีดีดีไวซ์ จะประกอบไปด้วย ทรานส์ฟอร์เมชั่นโมดูล (transformation module), ไลท์ทิ่งโมดูล (lighting module) และ แรสเทอไรซิ่งโมดูล (rasterizing module) ตามรูปข้างล่าง



รูปที่ 5-2 แสดงสถาปัตยกรรมของไดเร็กซ์ทรีดีดีไวซ์

ไดเร็กซ์ทรีดีในปัจจุบันสนับสนุนดีไวซ์อยู่สองชนิด คือ

1) ฮาร์ดแวร์แอบสแต็กเลเยอร์(HAL : hardware abstract layer) เป็นดีไวซ์ที่มีฮาร์ดแวร์ช่วยเร่งความเร็วในการทำ แรสเทอไรซ์เซชัน (rasterization) และ เชดดิง (shading) โดยประมวลผลเวอร์เท็กซ์ แบบทั้ง ฮาร์ดแวร์ และ ซอฟต์แวร์ ซึ่งหากแอพพลิเคชั่น(application) ที่กำลังทำงานอยู่มี ตัวแสดงผลที่สนับสนุนไดเร็กซ์ทรีดี แอพพลิเคชั่นจะใช้ ตัวแสดงผลนั้นในการทำทรีดีโอเปอเรชั่น (3D operation)

2) เรฟเฟอร์เรนซ์ดีไวซ์ (reference device) จะไม่เหมือนกับ ซอฟต์แวร์ดีไวซ์(software device) แบบเรฟเฟอร์เรนซ์สนับสนุนรูปแบบทุกอย่างของ ไดเร็กซ์ทรีดี ซึ่งจะเน้นไปทางความแม่นยำมากกว่าความเร็วในการประมวลผล

ซึ่งเราสามารถแยกแยะดีไวซ์ออกเป็น 2 ไดร์เวอร์(driver) คือ ซอฟต์แวร์และเรฟเฟอร์เรนซ์ เป็นตัวแทนของ ซอฟท์แวร์ไดร์เวอร์ และ HAL เป็นตัวแทนของฝั่งฮาร์ดแวร์ไดร์เวอร์

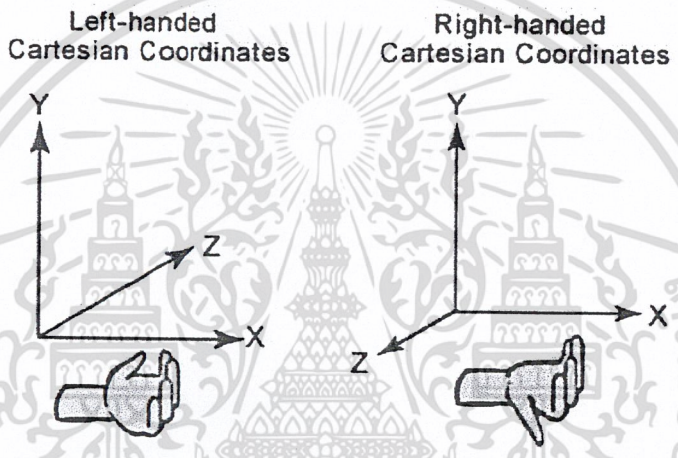
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.4 ทฤษฎีระบบพิกัด 3 มิติ

5.4.4.1 ระบบพิกัด 3 มิติ (3-D Coordinate System)

แบบฉบับของโปรแกรมประยุกต์ที่แสดงผลกราฟฟิก 3 มิติ จะมี 2 แบบในการแสดงระบบพิกัด 3 มิติ คือ มือซ้าย และมือขวา ในระบบพิกัดทั้งคู่ แกนด้านบวกของ x จะไปทางด้านขวา และแกนด้านบวกของ y จะขึ้นไปข้างบน สามารถสังเกตเพื่อจดจำได้จากทิศทางของแกนบวกของ z ได้ โดยการหงายมือขึ้น จากนั้นกางนิ้วมือออกชี้ไปยังแกนบวกของ x แล้วงอนิ้วขึ้นด้านบนตามทิศทางของแกนบวกของ y ทิศทางที่นิ้วโป้งชี้ไป จะเป็นทิศทางของแกนบวกของ z ตามมือที่ใช้

ไมโครซอฟท์ ไดเร็กซ์ทรีดี มักจะใช้ระบบพิกัด 3 มิติ แบบมือซ้าย ถ้าต้องการทำโปรแกรมที่เป็นระบบมือขวา จะต้องเปลี่ยนข้อมูลบางอย่าง ดังนี้



รูปที่ 5-3 แสดงระบบพิกัด 3 มิติ แบบมือซ้ายและมือขวา

- เปลี่ยนรายการของจุดของสามเหลี่ยม (triangle vertices) โดยสลับแบบตามเข็มนาฬิกาจากด้านหน้า ตัวอย่างเช่น ถ้าลำดับจุดเป็น v0, v1, v2 ให้เปลี่ยนข้อมูลก่อนที่จะส่งค่าไปให้ Direct3D คือ v0,v2, v1

- ใช้การวิวเมตริกซ์เพื่อปรับพิกัดเวิลด์สเปซ (World Space) ด้วย -1 ในทิศทางของแกน Z ซึ่งทำได้โดย สลับเครื่องหมายของ _31, _32, _33, _34 แถวที่สามในเมตริกซ์ ของโครงสร้าง D3DMATRIX ที่ใช้สำหรับวิวเมตริกซ์

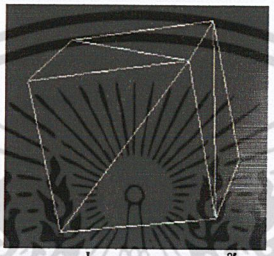
สำหรับฟังก์ชันที่จะใช้บอกไดเร็กซ์ทรีดีว่า เราจะใช้ระบบพิกัด 3 มิติ แบบไหนนั้น ก็คือ D3DXMatrixLookAtLH( ) สำหรับระบบมือซ้าย และ D3DXMatrixLookAtRH( ) สำหรับระบบมือขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.4.2 ชุดของจุดของวัตถุ 3 มิติ (3-D Primitive)

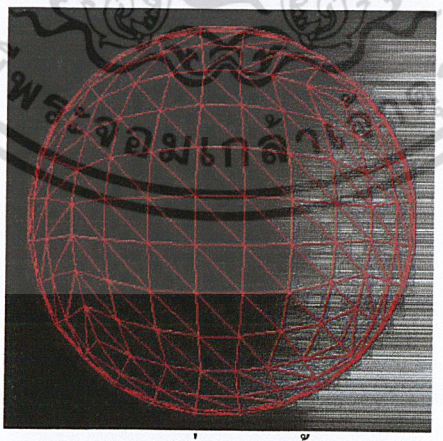
3-D Primitive เป็นชุดของจุด (vertices) ที่เป็นของวัตถุ 3 มิติ และพริมิทีฟ ที่ง่ายที่สุด คือ ชุดของจุดในระบบพิกัด 3 มิติ ที่ถูกเรียกว่า รายการจุด (Point List)

โดยปกติ 3-D Primitive จะหมายถึงโพลีกอน เป็นรูปปิด 3 มิติ ที่มีจุดที่มีการเชื่อมกันอย่างน้อย 3 จุด โพลีกอนแบบพื้นฐานที่สุดคือรูปสามเหลี่ยม ไมโครซอฟท์ ไดเร็กซ์ทรีดี ส่วนใหญ่ใช้สามเหลี่ยมในการประกอบโพลีกอน เพราะจุดสามจุดในรูปสามเหลี่ยมนั้นจะรับประกันได้ว่าอยู่ในระนาบเดียวกัน เพื่อให้การเรนเดอร์นั้นเป็นไปอย่างมีประสิทธิภาพ จึงสามารถประกอบรูปสามเหลี่ยมเป็นโพลีกอนหรือวัตถุที่มีความซับซ้อนได้



รูปที่ 5-4 แสดงรูปลูกบาศก์ที่ประกอบกันขึ้นมาจากรูปสามเหลี่ยม

รูปข้างบนนี้แสดงรูปสี่เหลี่ยมลูกบาศก์ ในแต่ละด้านประกอบขึ้นมาจากสามเหลี่ยม 2 รูป ซึ่งรูปสี่เหลี่ยมลูกบาศก์นี้ประกอบขึ้นมากจากชุดของรูปสามเหลี่ยม นอกจากนี้ ยังสามารถลงเท็กซ์เจอร์และชนิดวัสดุ (material) บนพื้นผิววัตถุได้ เพื่อสร้างพื้นผิวของพริมิทีฟ ให้เห็นเป็นลูกสี่เหลี่ยมลูกบาศก์ทรงตันได้



รูปที่ 5-5 แสดงภาพทรงกลมที่ประกอบขึ้นมาจากรูปสามเหลี่ยม

จากรูปด้านบนนี้ แสดงถึงทรงกลมที่ประกอบขึ้นมาจากรูปสามเหลี่ยม ซึ่งแสดงให้เห็นว่าสามารถใช้สามเหลี่ยมเพื่อสร้างพริมิทีฟสำหรับรูปทรงที่มีพื้นผิวราบเรียบได้ เมื่อทำการลงพื้นผิวจะเห็นเป็นรูปทรงกลมที่พื้นผิวเรียบได้

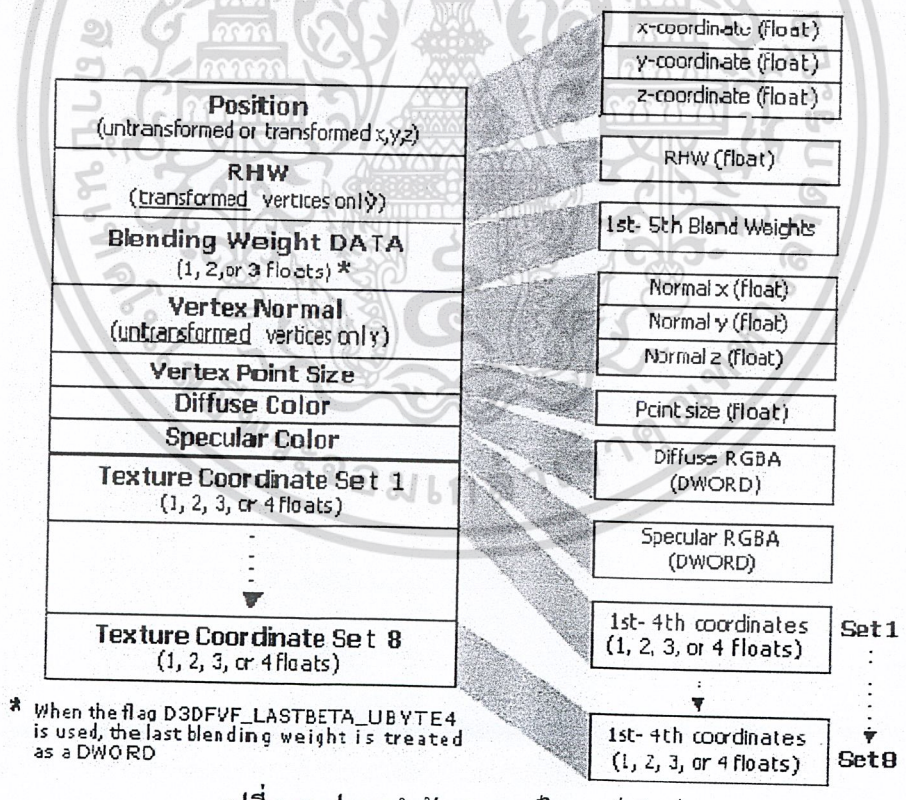
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.4.3 เวกเตอร์ (Vector)

เวกเตอร์ในไดเร็กซ์ทรีดี มีความหมายว่า ค่าที่แสดงถึงระยะและทิศทางของวัตถุ เช่น เวกเตอร์ที่แสดงพิกเซลบนจอภาพ จะประกอบด้วยค่าในแกน x และ y ถ้าทิศทางในแนวแกน x เป็นบวก จะหมายถึงพิกเซล นั้นจะอยู่ไปทางด้านขวาของจอภาพ ถ้าทิศทางในแนวแกน y เป็นบวก จะหมายถึงพิกเซลที่ตำแหน่ง y นั้นจะอยู่ไปทางด้านล่างของจอภาพ เพราะตำแหน่งเริ่มต้นของจอภาพ คือมุมบนซ้ายสุด (0, 0) ซึ่งเวกเตอร์ที่ประกอบด้วยค่า 2 ค่าอย่างนี้เราเรียกว่าเวกเตอร์ 2 มิติ (2-D Vector) ซึ่งในไดเร็กซ์ทรีดี จะใช้เวกเตอร์อยู่ 3 ประเภท คือ เวกเตอร์ 2 มิติ, เวกเตอร์ 3 มิติ, เวกเตอร์ 4 มิติ

5.4.4.4 เวอร์เท็กซ์ (Vertex)

เวอร์เท็กซ์ คือจุดในโลก 3 มิติ โดยอย่างน้อยจะมีเวกเตอร์ 1 เวกเตอร์ การสร้างเวอร์เท็กซ์ในไดเร็กซ์ทรีดี จะใช้วิธีสร้างแบบเฟล็กซิเบิลเวกเตอร์ (Flexible Vector Format : FVF) ซึ่งเป็นโครงสร้างที่ประกอบไปด้วยเวกเตอร์ที่ต้องการจะใช้งานเท่านั้น เวกเตอร์ไหนที่ไม่ใช้ก็ไม่ต้องใส่ โดยจะต้องเรียงลำดับเวกเตอร์และใช้ประเภทของตัวแปรตามรูปแบบในรูป 5-6 นี้



รูปที่ 5-6 รูปแบบลำดับเวกเตอร์ในเวอร์เท็กซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.4.5 โมเดลสเปซ (Model Space)

โมเดลสเปซ (Model Space) เป็นพิกัดในโลก 3 มิติ (3-D Coordinate) ที่ใช้ภายในโมเดลวัตถุแต่ละอัน ซึ่งมีสามเหลี่ยมแต่ละชิ้นนั้นเกิดขึ้นจากเวอร์เท็กซ์ จำนวน 3 จุด โดยแต่ละจุดจะมีจุดกำเนิดอ้างอิงจุดเดียวกัน ซึ่งจุดกำเนิดนี้จะอยู่ที่กลางโมเดลแต่ละอัน

สำหรับพิกัดต่างๆ ที่โมเดลแต่ละอันใช้นี้จะเรียกว่าพิกัดของตนเอง (Local Coordinate) ขอบเขตของโมเดลสเปซ ขึ้นอยู่กับขนาดของโมเดล ว่าจะมีความกว้าง, ยาว และสูงเพียงใด

5.4.4.6 รูปแบบในการให้แสงเงา (Shading Mode)

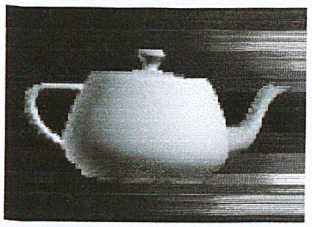
รูปแบบในการให้แสงเงา ใช้สำหรับการเรนเดอร์โพลีกอนให้มีผลกระทบจากแสงอย่างทั่วถึงปรากฏขึ้น ซึ่งรูปแบบในการให้แสงเงา จะกำหนดจากความหนาแน่นของสีและแสงที่จุดใดๆ บน พื้นผิวของโพลีกอน สำหรับไมโครซอฟท์ ไดเร็กซ์ทรีดีสนับสนุนการให้แสงเงา 2 รูปแบบ ได้แก่

1) การให้แสงเงาแบบแบนราบ (Flat Shading) ในรูปแบบการให้แสงเงาแบบแบนราบไดเร็กซ์ทรีดี จะทำการเรนเดอร์โพลีกอนโดยใช้สีของวัสดุ (material) ของโพลีกอน ที่จุด (vertex) แรก ให้กับโพลีกอนทั้งหมด วัตถุ 3 มิติ ที่มีการเรนเดอร์ด้วยรูปแบบการให้แสงเงาแบบแบนราบ จะมีขอบอย่างชัดระหว่างโพลีกอน ถ้าหากว่าเป็นคนละระนาบ ดังรูปที่ 6.8 เป็นการให้แสงเงาแบบแบนราบให้กับกาน้ำชา ซึ่งจะเห็นได้ว่าจะเห็นขอบของแต่ละโพลีกอนอย่างชัดเจน แต่การให้แสงเงาแบบนี้ จะมีข้อดีคือใช้เวลาในการคำนวณผลน้อยสุด

รูปที่ 5-7 แสดงการให้แสงเงาแบบแบนราบให้กับกาน้ำชา

2) รูปแบบการให้แสงเงาแบบเการาด์ (Gouraud Shading) เมื่อไดเร็กซ์ทรีดีเรนเดอร์โพลีกอนโดยใช้รูปแบบการให้แสงเงาแบบเการาด จะคำนวณสีสำหรับแต่ละจุดโดยใช้จุดปกติและค่าตัวแปรของแสง จากนั้นแทรกสีไปผ่านผิวหน้าของโพลีกอน การแทรกนี้จะทำแบบเป็นเชิงเส้น ตัวอย่างเช่น ถ้ามีส่วนประกอบที่เป็นสีแดงของสีในจุดแรก เท่ากับ 0.8 และส่วนประกอบที่เป็นสีแดงของจุด 2 เท่ากับ 0.4 ใช้การให้แสงเงาแบบเการาด์และโครงสร้างสี RGB จะกำหนดค่าส่วนประกอบสีแดงได้เท่ากับ 0.6 ให้กับจุด ที่ตำแหน่งกลางของเส้นระหว่างจุด (vertice) ทำให้สีเรียบเนียนกว่าการให้แสงเงาแบบแบนราบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

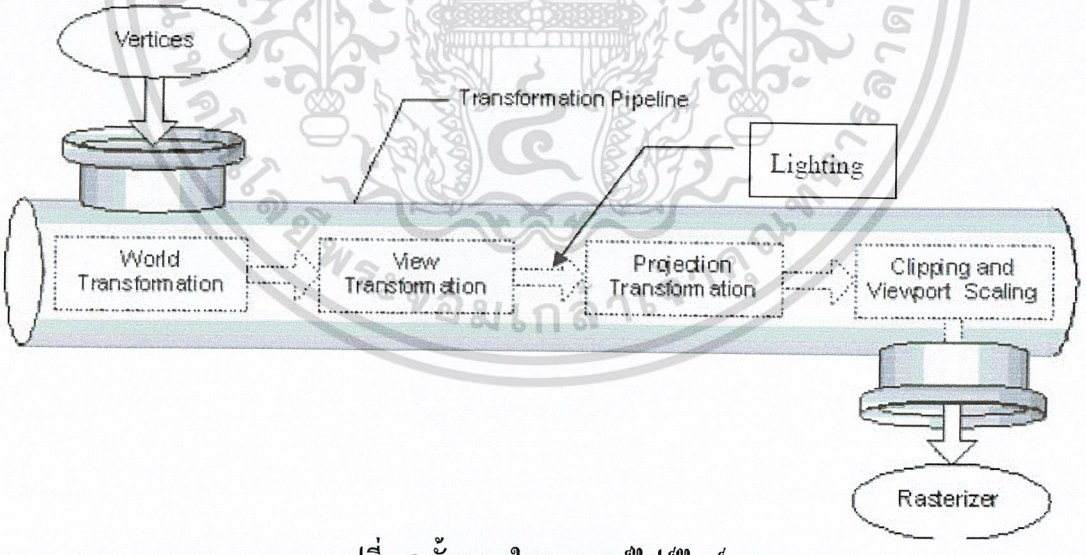


รูปที่ 5-8 แสดงการให้แสงเงาแบบเการาดให้กับกาน้ำชา

5.5 กระบวนการเรนเดอร์ออบเจ็กท์ 3 มิติ

ในการพัฒนาโปรแกรมประยุกต์ที่ใช้ไดเร็กซ์ทรีดีนั้น จะใช้พื้นฐานอยู่บนทฤษฎีเกี่ยวกับ 3 มิติ ได้แก่ เวอร์เท็กซ์, โพลีกอนและคำสั่งที่ใช้ควบคุมข้อมูลเหล่านี้ รวมทั้งการเข้าถึงการแปลงพิกัดของวัตถุ ซึ่งเป็นความสามารถในการเข้าถึงไปป์ไลน์ของกราฟฟิก 3 มิติ ในส่วนของการเปลี่ยนแปลงตำแหน่ง (transformation), การให้แสง (Lighting) และขั้นตอนการแสดงผลเพื่อสร้างภาพ (Rasterization)

ซึ่งขั้นตอนในกระบวนการเรนเดอร์ออบเจ็กท์ 3 มิติ แบ่งเป็น 2 ขั้นตอน คือ กระบวนการแปรพิกัดและให้แสง (Transformation and Lighting : T&L) ซึ่งเป็นกระบวนการแปรพิกัดของเวอร์เท็กซ์ที่มีรูปแบบทศนิยม ไปสู่พิกัดแบบพิกเซลตามมุมมองของกล้องจำลองภายในซีน รวมถึงกระบวนการให้แสง ขั้นตอนที่ 2 คือ กระบวนการแปรมาสู่พิกเซล (Rasterization) เป็นขั้นตอนลงจุด เส้น และ รูปสามเหลี่ยม ด้วยภาพกราฟฟิกที่ได้จากกระบวนการ T&L



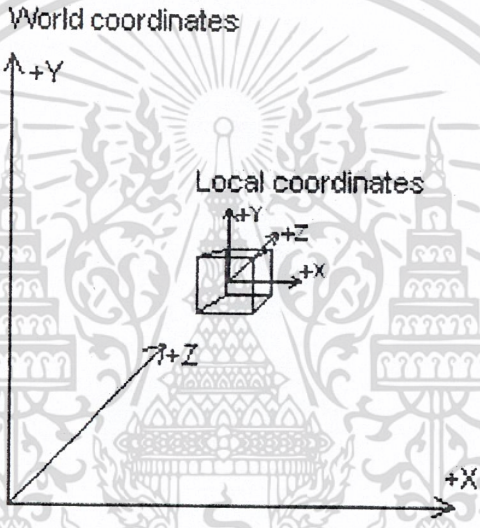
รูปที่ 5-9 ขั้นตอนในเรนเดอร์ไปป์ไลน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.1 กระบวนการแปรพิกัดและให้แสง (Transformation and Lighting : T&L)

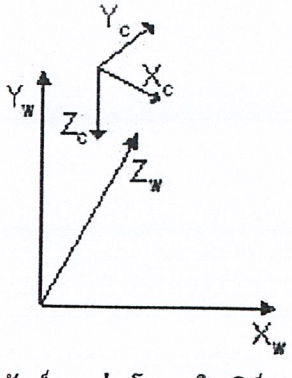
กระบวนการแปรพิกัด (Transformation) จะมีส่วนที่ทำหน้าที่ส่งอินพุทต่างๆ ผ่านไปยังฟังก์ชันการทำงานในรูปแบบไปป์ไลน์ของไดเร็กซ์ทรีดี เรียกว่า Transformation Engine ซึ่งมีขั้นตอนการคำนวณเพื่อแปลงพิกัดต่างๆ ดังนี้

1) World Transformation เป็นการคำนวณเพื่อแปลงพิกัด (Coordinate) หรือตำแหน่งของเวอร์เท็กซ์ (Vertex) ต่างๆ ที่ประกอบขึ้นเป็นโมเดลที่อยู่ในโมเดลสเปซ (Model Space) ให้เป็นพิกัดในเวิร์ลสเปซ (World Space) กล่าวได้ว่า เป็นการแปลงตำแหน่งเวอร์เท็กซ์เดิมที่มีความสัมพันธ์อยู่ในท้องถิ่นตนเอง (Local Coordinate) ให้มีความสัมพันธ์อ้างอิงกับตำแหน่งเวอร์เท็กซ์อื่นๆ ที่อยู่ในโลก 3 มิติ ด้วยกัน



รูปที่ 5-10 แสดงความสัมพันธ์ระหว่างโมเดลในโมเดลสเปซที่มาอ้างอิงตำแหน่งใหม่ในเวิร์ลสเปซ

5.5.1.2 View Transformation เป็นการคำนวณเพื่อแปลง ตำแหน่งของเวอร์เท็กซ์ต่างๆ บางส่วนของเวิร์ลสเปซ (World Space) ให้ไปเป็นพิกัด ในคาเมร่าสเปซ (Camera Space) ซึ่งถือได้ว่าเป็นการแปลงให้มาอยู่ในมุมมองของผู้ดู

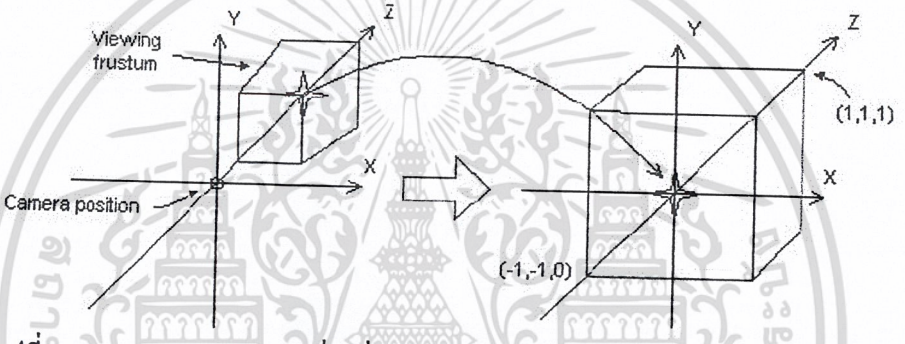


รูปที่ 5-11 แสดงความสัมพันธ์ระหว่างโมเดลในเวิร์ลสเปซกับมุมมองของกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

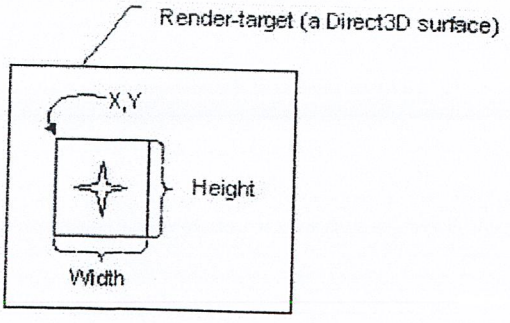
3) Lighting เป็นการคำนวณค่าสีต่างๆ ให้กับเวอร์เท็กซ์ เช่น สีที่ได้จากเท็กเซล (Texel) ของเท็กซ์เจอร์ (Texture) ที่นำมาลงให้ตรง, สีของจุดเวอร์เท็กซ์นั้นๆ หรือสีจากแหล่งกำเนิดแสงต่างๆ เป็นต้น

4) Projection Transformation เป็นการคำนวณเพื่อแปลงสิ่งที่อยู่ในคาเมร่าสเปซ (Camera Space) ให้ไปอยู่ในโปรเจ็กชั่นสเปซ (Projection Space) โดยที่ขั้นตอนนี้จะมีการคำนวณตำแหน่งและขนาดของโมเดลให้ผิดแปลกไปจากเดิม เพื่อที่ว่าเมื่อถูกแปลงให้เห็นเป็นภาพ 2 มิติแล้ว จะได้เห็นเป็นแบบเพอร์สเปคทีฟ (Perspective) คือสิ่งที่อยู่ใกล้ตา จะใหญ่กว่าสิ่งที่อยู่ไกลออกไป ซึ่งจะทำให้มองเห็นเสมือนว่า ภาพ 2 มิตินั้นดูมีความลึก ซึ่งการเขียนโปรแกรมเพื่อทำงานในส่วน Transformation ทั้ง 3 แบบที่กล่าวมานี้จะต้องใช้เมตริกซ์ (Matrix)



รูปที่ 5-12 แสดงการทำโปรเจ็กชั่น ซึ่งจะเห็นโมเดลในมุมมองภาพ 2 มิติที่มีความลึก

5) Clipping and Viewport Scaling เป็นการทำงานที่แทรกก่อนจะไปถึงการทำงานในส่วนของการลงสี (Rasterization) นั่นคือ การตัดส่วนของโมเดลที่ถูกโมเดลอื่นบัง หรือตัดส่วนที่อยู่ด้านหลังออกไป แล้วก็ลดหรือขยายขนาดโมเดลต่างๆ ให้พอดีกับขนาดของความละเอียด (Resoluion) ของหน้าจอที่ใช้ เป็นการช่วยลดการทำงานของขั้นตอนการลงสี คือไม่ต้องไปสนใจสิ่งที่มองไม่เห็นนั่นเอง



รูปที่ 5-13 หลังจากถูกวิวพอร์ตและขริบออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.2 กระบวนการลงสี (Rasterization)

ขั้นสุดท้ายในกระบวนการเรนเดอร์ไปป์ไลน์ จะเป็นการลงสี (Rasterization) โดยตัวที่ทำหน้าที่นี้จะเรียกว่า Rasterizer ซึ่งจะแปลงสิ่งที่อยู่ในโปรเจ็กชั่นสเปซ (Projection Space) ให้ไปอยู่ในสกรีนสเปซ (Screen Space) คือแปลงพิกัดของเท็กเซล (Texel) ต่างๆ ที่เป็นแบบ 3 มิติ ให้เป็น 2 มิติ เมื่อเทียบกับขนาดของจอภาพ หลังจากนี้เราก็จะได้ภาพปรากฏอยู่ในแบ็คบัฟเฟอร์ (Back Buffer) เพื่อที่เราจะได้สั่งให้ไดเร็กทรีดีไวซ์ออบเจ็กท์ (Direct3D Device Object : DDI) จัดการนำข้อมูลในแบ็คบัฟเฟอร์ (Back Buffer) นี้ไปสลับหรือก๊อปปี้เป็นฟร็อนท์บัฟเฟอร์ (Front Buffer) เพื่อให้ภาพไปปรากฏให้เห็นบนจอภาพต่อไป

5.6 ไฟล์ข้อมูลทาง 3 มิติสำหรับไดเร็กซ์เอ็กซ์

คือ ไฟล์ที่มีนามสกุลเป็น x (.x file) โดยเอ็กซ์ไฟล์ได้กำหนดรูปแบบในการเก็บข้อมูลของ เมช (mesh), เท็กซ์เจอร์(texture), อนิเมชั่น(animation) และ วัตถุที่ผู้ใช้ได้กำหนดขึ้นเอง อีกทั้งยังสนับสนุนการตั้งค่าเพื่อการเล่นอนิเมชั่นแบบเวลาจริง(real time) รวมทั้ง การทำ อินสแตนซิ่ง(instancing) และ เฮียราชี่ (hierarchies) การทำอินสแตนซิ่ง คือการที่อ้างอิงข้อมูลหลายๆวัตถุ เช่น เมช โดยทำการบันทึกไว้ในไฟล์เดียว การทำเฮียราชี่คือการอธิบายถึงความสัมพันธ์ระหว่างข้อมูล (data record)

ส่วนโปรแกรมที่ทำการสร้างโมเดลอย่าง Discreet's 3ds max หรือ Alias Wavefront's Maya สามารถที่จะทำการแปลงเป็นเอ็กซ์ไฟล์ได้

ส่วนการติดต่อกับเอ็กซ์ไฟล์นั้น ไดเร็กทรีดีได้มีการสร้างฟังก์ชันไว้หลายฟังก์ชันโดยฟังก์ชันหลักที่ใช้ในการติดต่อ เช่น การโหลด, การเซฟ เป็นต้น

5.7 เมช (mesh)

ดีทรีดีเอ็กซ์ได้ทำสร้างเมชเพื่อ โหลด, จัดการ และ เรนเดอร์ ข้อมูลของ เอ็กซ์ไฟล์ เมชโดยพื้นฐานแล้วเป็นที่เก็บของ เวอร์ไทซ์ที่ใช้กำหนดจีโอเมททรี(geometry) และ เซตของอินไดซ์ที่ทำการกำหนดหน้า

5.7.1 สถาปัตยกรรม ของ เมช

เมชได้มีการบรรจุข้อมูลสำหรับโมเดลที่มีความซับซ้อนไว้ ซึ่งเป็นลักษณะนามธรรมที่เก็บทรัพยากร เช่น เท็กซ์เจอร์ แมททีเรียล และ แอททริบิวต์ เช่น ข้อมูลตำแหน่ง(position data) และแอดจาเซนซี่ดาต้า รวมถึง โอเปอเรชั่นในการพัฒนาประสทิธิภาพในการวาดโมเดล และ ลักษณะของพื้นผิว นอกจากนั้นยังมีลักษณะแนวคิดของเมชอย่างอื่นที่มีผลต่อฟังก์ชันของโอเปอเรชันที่มีต่อเมช การที่เราเข้าใจลักษณะแนวคิดนี้จะสามารถนำไปประยุกต์เพื่อทำให้เกิดการพัฒนาประสิทธิภาพของเมชได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.7.1.1 เมชออปเจกดาต้า(Mesh Object Data)

เมชได้บรรจุไว้ด้วย เวอร์เท็กซ์ บัฟเฟอร์ (Vertex Buffer), อินเด็กซ์บัฟเฟอร์ (Index Buffer) และ แอททริบิวต์ บัฟเฟอร์ (Attribute Buffer)

1) เวอร์เท็กซ์ บัฟเฟอร์ ประกอบไปด้วย เวอร์เท็กซ์ ดาต้า คือ เวอร์ไทซ์ (vertices data) ของ เมชนั้นเอง

2) อินเด็กซ์ บัฟเฟอร์ ประกอบไปด้วย เวอร์เท็กซ์ อินไดซ์ (vertex indices) สำหรับการเข้าถึงข้อมูล เวอร์เท็กซ์ บัฟเฟอร์ ซึ่งเป็นวิธีการลดขนาดของ เวอร์เท็กซ์บัฟเฟอร์ โดยการลดขนาดของเวอร์ไทซ์ที่ซ้ำกัน เฉพาะ อินเด็กซ์เมช (indexed mesh) เท่านั้นที่มีการใช้ อินเด็กซ์บัฟเฟอร์ เช่น ถ้า เมชที่เกิดจากลิสต์ของสามเหลี่ยม จะไม่จำเป็นต้องใช้อินเด็กซ์บัฟเฟอร์

3) แอททริบิวต์ บัฟเฟอร์ ประกอบด้วย แอททริบิวต์ดาต้า แอททริบิวต์ คือ คุณสมบัติของ เมช เวอร์ไทซ์ ซึ่งไม่ได้มีส่วนร่วมกับ การเก็บข้อมูลของเมชใน ดีทรีดีเอ็กซ์ในกลุ่มของแอททริบิวต์สำหรับแต่ละหน้า

5.7.1.2 แอพพลิเคชันดาต้า (application data)

แอพพลิเคชันดาต้า คือ ข้อมูลของเมชที่มีการจัดการโดย แอพพลิเคชัน

1) แมททีเรียล บัฟเฟอร์ ประกอบไปด้วย n แมททีเรียล ซึ่งจะให้ค่าจากการคืนค่าของฟังก์ชันโหลด เมื่อ เอ็กซ์ไฟล์ ถูกโหลด แต่ละ ซับเซต (subset) สามารถที่จะมี แมททีเรียล และเท็กซ์เจอร์ เป็นของตนเองได้ แมททีเรียลบัฟเฟอร์นั้นเป็นแบบ สเตติก(static)

2) แอดจาเซนซี่ บัฟเฟอร์ ประกอบไปด้วยข้อมูลเกี่ยวกับ ขอบ, หน้า และ หน้าใกล้เคียง บางโอเปอเรชันของเมชนั้นขึ้นอยู่กับการที่รู้หน้าที่อยู่ชิดกับหน้าอื่น ซึ่งข้อมูลเหล่านี้เรียกว่า แอดจาเซนซี่ดาต้า ซึ่งจะถูกเก็บไว้ใน แอดจาเซนซี่บัฟเฟอร์ โดยที่ตัวมันเองไม่ใช่ส่วนหนึ่งของเมช แต่ถูกจัดเก็บโดย แอพพลิเคชัน และจะต้องถูก ส่งให้กับ วิธีของเมช เมื่อจำเป็น

3) เอฟเฟกต์ อินสแตนซ์บัฟเฟอร์ ประกอบไปด้วย ลิสต์ของ เอฟเฟกต์ อินสแตนซ์ เอฟเฟกต์อินสแตนซ์จะถูกเก็บในรูปแบบสถานะ ซึ่งเป็นข้อมูลของสถานะที่ใช้ในการเริ่มต้นไปป์ไลน์ (pipeline)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.7.2 ฟังก์ชันของเมช

ID3DXBaseMesh ถูกสร้างขึ้นมาเพื่อติดต่อ จัดการ กับ เมช โดยมีฟังก์ชันหลักๆดังนี้

- CloneMesh โคลนเมชโดยใช้การประกาศตัวแปร
- CloneMeshFVF โคลนเมชโดยใช้ เอฟวีเอฟ(FVF : flexible vertex format)
- ConvertAdjacencyToPointReps แปลงข้อมูลของ แอดจาเซนซี่เมช ไปเป็น อาร์เรย์ของจุดที่ใช้แสดง
- ConvertPointRepsToAdjacency แปลงข้อมูลที่เป็นจุดที่ใช้แสดงเป็นข้อมูล แอดจาเซนซี่เมช
- DrawSubset วาดซับเซตของเมช
- GenerateAdjacency สร้างลิสต์ของขอบและหน้า ของ เมช ที่ใช้ขอบร่วมกัน
- GetAttributeTable เรียกข้อมูล ตารางแอททริบิวต์ของเมช หรือ จำนวนของรายการที่อยู่ในตารางแอททริบิวต์สำหรับเมช
- GetDeclaration เรียกเวอร์ไทซ์ที่ถูกประกาศในเมช
- GetDevice เรียกค่าของดีไวซ์ที่ใช้กับเมช
- GetFVF เรียก ค่าฟังก์ชันเวอร์เท็กซ์ ที่ได้กำหนดไว้
- GetIndexBuffer เรียกค่าใน อินเด็กซ์บัฟเฟอร์
- GetNumBytesPerVertex เรียกค่าจำนวนของไบต์ของแต่ละเวอร์เท็กซ์
- GetNumFaces เรียกค่าของจำนวนของหน้าในเมช
- GetNumVertices เรียกค่าของจำนวนเวอร์ไทซ์ในเมช
- GetOptions เรียกค่าของตัวเลือกของเมชที่เรียกใช้ในการสร้างเมช
- GetVertexBuffer เรียกข้อมูลใน เวอร์เท็กซ์บัฟเฟอร์
- LockIndexBuffer ทำการล็อคข้อมูล อินเด็กซ์บัฟเฟอร์และเรียกค่าของพอยต์เตอร์ที่ชี้ไปยังหน่วยความจำที่เก็บ อินเด็กซ์บัฟเฟอร์
- LockVertexBuffer ทำการล็อคข้อมูล เวอร์เท็กซ์บัฟเฟอรีและเรียกค่าของพอยต์เตอร์ที่ชี้ไปยังหน่วยความจำที่เก็บ เวอร์เท็กซ์บัฟเฟอร์
- UnlockIndexBuffer ทำการปลดล็อก อินเด็กซ์บัฟเฟอร์
- UnlockVertexBuffer ทำหารปลดล็อค เวอร์เท็กซ์บัฟเฟอร์
- UpdateSemantics เป็นวิธีการที่ให้ผู้ใช้เปลี่ยนแปลงการประกาศเมชโดยที่ไม่ต้องเปลี่ยนแปลงโครงร่างของเวอร์เท็กซ์บัฟเฟอร์ ซึ่งสามารถทำได้ก็ต่อเมื่อ เวอร์เท็กซ์เก่าและใหม่ มีขนาดเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การออกแบบระบบ

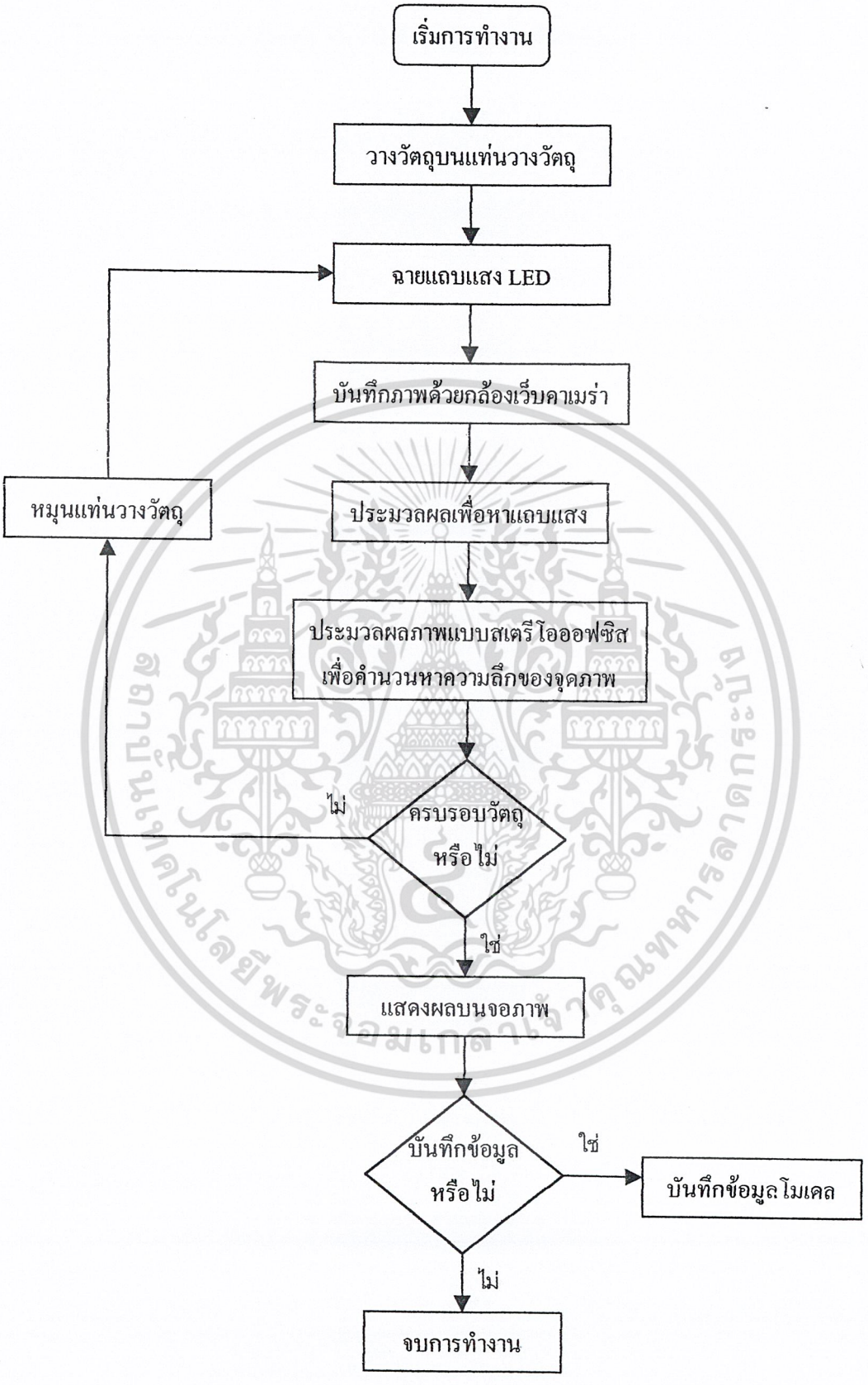
6.1 การออกแบบระบบโดยรวม

ระบบสร้างภาพคอมพิวเตอร์แบบสามมิติโดยอาศัยข้อมูลระยะห่างกล้องกับวัตถุเป็นระบบที่มีส่วนประกอบทั้งส่วนที่เป็นฮาร์ดแวร์และซอฟท์แวร์ ซึ่งจะทำงานติดต่อกันตลอดเวลาโดยตัวซอฟท์แวร์หรือแอพพลิเคชันที่ทำงานอยู่บนเครื่องคอมพิวเตอร์จะเป็นตัวสั่งการให้ฮาร์ดแวร์ทำงานตามขั้นตอน ส่วนฮาร์ดแวร์นั้นจะรับอินพุตจากแอพพลิเคชันอย่างเดียวไม่มีการส่งค่าเอาท์พุตกลับ

การทำงานจะเริ่มต้นจากในส่วนฮาร์ดแวร์ ซึ่งจะเป็นการควบคุมการหมุนของแท่นหมุนวัตถุและควบคุมการฉายแถบแสงของแหล่งกำเนิดแสงLED หลังจากนั้นกล้องก็จะทำการบันทึกภาพของวัตถุเพื่อส่งไปให้ยังส่วนของซอฟท์แวร์ โดยที่ส่วนของซอฟท์แวร์จะมีส่วนประกอบที่สำคัญคือ การประมวลผลภาพเพื่อหาแถบแสงโดยใช้วิดีโอโอซีเอ็กซ์ แล้วจึงส่งข้อมูลต่อเพื่อ ประมวลผลให้ได้ข้อมูลทางสามมิติโดยใช้วิธีสเตรีโอออฟซิส เมื่อหาข้อมูลทางด้านสามมิติได้ครบทั่ววัตถุแล้ว จึงส่งต่อให้ขั้นตอนการแสดงผลบนจอภาพกับการบันทึกข้อมูลโมเดลของวัตถุให้เป็น .x โดยใช้ไดเร็กซ์เอ็กซ์

ขั้นตอนการทำงาน จะมีลักษณะดังรูปที่ 6-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



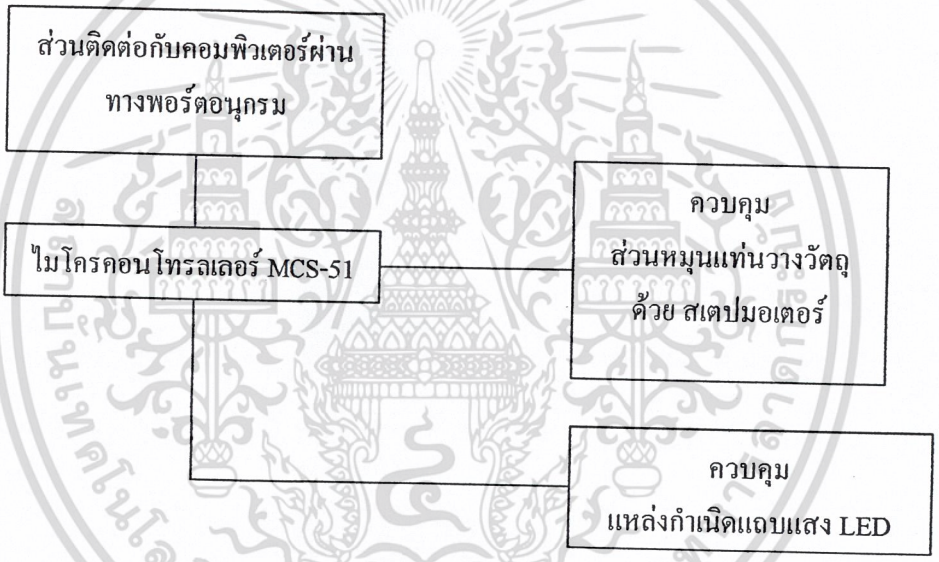
รูปที่ 6-1 รูปแสดงขั้นตอนการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 การออกแบบฮาร์ดแวร์

การทำงานเริ่มจากเครื่องคอมพิวเตอร์จะส่งข้อมูลมาจากโปรแกรม โดยจะสั่งการทำงานผ่านทางพอร์ตอนุกรม (Serial Port) จากนั้นข้อมูลจะถูกประมวลผลภายในไมโครคอนโทรลเลอร์ MCS-51 แล้วส่งไปทำงานยังส่วนต่างๆ คือ แหล่งกำเนิดแสงจะมีหน้าที่ในการส่องแสงทาบบนตัววัตถุจากนั้นกล้องวิดีโอจะทำการบันทึกภาพวัตถุที่มีแถบแสงอยู่แล้วส่งข้อมูลไปทำการประมวลผล ตัวหมุนวัตถุก็จะหมุนเปลี่ยนไปยังตำแหน่งถัดไปเพื่อทำการบันทึกภาพต่อไป

จากการทำงานที่กล่าวมานั้นทำให้สามารถที่จะออกแบบฮาร์ดแวร์ของระบบสร้างภาพคอมพิวเตอร์แบบสามมิติโดยอาศัยข้อมูลระยะห่างระหว่างกล้องกับวัตถุออกแบบเป็นสี่ส่วนหลักดังรูป 6-2



รูปที่ 6-2 ภาพแสดงการติดต่อในแต่ละส่วน

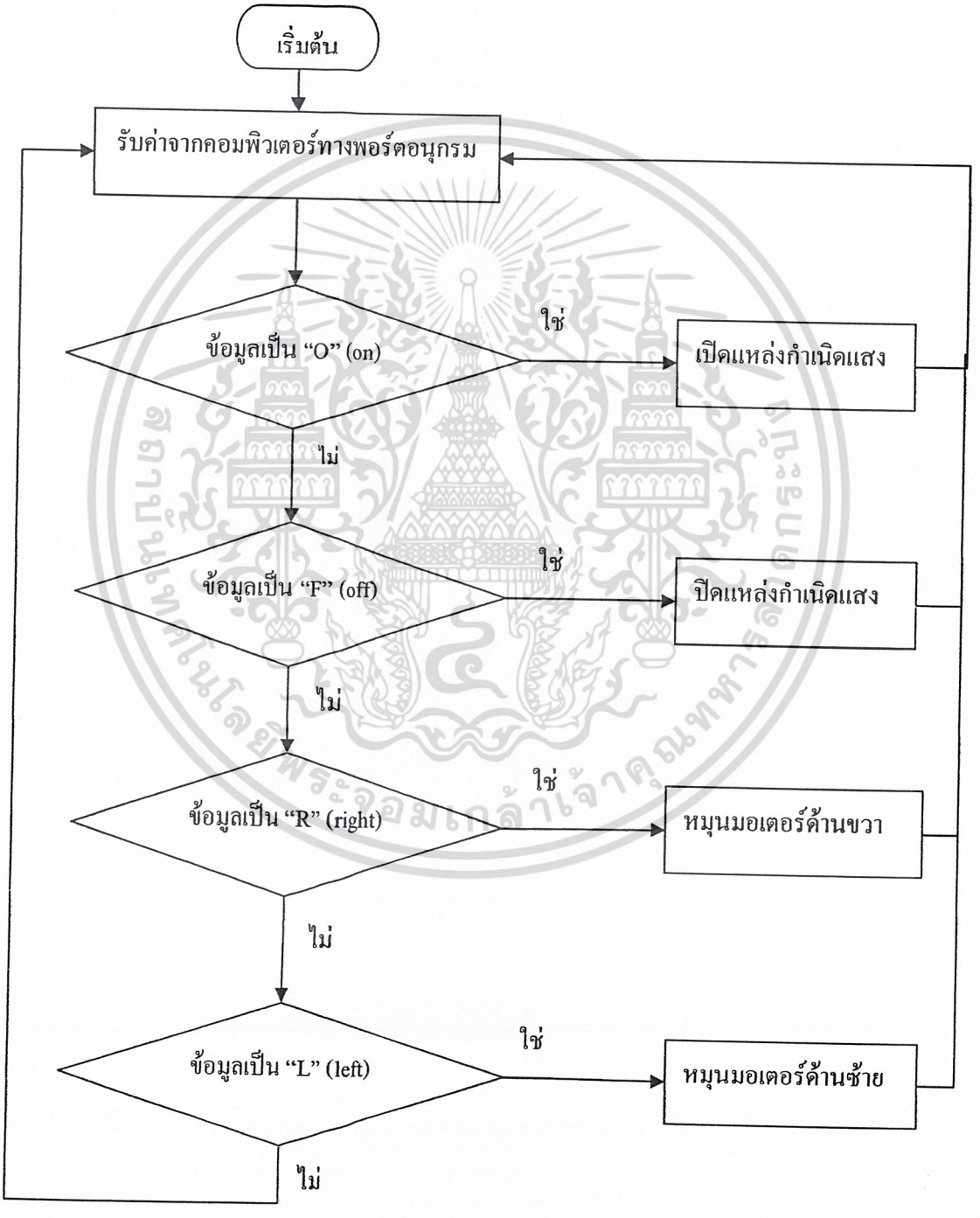
6.2.1 ส่วนควบคุมด้วยไมโครคอนโทรลเลอร์ MCS-51

สำหรับส่วนควบคุมจะมีไมโครคอนโทรลเลอร์ (MCS-51) เป็นศูนย์กลางในการประมวลผลเพื่อติดต่อกับส่วนอื่น โดย พอร์ต 0 ไม่มีการใช้งานจึงปล่อยลอยไว้ สำหรับพอร์ต 1 จะใช้ทำการติดต่อกับแหล่งกำเนิดแสงเส้นตรง ในพอร์ต 2 เป็นส่วนสำคัญที่ใช้ติดต่อกับสเตปมอเตอร์เพื่อหมุนวัตถุ ในพอร์ต 3 จะใช้เพียง P3.0 และ P3.1 ซึ่งเป็นขาในการรับส่งข้อมูลทางพอร์ตอนุกรม RxD และ TxD ตามลำดับ

ในส่วนของสัญญาณนาฬิกาที่ใช้จะทำการต่อ คริสตอล 11.0592 MHz เพื่อทำหน้าที่สร้างสัญญาณนาฬิกาให้กับไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

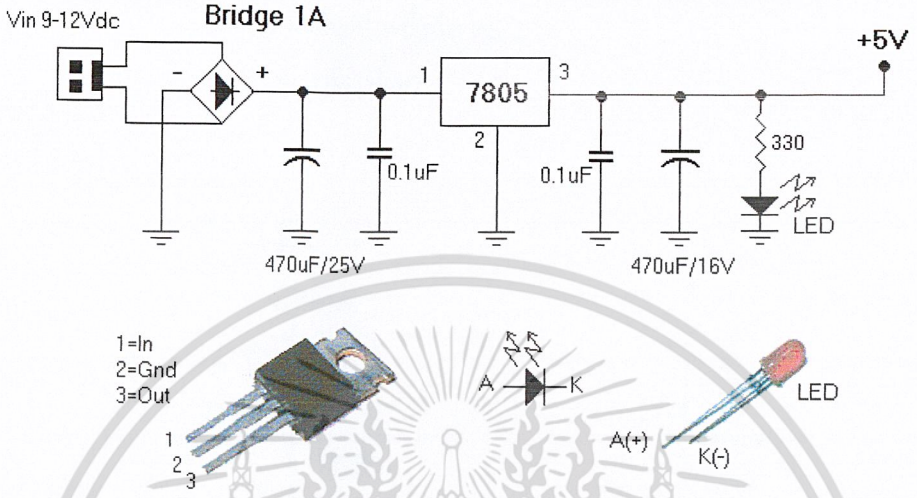
ในส่วนของการโปรแกรม MCS-51 จะใช้ภาษาซี โปรแกรมที่ใช้พัฒนาคือ Keil uVision2 โดยลักษณะการเขียนโปรแกรมทั่วไปจะคล้ายกับภาษาซีปกติแต่ตัวโปรแกรมนี้จะอ้างถึง รีจิสเตอร์ (Register) บางตัวได้โดยไม่ต้องประกาศตัวแปรนั้นไว้ เช่น P1 จะหมายถึงพอร์ต 1 ทั้งหมด อีกทั้งยังสามารถสร้างไฟล์ .hex ซึ่งเป็นรูปแบบไฟล์ที่ใช้ในการโปรแกรมข้อมูลลงบน MCS-51 ได้ด้วย การทำงานของ MCS-51 ที่ได้โปรแกรมจะมีการทำงานดังรูป 6-3



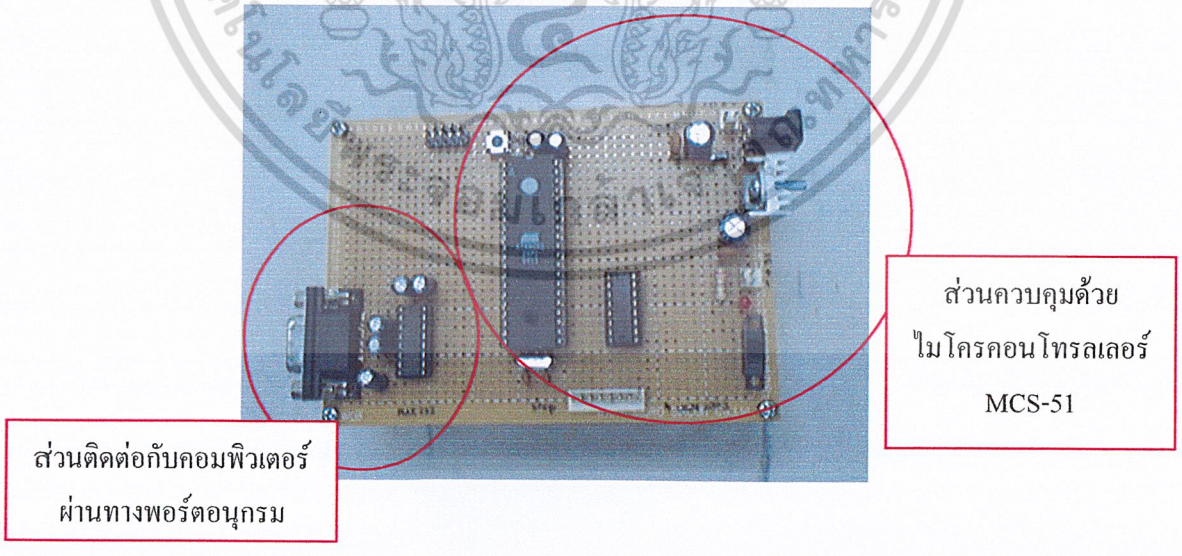
รูปที่ 6-3 ภาพแสดงกระบวนการทำงานของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรที่สำคัญอีกวงจรหนึ่งในส่วนนี้คือ วงจรแปลงแรงดันไฟฟ้าจาก 12 โวลท์ เป็น 5 โวลท์ โดย IC ที่ใช้ในการทำงานคือ 7805 ที่ต้องมีวงจรนี้จากแหล่งจ่ายไฟที่ใช้ในระบบเป็นหม้อแปลงไฟฟ้า 12 โวลท์ แต่ไอซี ทั่วไปจำเป็นต้องใช้แรงดันที่ 5 โวลท์

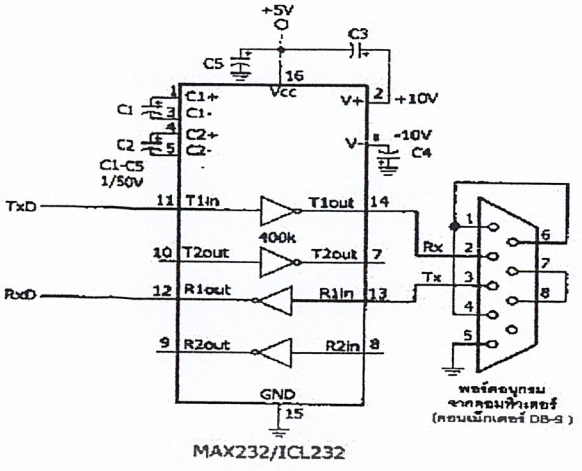


รูปที่ 6-4 แสดงวงจรแปลงแรงดันไฟ 12 โวลท์ เป็น 5 โวลท์



รูปที่ 6-5 ภาพส่วนควบคุมด้วยไมโครคอนโทรลเลอร์ MCS-51
และ ส่วนติดต่อกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-7 แสดงการต่อวงจรรับส่งข้อมูลผ่านทางพอร์ตอนุกรม

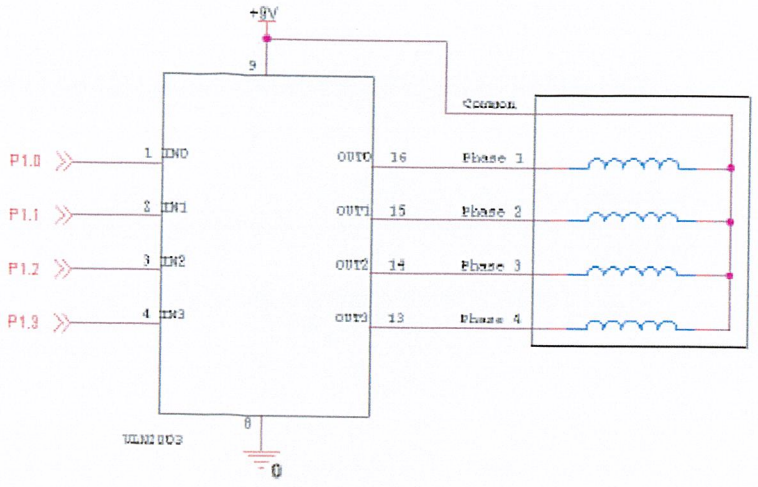
ในส่วนของการโปรแกรมเพื่อใช้งาน Serial port นั้นจะใช้ Mode 1 ในการทำงานที่ ค่า Baudrate เท่ากับ 9600 โดยคำสั่งพื้นฐานอย่าง printf จะสามารถส่งข้อความจาก MCS-51 มาแสดงบนคอมพิวเตอร์ได้

6.2.3 ส่วนหมุนแท่นวางวัตถุด้วยสเตปมอเตอร์

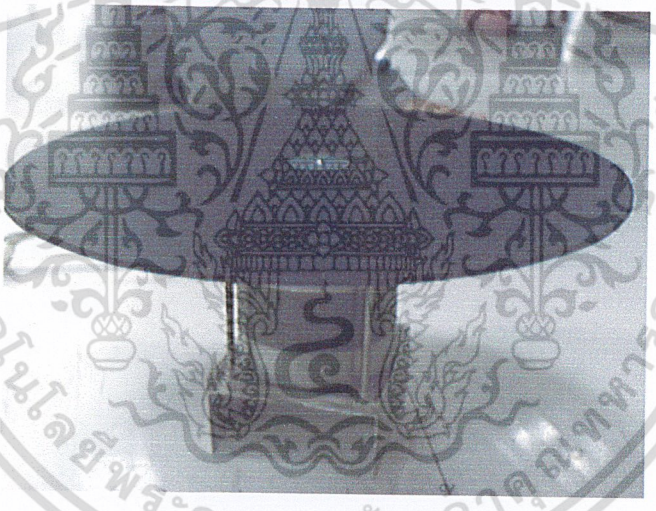
ส่วนเป็นส่วนสำคัญอีกส่วนหนึ่งเนื่องจากความละเอียดในการเก็บข้อมูลจากวัตถุจะขึ้นอยู่กับส่วนนี้ทั้งหมด ซึ่งถ้าหากการควบคุมไม่ดีจะทำให้เกิดข้อผิดพลาดของข้อมูลได้ด้วย

สำหรับส่วนหมุนวัตถุนี้ส่วนสำคัญคือ สเตปมอเตอร์ และ ไอซี ULN 2003 สเตปมอเตอร์ที่ใช้ในระบบมีรายละเอียดคือ องศาต่อการหมุน 1 ครั้งคือ 1.8 องศา ใช้แรงดัน 12 โวลท์ 4 เฟส ดังนั้นสามารถเก็บข้อมูลจากวัตถุได้ละเอียดที่สุดคือ 400 ภาพรอบวัตถุ ถ้าขับมอเตอร์นี้ด้วยวิธีแบบครึ่งสเตป (Half step) แต่ในระบบจะขับด้วยวิธีแบบฟูลสเตป 1 เฟส ทำให้ได้ความละเอียดที่ 200 ภาพ ซึ่งเพียงพอต่อวัตถุขนาดเล็กแล้ว สำหรับไอซี ULN 2003 เป็นไอซี ที่ใช้ในการขับสเตปมอเตอร์ตั้งแต่ 5-30 โวลท์ จ่ายกระแสได้สูงสุดที่ 500 มิลลิแอมป์ วงจรจะเป็นดังรูป 6-8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-8 วงจรขับสเตปมอเตอร์โดยใช้ ULN 2003



รูปที่ 6-9 ภาพส่วนหมุนแท่นวางวัตถุด้วยสเตปมอเตอร์

ส่วนการโปรแกรมเพื่อขับสเตปมอเตอร์นั้นจะต้องทำการวนจ่าย 1 ไล่ไปทีละเฟส ของมอเตอร์ เมื่อครบทั้งหมด 4 เฟส มอเตอร์เคลื่อนที่ไป 1 สเตป โดยการจ่ายค่าแต่ละครั้งต้องเหมาะสมกับการทำงานของมอเตอร์ด้วย ไม่เช่นนั้นมอเตอร์จะไม่เคลื่อนที่ ส่วนหมุนวัตถุนี้จะต่ออยู่กับ P2 ตั้งแต่บิต 0 ถึง 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.4 ส่วนแหล่งกำเนิดแถบแสง LED

ส่วนนี้ใช้เพื่อสร้างแถบแสงทาบลงบนวัตถุเพื่อนำไปผ่านกระบวนการประมวลผลภาพต่อไป โดยอุปกรณ์ที่ใช้คือ ไดโอดเปล่งแสง (Light Emitting Diode: LED) ลักษณะการทำงานในส่วนนี้จะใช้ LED สีแดงนำมาต่อเรียงกันเป็นเส้นตรงบนแผงวงจรดังในวงกลมรูปที่ 6-10

รูปที่ 6-10 แสดงวงจรภายในส่วนแหล่งกำเนิดแถบแสง LED

จากนั้นนำอะลูมิเนียมมากั้นแสงให้ผ่านเป็นเส้นตรง โดยพยายามให้แสงมีความแคบมากที่สุด ลักษณะดังรูป

รูปที่ 6-11 แสดงส่วนหน้าของส่วนแหล่งกำเนิดแถบแสง LED

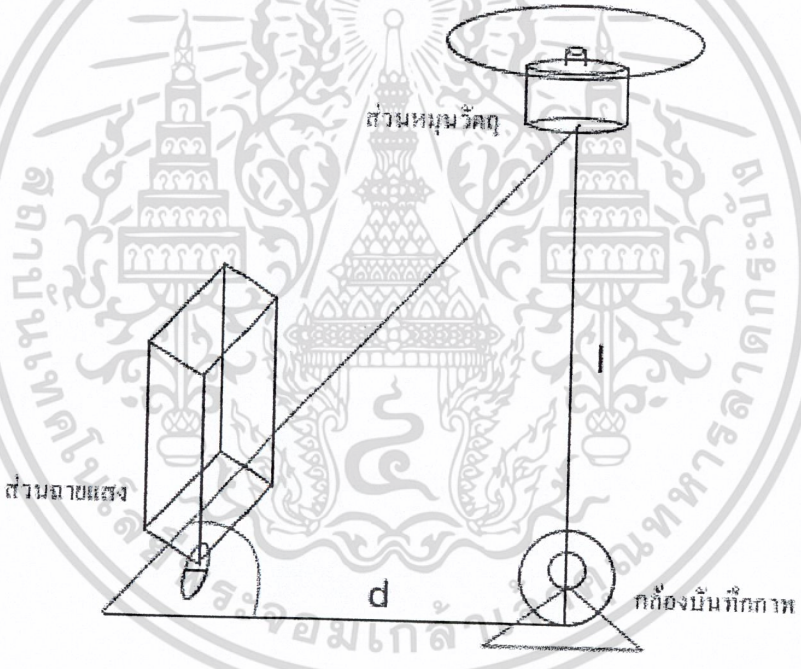
การปิดเปิดแสงทำได้โดยการจ่ายค่า 0และ 1 ออกพอร์ต 1 ของ MCS-51 ทั้งหมดโดยค่า 0 จะเป็นการเปิด ส่วน ค่า 1 จะเป็นการปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การจัดวางตำแหน่งของฮาร์ดแวร์

ทุกส่วนในฮาร์ดแวร์จำเป็นต้องมีการวางในตำแหน่งที่ถูกต้อง เนื่องจากตำแหน่งเหล่านั้นจะมีผลต่อกระบวนการสเตรีโอออฟซิส เพราะต้องใช้ค่าระยะห่างระหว่างส่วนต่างในการประมวลผล โดยการวางตำแหน่งนั้น อุปกรณ์ 3 ส่วนที่ต้องคำนึงถึงคือ มอเตอร์ ส่วนกำเนิดแสง และ ตัวกล้องบันทึกภาพ

มอเตอร์จะเป็นส่วนที่ใช้ในการวางวัตถุเพื่อทำการหมุนไปทีละสเตป มีจุดที่ต้องคำนึงถึงคือจุดศูนย์กลางการหมุน โดยจะมีส่วนกำเนิดแสงฉายแสงไปยังจุดศูนย์กลางนั้น การวางส่วนกำเนิดแสงนี้เราจะคำนึงถึงมุมที่ทำกับเส้นที่ตั้งฉากจากตัวกล้อง ส่วนตัวกล้องนั้นจะต้องจัดในแนวของกล้องขนานกับพื้นโลก โดยวิธีการนี้จะเรียกว่าการตั้งค่าเริ่มต้นของกล้อง (calibration) ซึ่งจะกล่าวต่อ การจัดวางตำแหน่งจึงมีลักษณะดังรูป 6-12



รูปที่ 6-12 แสดงการจัดวางตำแหน่งของฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.1การจัดวางตำแหน่งของกล้อง (Camera's Position Calibration)

การจัดวางตำแหน่งของกล้องนี้เป็นขั้นตอนที่จำเป็นต่อการทำสเตรีโอออฟซิส เป็นขั้นตอนในการกำหนดค่าที่จำเป็นต่อการทำสเตรีโอออฟซิส จัดตำแหน่งกล้องให้อยู่ในระดับขนานกับพื้นโลก โดยวิธีการที่ใช้คือจะสร้างฉากรับภาพที่ตั้งฉากกับพื้นโลกเพื่อนำมาวางไว้ที่หน้ากล้อง (เนื่องจากกล้องต้องกำหนดตำแหน่งให้คงที่ การเคลื่อนย้ายไปตั้งค่าเริ่มต้อนของกล้องที่อื่นจึงไม่เหมาะสม) โดยฉากนี้มีรูปสีเหลี่ยมผืนผ้าวาดอยู่การตั้งค่าเริ่มต้นของกล้องสามารถทำได้โดยการปรับกล้องให้สีเหลี่ยมผืนผ้านี้พอดีกับภาพที่กล้องรับได้พอดี หรือ พยายามให้จุดสีดำที่มุมสี่เหลี่ยมฉากอยู่ที่มุมของจอภาพ

รูปที่ 6-13 แสดงการตั้งค่าเริ่มต้นของกล้องบันทึกภาพ

จากภาพจะว่าขอบสีเงินซึ่งเป็นส่วนของรูปสี่เหลี่ยมผืนผ้าที่อยู่บนฉากได้ถูกปรับให้พอดีกับขอบของภาพที่กล้องสามารถรับได้

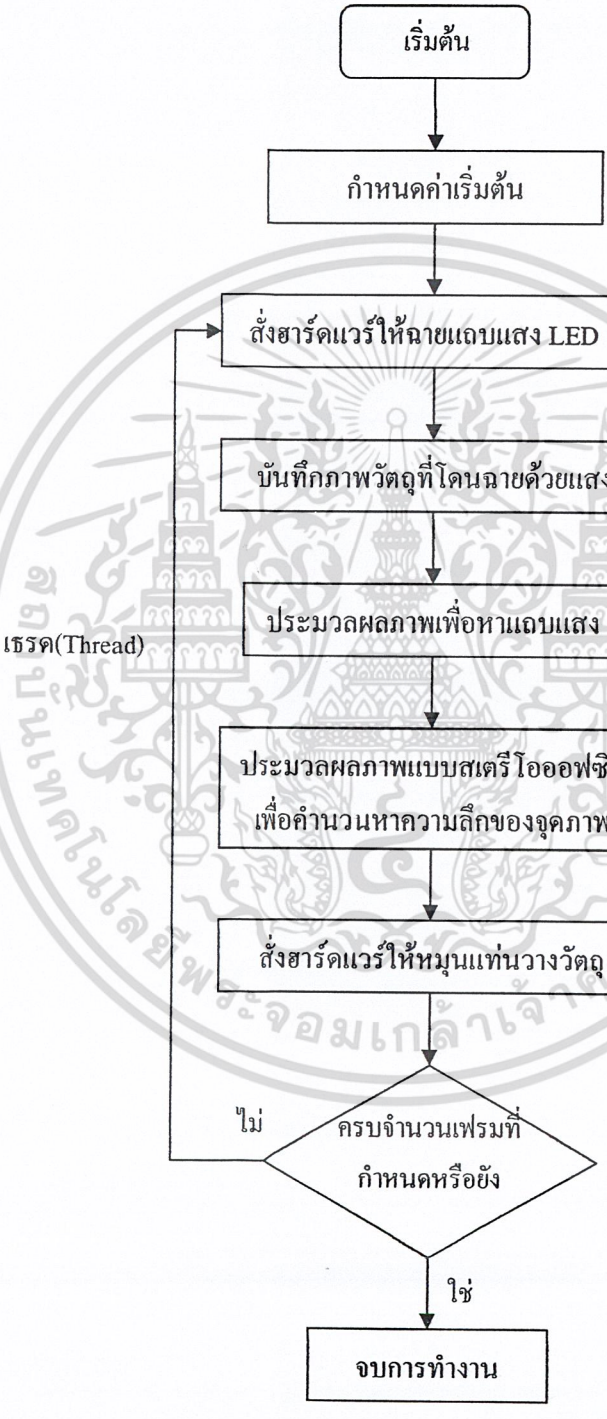
6.4 การออกแบบแอพพลิเคชัน(ส่วนแปลงข้อมูล)

กระบวนการนี้จะทำตั้งแต่รับภาพจากกล้องวิดีโอจากนั้นจะส่งไปทำการประมวลผลภาพแล้วนำเอาท์พุตที่ได้ไปผ่านสมการที่ได้จากวิธีการสเตรีโอออฟซิสจนได้ค่าบนพิกัด xyz

กระบวนการในส่วนนี้จะทำภายในเครื่องคอมพิวเตอร์ซึ่งต่อพ่วงด้วยฮาร์ดแวร์ผ่านทางพอร์ตอนุกรม โดยภาษาที่ใช้ในการพัฒนาคือ วิช่วลซีพลัสพลัส และ เครื่องมือที่ใช้ในการควบคุมกล้องและทำการประมวลผลภาพคือวิดีโอโอซีเอ็กซ์ (VideoOCX) ในส่วนของวิช่วลซีพลัสพลัสจะใช้ MFC dialog based เป็นโปรเจคในการพัฒนา โดยจะทำการรวมเอาโอซีเอ็กซ์เข้าไปในลักษณะของแอคทีฟเอ็กซ์ (ActiveX) บนโปรเจค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานจะสร้างตัวแปรหนึ่งขึ้นมาควบคุมกล้อง (Handle) โดยการติดต่อกับกล้องจะกระทำผ่านตัวแปรนี้ และสร้างตัวแปรขึ้นมารองรับภาพที่ได้จากการจับภาพนิ่งจากกล้องในลักษณะของ อิมเมจ แฮนด์เดล ซึ่งสามารถทำการประมวลผลภาพผ่านทางตัวควบคุมนี้ เนื่องจากตัวควบคุมนี้สามารถเข้าถึงข้อมูลในแต่ละพิกเซล (Pixel) ได้



รูปที่ 6-14 แสดงการทำงานในส่วนแอพพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เธรด (Thread) เป็นส่วนหนึ่งของโปรแกรมที่จะดำเนินอยู่ตลอดเวลาแม้ไม่ได้รับการสั่งหรือเรียกจากส่วนอื่นๆ

กระบวนจะดำเนินไปตามภาพโดยเริ่มต้นจะเป็นกำหนดค่าสร้างตัวแปรขึ้นมารองรับการทำงานของกล้องวิดีโอ จากนั้นจะเริ่มการเรียกใช้เธรดโดยเธรดจะทำงานอยู่เบื้องหลังการทำงานทั้งหมดของแอพพลิเคชัน เมื่อมีการสั่งให้เริ่มสแกนวัตถุ แอพพลิเคชันจะส่งข้อมูลออกทางพอร์ตอนุกรมไปยังไมโครคอนโทรลเลอร์เพื่อสั่งให้ทำการฉายแถบแสงไปยังวัตถุ จากนั้นกล้องจะทำการบันทึกภาพวัตถุขณะที่โดนฉายด้วยแถบแสง แล้วจึงนำภาพที่ได้ไปผ่านการประมวลผลภาพ ผลลัพธ์ที่ได้จะเหลือเพียงภาพที่มีเส้นของแถบแสงที่ฉายบนวัตถุเพียงเส้นเดียว แล้วจึงนำไปเข้าสู่การหาค่าทางสามมิติด้วยวิธีการสเตรีโอออฟซิส แล้วนำค่าที่ได้เก็บไว้ในตัวแปรแบบอาร์เรย์แล้วส่งค่าให้กระบวนการต่อไป

6.5 การออกแบบการประมวลผลภาพ

จากการออกแบบแอพพลิเคชันที่ผ่านมาจะเห็นว่าส่วนสำคัญที่สุดในส่วนนี้คือ การประมวลผลภาพ เนื่องจากว่าข้อมูลที่ได้มานั้นฮาร์ดแวร์เป็นส่วนที่ควบคุมสภาวะแวดล้อมได้ยาก เมื่อข้อมูลมาถึงส่วนประมวลผลภาพ การทำให้ได้ข้อมูลที่ถูกต้องที่สุดเพื่อนำไปคำนวณต่อไปจึงเป็นสิ่งสำคัญในส่วนนี้ ดังนั้นจำเป็นต้องหาวิธีการที่ดีที่สุดที่จะใช้ในการทำงานในส่วนนี้

จากข้อมูลที่ได้รับจากกล้องวิดีโอจะเห็นว่าสิ่งที่เราต้องการให้เป็นเอาท์พุตสุดท้ายนั้นคือ เส้นแสงที่ทาบอยู่บนวัตถุ ดังนั้นขั้นตอนในการทำการประมวลผลภาพจะเป็นดังนี้

6.5.1 การประมวลผลภาพด้วยการเปรียบเทียบส่วนที่แตกต่าง

ขั้นตอนนี้เป็นขั้นตอนแรกในการทำงานโดยวิธีนี้ใช้เพื่อลบพื้นหลังของภาพออกไปและกำหนดขอบเขตที่เราต้องการจะคำนวณ โดยเริ่มแรกจะต้องทำการเก็บภาพของพื้นหลังก่อน หลังจากนั้นทุกครั้งที่มีการเปลี่ยนตำแหน่งของวัตถุ ภาพที่เก็บได้จะถูกนำมาเปรียบเทียบกับภาพของพื้นหลัง โดยจะใช้การเข้าถึงข้อมูลในระดับพิกเซล แล้วนำค่าของสีในแต่ละพิกเซลมาเปรียบเทียบ ผลที่ได้ถ้ามีความใกล้เคียงกันจะลบส่วนนั้นทิ้งไป ส่วนตำแหน่งที่แตกต่างกันจะคงค่าสีเดิมของภาพวัตถุไว้ ทำให้ผลลัพธ์ที่ได้จะเหลือเพียงวัตถุเท่านั้น

วิธีการคำนวณความแตกต่างของค่าสีนั้น จำเป็นต้องกำหนดช่วงของค่าคงที่ขึ้นมาหนึ่งค่า (Background Removal) เพื่อใช้ในการเปรียบเทียบหลังจากการลบค่าสีในแต่ละพิกเซลแล้ว โดยถ้าค่าอยู่นอกช่วงนี้จะทำการคงค่าสีไว้ ดังนั้นการกำหนดช่วงค่าคงที่ให้ถูกต้องจึงเป็นสิ่งสำคัญ เพราะถ้ากำหนดไว้กว้างส่วนของวัตถุจะถูกถือว่าเป็นพื้นหลังไปด้วย และถ้ากำหนดไว้แคบสิ่งรบกวน (Noise) จะมีผลต่อระบบมากขึ้น

หลังจากทำการเปรียบเทียบส่วนที่แตกต่างแล้วขอบเขตที่ได้จะถูกนำไปใช้ในการประมวลผลภาพถัดไปเพื่อกำหนดขอบเขตให้อยู่ในเฉพาะส่วนของวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5.2 การประมวลผลภาพด้วยการลบภาพ

หลังจากที่เรารู้ขอบเขตของภาพที่เราต้องทำการประมวลผลแล้วส่วนนี้จะเป็นส่วนที่ทำให้ข้อมูลของภาพเหลือเพียงส่วนที่เราต้องการ โดยการทำงานจะต้องเก็บภาพอีกหนึ่งภาพโดยภาพนี้เป็นภาพของวัตถุที่ตำแหน่งเดิมแต่จะมีส่วนที่เพิ่มขึ้นมาคือแถบแสงที่ทาบอยู่บนพื้นผิว จากนั้นจะนำภาพนี้ไปลบกับภาพเก็บไว้ในครั้งก่อน (ก่อนยิงแสง) โดยกำหนดขอบเขตของการลบภาพให้อยู่ในช่วงที่หาได้จากกระบวนการข้างต้น

การลบภาพจะทำในลักษณะที่คล้ายกับการเปรียบเทียบความแตกต่างโดยจะนำข้อมูลในระดับพิกเซล คือสีของแต่ละพิกเซลมาลบกันผลลัพธ์ที่ จะเป็นภาพที่เหลือเพียงเส้นแสงเท่านั้น แต่จะมีสิ่งรบกวนอื่นด้วยเนื่องจากการจับภาพในตำแหน่งเดียวกันในเวลาที่ต่างกัน ค่าของสีจะแตกต่างไปตามแสงที่ได้ จึงต้องนำผลลัพธ์ที่ได้จากกระบวนการนี้ไปผ่านการประมวลผลภาพถัดไปเพื่อทำการลดสิ่งรบกวนที่เกิดขึ้น

6.5.3 การลดสิ่งรบกวน

การลดสิ่งรบกวนจะทำโดยการประมวลผลภาพแบบเทรชโฮลด์ซึ่งสิ่งรบกวนที่ติดมากับภาพที่ได้จากลบภาพข้างต้นจะถูกกำจัดออกไป โดยระบบนี้จะใช้การเทรชโฮลด์แบบค่าคงที่ ดังนั้นค่าคงที่ที่กำหนดขึ้นควรจะมีค่าที่อยู่ในระดับที่ทำให้สิ่งรบกวนหมดไปแต่ไม่มีผลกระทบกับส่วนที่ระบบต้องการ จากการทดลองค่าเทรชโฮลด์ควรตั้งให้อยู่ระหว่าง 40-120 ซึ่งค่าที่ตั้งนั้นจะขึ้นอยู่กับลักษณะของแถบแสงที่ฉายลงบนวัตถุว่ามีชัดเจนมากน้อยเท่าไร หากชัดเจนมากก็สามารถที่จะตั้งค่าเทรชโฮลด์ได้มาก สามารถลดสิ่งรบกวนออกไปได้มาก แต่ถ้าแถบแสงจางจะตั้งค่าเทรชโฮลด์ได้น้อย มิฉะนั้นจะไม่สามารถหาเส้นแสงออกมาได้ แต่ก็จะลดสิ่งรบกวนออกได้น้อย หรือทำให้เกิดสิ่งรบกวนมากขึ้นกว่าเดิม

หลังจากทำการลดสิ่งรบกวนด้วยเทรชโฮลด์เสร็จเรียบร้อยแล้วเราจะได้ภาพที่เหลือแถบแสงที่เราต้องการแต่เนื่องจากว่าแหล่งกำเนิดแสงของเรามีลักษณะกระจายแสงดังนั้น ถึงแม้ว่าจะผ่านการกั้นให้เส้นแสงมีขนาดเล็กที่สุด แต่แถบที่ได้ก็ยังไม่สามารถนำไปใช้คำนวณต่อไปได้ จึงต้องผ่านเข้าสู่ขั้นตอนสุดท้าย

6.5.4 การคำนวนหาตำแหน่งศูนย์กลาง

การหาตำแหน่งศูนย์กลางเป็นการหาศูนย์กลางของแถบแสง เนื่องจากเราต้องการเส้นที่มีขนาดเพียง 1 พิกเซลเพื่อนำแต่ละพิกเซลไปคำนวณหาตำแหน่งที่แท้จริงต่อไป

วิธีการที่ใช้ในการหาจะใช้การหาค่าเฉลี่ยของพิกเซลทั้งหมดแนวแกนนั้นโดยหาได้จาก ผลรวมของค่าพิกัด X ของพิกเซลที่อยู่บนเส้น หารด้วย จำนวนพิกเซลที่อยู่บนเส้นทั้งหมด

ผลลัพธ์สุดท้ายที่ได้จะเป็นภาพที่เหลือเพียงเส้นแสงขนาด 1 พิกเซลพร้อมที่จะนำไปผ่านกระบวนการสเตรีโอออฟซิสเพื่อหาตำแหน่งพิกัด xyz ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.6 สมการสำหรับหาข้อมูลจุดภาพสามมิติของวัตถุ

การหาสมการที่จะนำมาใช้ในระบบสร้างภาพคอมพิวเตอร์แบบสามมิติโดยอาศัยข้อมูลระยะห่างกล้องกับวัตถุ จะใช้หลักการของสเตรีโอออฟซิส ซึ่งเป็นวิธีการในการหาระยะของวัตถุดังที่ได้กล่าวข้างต้น โดยสำหรับระบบนี้เราได้ใช้วิธีการแบบ Active Stereo System ทำให้ได้สมการที่ใช้หาระยะห่างของวัตถุในระบบพิกัดของ xyz ในแกนสามมิติดังนี้

$$X_{position} = \left(\frac{\left(x - \frac{width}{2} \right) * d_{pixel}}{f} \right) * Z_{position}$$

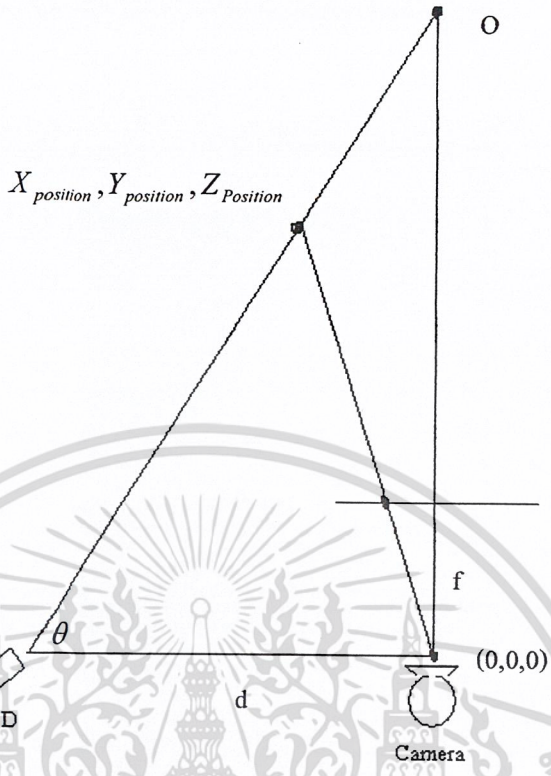
$$Y_{position} = \left(\frac{height}{2} - y \right) * d_{pixel}$$

$$Z_{position} = \left(d / \left(\frac{\left(x - \frac{width}{2} \right) * d_{pixel}}{f} \right) \right) + \cot(\theta)$$

โดย

$width$	คือ ความกว้างของความละเอียดของรูปที่กล้องได้บันทึก
$height$	คือ ความสูงของความละเอียดของรูปที่กล้องได้บันทึก
x	คือ ตำแหน่งของพิกเซลที่ต้องการหาข้อมูลสามมิติของภาพสองมิติในแนวราบ
y	คือ ตำแหน่งของพิกเซลที่ต้องการหาข้อมูลสามมิติของภาพสองมิติในแนวดิ่ง
d_{pixel}	คือ ค่าระยะห่างระหว่างพิกเซลของ Image Plane ของกล้อง
f	คือ ระยะโฟกัสของกล้อง
d	คือ ระยะห่างระหว่างกล้องจนถึงแหล่งกำเนิดแสง
θ	คือ มุมที่แนวแสงทำมุมกับระนาบของกล้อง
$X_{position}, Y_{position}, Z_{Position}$	คือ ตำแหน่งจริงของจุดบนวัตถุโดยที่มีจุดกำเนิดอยู่ที่กล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-15 แสดงถึงตำแหน่งต่างๆในระบบ

เนื่องจากข้อมูลทางสามมิติที่ได้ออกมาจะเป็นข้อมูล ณ เฟรมนั้นๆ โดยที่ยังมีจุดกำเนิดอยู่ที่ตำแหน่งของกล้องดังรูปที่ 6-15 ซึ่งจำเป็นต้องนำค่าที่ได้ดังกล่าวนั้นมาทำการเปลี่ยนแปลงให้มีจุดกำเนิดอยู่ที่จุดศูนย์กลางของจุดหมุน (จุด O ในรูปที่ 6-15) ซึ่งวิธีการเปลี่ยนให้จุดกำเนิดนั้นมาอยู่ที่จุด O สามารถทำได้โดยใช้สมการดังนี้

$$Z_O = Z_{camera} - 31.75$$

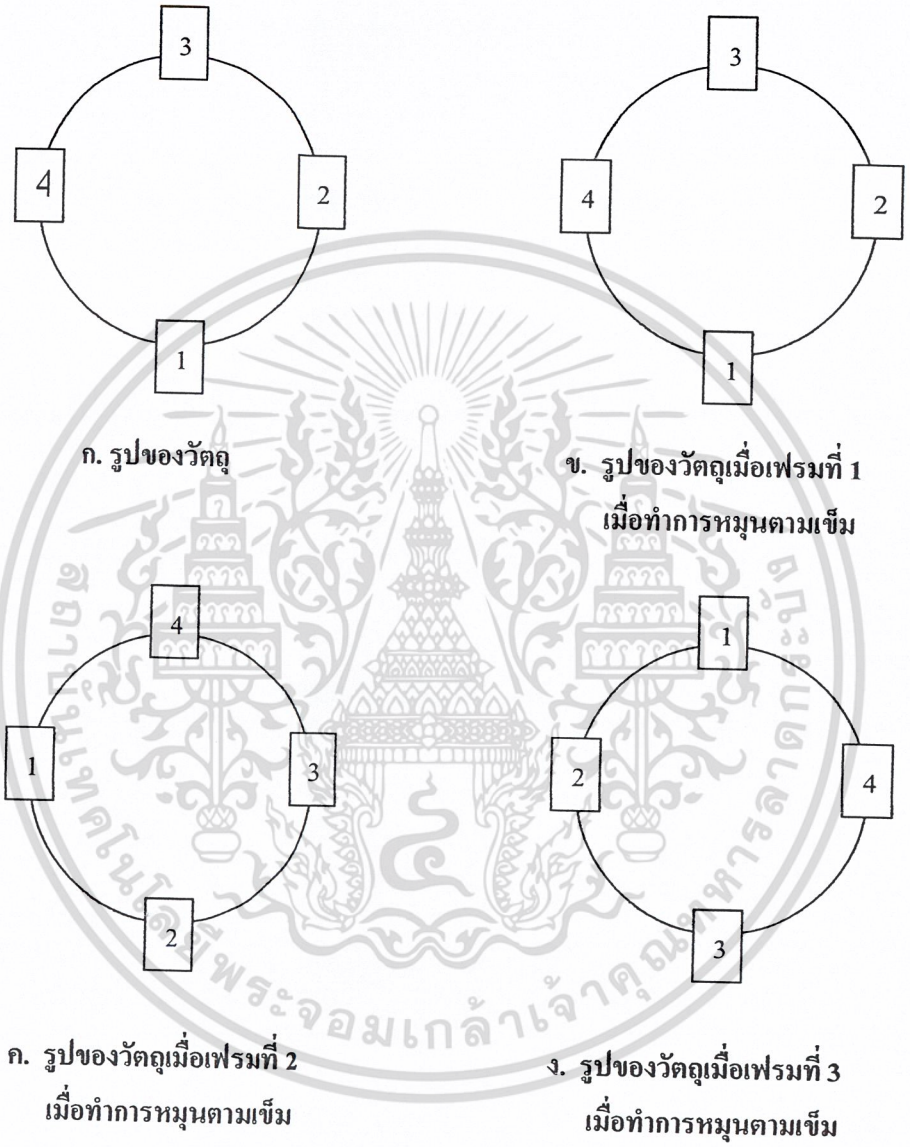
โดย

Z_0	คือ ค่า Z ที่มีจุดกำเนิดอยู่ที่ตำแหน่ง O
Z_{camera}	คือ ค่า Z ที่มีจุดกำเนิดอยู่ที่ตำแหน่งกล้อง
31.75	คือ ระยะห่างของกล้องกับจุด O

ซึ่งในระบบนี้จะทำการเปลี่ยนเฉพาะค่า Z อย่างเดียวเนื่องจาก จุดของกล้อง กับ จุด O นั้นต่างกันเพียงในแนวแกน Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการเปลี่ยนแปลงจุดกำเนิดแล้วก็จะทำให้ได้ข้อมูลทางสามมิติของวัตถุมา ณ เฟรมนั้นๆแต่การทำงานของระบบนั้นจะมีการหมุนวัตถุเพื่อทำการเก็บข้อมูลให้ได้ทั่วทั้งวัตถุทำให้จำเป็นที่จะต้องทำการเปลี่ยนค่าของข้อมูลสามมิติที่ได้มานั้นให้เป็นไปตามการหมุนของวัตถุ ดังรูปที่ 6-16



ก. รูปของวัตถุ

ข. รูปของวัตถุเมื่อเฟรมที่ 1 เมื่อทำการหมุนตามเข็ม

ค. รูปของวัตถุเมื่อเฟรมที่ 2 เมื่อทำการหมุนตามเข็ม

ง. รูปของวัตถุเมื่อเฟรมที่ 3 เมื่อทำการหมุนตามเข็ม

รูปที่ 6-16 รูปการหมุนของวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าเมื่อเฟรมที่ 2 จะได้ตำแหน่งของจุดที่ 2 มาซึ่งยังไม่ใช่ตำแหน่งที่ข้อมูลที่จริงระบบจะต้องทำการหมุนเปลี่ยนแปลงค่าจุดที่ 2 ที่ได้กลับไป -90° ตามระบบในไดเร็กซ์เอ็กซ์ เช่นเดียวกันกับเฟรมที่ 3 ระบบจำเป็นต้องหมุนกลับไปที่ -270°

ซึ่งการหมุนสามารถทำได้โดยใช้สมการดังนี้

$$[x'\ y'\ z'1] = [xyz\ 1]\begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

โดย

$[x'\ y'\ z'1]$ คือ เมตริกซ์ที่ประกอบไปด้วย ตำแหน่งของวัตถุที่ทำการหมุนแล้ว

$[xyz\ 1]$ คือ เมตริกซ์ที่ประกอบไปด้วยตำแหน่งของวัตถุที่ต้องการเปลี่ยน

$\begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & \\ 0 & 0 & 0 & 1 \end{bmatrix}$ คือ เมตริกซ์ที่ใช้ในการหมุนตำแหน่งรอบแกน y โดย θ คือ มุมที่ใช้หมุน

หลังจากทำการเปลี่ยนแปลงค่าทั่ได้ทั้งหมดแล้วจึงนำข้อมูลที่ได้ทั้งหมดส่งไปยังส่วนที่ทำการเรนเดอร์โมเดลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.7 การออกแบบระบบแสดงผล (ไดเร็กซ์เอ็กซ์)

6.7.1 เรนเดอร์ (Render)

6.7.1.1 การตั้งค่าสิ่งแวดล้อม(environment)

ก่อนที่เราจะทำการเรนเดอร์นั้น จำเป็นที่จะต้องตั้งค่าต่างๆเพื่อให้เหมาะสมกับโมเดลที่ต้องการเรนเดอร์ ซึ่งค่าที่จะตั้งค่านั้นจะขึ้นอยู่กับความต้องการว่าจะให้เรนเดอร์ออกมาอย่างไร และจะใช้ตัวเลือกไหน แต่โดยหลักที่โปรแกรมนี้ได้ทำการตั้งค่าจะมีดังนี้

1) การตั้งค่าเวิร์ลเมตริกซ์ (world matrix) โดยใช้ฟังก์ชันดังนี้

```
SetTransform( state,  &matrix );
```

พารามิเตอร์

state คือ ให้ใช้ค่า D3DTS_WORLD เพื่อระบุว่าเป็นการตั้งค่าเวิร์ลเมตริซ์

matrix คือ ค่าเมตริกซ์ที่ใช้ในการตั้งค่าให้ใช้ค่า เมตริกซ์เอกลักษณ์ หรือ ใช้ค่า 0 ก็จะได้เป็นเมตริกซ์เอกลักษณ์เหมือนกัน

2) การตั้งค่าวิวเมตริกซ์(view matrix) โดยการจะตั้งค่าทีละเวกเตอร์แล้วจึงนำมารวมกันกลายเป็น วิวเมตริกซ์ โดยต้องกำหนดเวกเตอร์เหล่านี้ คือ

- กำหนดจุดที่อยู่ของกล้อง
- กำหนดจุดที่กล้องมองไป
- กำหนดทิศทางของกล้อง

โดย ใช้ฟังก์ชันดังนี้

```
D3DXVECTOR3 vFromPt   = D3DXVECTOR3( 0.0f, 0.0f, -5.0f );
```

กำหนดว่าจุดที่กล้องอยู่คือ 0,0,-5

```
D3DXVECTOR3 vLookatPt = D3DXVECTOR3( 0.0f, 0.0f, 0.0f );
```

กำหนดว่าให้กล้องมองไปที่จุด 0,0,0

```
D3DXVECTOR3 vUpVec    = D3DXVECTOR3( 0.0f, 1.0f, 0.0f );
```

กำหนดว่ากล้องมีทิศทางใน 0,1,0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วทำการรวมเวกเตอร์เป็นวิวเมตริกซ์โดยใช้ฟังก์ชันดังนี้

D3DXMatrixLookAtLH(&matrix, &vFromPt, &vLookatPt, &vUpVec);

เป็นการรวมเมตริกซ์โดยใช้ในพิกัด x,y,z แบบมือซ้าย

พารามิเตอร์

matrix คือ เมตริกซ์ที่มีมารับค่าจากการรวมเวกเตอร์

vFrompt,vLookatPt,vUpVec คือเวกเตอร์ที่ต้องการรวมเป็นเมตริกซ์ตามลำดับดังนี้ จุดที่กล้องอยู่,จุดที่กล้องมองไป และ ทิศทางของกล้อง

หลังจากนั้นจึงนำเมตริกซ์ที่ได้ไปตั้งค่าเป็นวิวเมตริกซ์ดังนี้

SetTransform(state, &matrix);

พารามิเตอร์

state คือ ให้ใช้ค่า D3DTS_VIEW เพื่อระบุว่าเป็นการตั้งค่าวิวเมตริซ์

matrix คือ ค่าเมตริกซ์ที่ใช้ในการตั้งค่าให้ใช้ค่าที่ได้มาจากการรวมเวกเตอร์

3) ทำการตั้งค่าโปรเจ็กชันเมตริกซ์(projection matrix) โดยทำการสร้างโปรเจ็กชันเมตริกซ์ที่ใช้ระบบมือซ้ายโดยใช้ฟังก์ชันดังนี้

D3DXMatrixPerspectiveFovLH(&matrix, fovy, fAspect, zn, zf);

พารามิเตอร์

matrix คือ เมตริกซ์ที่มารับค่าที่ได้จากฟังก์ชัน เพื่อเป็นเมตริกซ์โปรเจ็กชัน

fovy คือ ค่าที่ใช้ในการกำหนดฟิลด์ออฟวิว(field of view) โดยจะกำหนดค่าเป็น D3DX_PI/4 คือค่า pi หารด้วย 4 = 45 องศา

fAspect คือ ค่าที่เกิดจากความกว้างหารด้วยความสูง

zn คือ ค่าที่บอกว่าวิวเพลนที่อยู่ใกล้มีระยะห่างเท่าไรหน่วยในแกน z ให้กำหนดค่าเป็น 1.0

zf คือ ค่าที่บอกว่าวิวเพลนที่อยู่ไกลมีระยะห่างเท่าไรหน่วยในแกน z ให้กำหนดค่าเป็น 100.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วจึงนำเมตริกซ์ที่ได้มากำหนดเป็นโปรเจ็กชันเมตริกซ์โดยใช้ฟังก์ชันดังนี้

```
SetTransform( state, &matrix );
```

พารามิเตอร์

state	คือ ให้ใช้ค่า D3DTS_PROJECTION เพื่อระบุว่าเป็นการตั้งค่าโปรเจ็กชันเมตริซ์
matrix	คือ ค่าเมตริกซ์ที่ต้องการจะให้ใช้ในการตั้งค่าโปรเจ็กชันเมตริกซ์ ให้ใช้ค่าที่ได้มาจากการใช้ฟังก์ชันการกำหนดเมตริกซ์โปรเจ็กชัน

4) ทำการตั้งค่าเกี่ยวกับแสงโดยใช้ฟังก์ชันดังนี้

```
D3DUtil_InitLight( light, D3DLIGHT_DIRECTIONAL, -1.0f, -1.0f, 2.0f );
```

พารามิเตอร์

light	คือ ค่าดีทรีดีไลท์ไนน์(D3DLIGHT9)ซึ่งในไดเร็กซ์ทรีดีได้ทำการกำหนดไว้แล้วซึ่งเอาไว้เก็บค่าของแสงที่ต้องการ
state	คือ ค่าที่บอกถึงลักษณะแสงที่จะใช้ ให้กำหนดเป็นแบบไดเร็กชันนอล โดยให้ใช้ค่า D3DLIGHT_DIRECTIONAL
x,y,z	คือ ค่าที่ระบุทิศทางของแสง โดยกำหนดเป็น -1.0f, -1.0f, 2.0f

หลังจากนั้นเราจะได้ค่ามาอยู่ที่ light จึงนำไปกำหนดในค่าของดีไวซ์โดยใช้ฟังก์ชันดังนี้

```
SetLight(n, &light );
```

พารามิเตอร์

n	คือ ค่าอินเด็กซ์ของดีไวซ์ซึ่งกำหนดเป็น 0 เนื่องจากมีเพียงดีไวซ์เดียว
light	คือ ตัวแปรแบบดีทรีดีไลท์ที่จะนำไปกำหนดค่าของดีไวซ์ ซึ่งใช้ค่าที่ได้มาจากการใช้ฟังก์ชันกำหนดค่าแบบดีทรีดีไลท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นทำการกำหนดให้มีการใช้แสงของดีไวซ์โดยใช้ฟังก์ชันดังนี้

```
LightEnable(n, enablelight );
```

พารามิเตอร์

n	คือ ค่าอินเด็กซ์ของแสงซึ่งกำหนดเป็น 0 เนื่องจากมีเพียงแสงจุดเดียว
enablelight	คือ การเปิดหรือปิดไม่ให้ใช้ค่าแสงในอินเด็กซ์ กำหนดเป็น TRUE

หลังจากนั้นก็ทำการตั้งค่าให้ใช้แสงในการเรนเดอร์โดยใช้ฟังก์ชัน

```
SetRenderState( D3DRS_LIGHTING, enablelight );
```

พารามิเตอร์

state	คือ ค่าที่ระบุถึงตัวเลือกในการเรนเดอร์ ซึ่งให้ใช้ค่า D3DRS_LIGHTING เพื่อเป็นการบอกว่าให้ใช้แสงหรือไม่
enablelight	คือ การเปิดหรือปิดว่าให้มีการใช้แสงหรือไม่ กำหนดค่าเป็น TRUE

เมื่อตั้งค่าเหล่านี้เสร็จแล้วก็สามารถนำเมชที่มีข้อมูลมาทำการเรนเดอร์ได้ ซึ่งต่อไปเราจะทำการสร้างเมชขึ้นมาเองโดยอาศัยข้อมูลโมเดลที่ได้ในเมชมาเรนเดอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.7.1.2 การตั้งค่าเมช

นำข้อมูลในอาร์เรย์ซึ่งเก็บเป็นระบบพิกัด xyz มาทำการหาจำนวนของเวอร์เท็กซ์ทั้งหมดของโมเดลวัตถุ ,จำนวนหน้าของโมเดล เพื่อนำไปทำการสร้างเมชใหม่ขึ้นมา โดยใช้ฟังก์ชัน ดังนี้

```
D3DXCreateMeshFVF
(
        NumFaces,
        NumVertices,
        Options,
        FVF,
        Device,
        &Mesh
);
```

พารามิเตอร์

NumFaces	คือ จำนวนหน้าที่มีในเมช
NumVertices	คือ จำนวนเวอร์ไทซ์ที่มีในเมช
Options	คือ ตัวเลือกในการสร้างเมช ให้กำหนดค่าเป็น D3DXMESH_MANAGED เป็นการสร้างเมชแบบมี เวอร์เท็กซ์บัฟเฟอร์ และ อินเด็กซ์บัฟเฟอร์
FVF	คือ การกำหนดรูปแบบของเวอร์เท็กซ์ในการสร้างเมช ซึ่งในที่นี้ให้ใช้เป็น Vertex::FVF คือ เวอร์เท็กซ์ที่มีข้อมูลของ พิกัด xyz, นอร์มอลเวกเตอร์ และ ค่าของสี
Device	คือ พอยต์เตอร์ที่ชี้ไปยัง IDirect3DDevice9 ซึ่งเป็น ดีไวซ์ที่ใช้กับเมช
Mesh	คือ ที่อยู่ของพอยต์เตอร์ที่ชี้ไปยัง ID3DXMesh คือ ตัวเมชนั้นเอง

เมื่อสร้างเมชขึ้นมาได้แล้วจึงนำค่าของเวอร์เท็กซ์ทั้งหมดไปใส่ไว้ใน เวอร์เท็กซ์บัฟเฟอร์ ของเมชที่จะทำการเรนเดอร์ โดยใช้ฟังก์ชันดังนี้

```
LockVertexBuffer(flag0, (void**)&v);
```

พารามิเตอร์

flag	คือ การระบุว่าจะล็อคแบบไหน ให้กำหนดเป็น 0 หมายถึงไม่ระบุ
ppdata	คือ ค่าที่นำมารับพอยต์เตอร์ที่ชี้ไปยังข้อมูลเวอร์เท็กซ์ ให้ใช้เป็น (void**)&v ทำให้ v ชี้ไปยังเวอร์เท็กซ์บัฟเฟอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นจะทำการใส่ค่าลงใน v ซึ่งกำหนดค่าดังนี้

```
v[n] = Vertex(x, y, z, nx, ny, nz, D3DCOLOR_XRGB(r,g,b));
```

*รูปแบบอาจจะมีการเปลี่ยนแปลง แล้วแต่การประกาศ เนื่องจากในโปรแกรมนี้ได้กำหนดให้มีการเก็บ พิกัด xyz นอร์มอลเวกเตอร์ xyz และค่าสีของเวอร์เท็กซ์แบบ RGB

พารามิเตอร์

n	คือ ค่าอินเด็กซ์ของเวอร์เท็กซ์ที่บอกว่าเป็นเวอร์เท็กซ์ที่เท่าไร
x,y,z	คือ ค่าพิกัดในระบบ xyz
nx,ny,nz	คือ ค่านอร์มอลเวกเตอร์ในระบบพิกัด xyz
r,g,b	คือ ค่าสีของเวอร์เท็กซ์

แล้วจากนั้นจึงทำการปลดล็อคเวอร์เท็กซ์บัฟเฟอร์โดยใช้ฟังก์ชันดังนี้

```
UnlockVertexBuffer();
```

หลังจากนั้นทำการระบุอินเด็กซ์ของอินเด็กซ์เพื่อทำการเรนเดอร์ผิวหน้า โดยใช้ฟังก์ชันดังนี้

```
LockVertexBuffer(0, (void**)&i);
```

พารามิเตอร์

flag	คือ การระบุว่าจะล็อคแบบไหน ให้กำหนดเป็น 0 หมายถึงไม่ระบุ
ppdata	คือ ค่าที่นำมารับพอยต์เตอร์ที่ชี้ไปยังข้อมูลอินเด็กซ์ ให้ใช้เป็น (void**)&i ทำให้ i ชี้ไปยังอินเด็กซ์บัฟเฟอร์

หลังจากนั้นจึงทำการกำหนดค่าลงใน i ดังนี้

```
i[n] = iv
```

พารามิเตอร์

n	คือ คือค่าอินเด็กซ์ที่บอกว่าอินเด็กซ์ที่เท่าไรของอาร์เรย์
iv	คือ ลำดับจุดของเวอร์เท็กซ์ที่ใช้ เรนเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดค่าอินเด็กซ์นั้นจะต้องกำหนดทีละหน้านั้นหมายถึงต้องกำหนดสามจุดเพื่อให้ได้สามเหลี่ยมหนึ่งหน้า หลังจากกำหนดค่าอินเด็กซ์เสร็จแล้วจึงทำการปลดล็อคอินเด็กซ์โดยใช้ฟังก์ชันดังนี้

UnlockIndexBuffer();

หลังจากกำหนดค่าเวอร์เท็กซ์และอินเด็กซ์เสร็จแล้ว หากยังไม่ได้กำหนดค่าของนอร์มอลเวกเตอร์ หรือไม่รู้ค่า ก็สามารถใช้ฟังก์ชันเพื่อให้คำนวนนอร์มอลเวกเตอร์ได้ ด้วยฟังก์ชันนี้

D3DXComputeNormals(Mesh,pAdjacency);

พารามิเตอร์

Mesh	คือ พอยต์เตอร์ที่ชี้ไปยังเมช
pAdjacency	คือ พอยต์เตอร์ที่ชี้ไปอาร์เรย์ของซึ่งระบุถึงหน้าที่ติดกับด้านทั้งสาม ซึ่งเอาไว้ใช้ในการสร้าง โปรเกรสซิพเมช ซึ่งในโปรแกรมนี้เราไม่ใช้ จึงกำหนดเป็น 0 หรือ NULL

เมื่อเสร็จขั้นตอนนี้แล้วเราก็จะมีเมชที่มีข้อมูลเวอร์เท็กซ์บัฟเฟอร์,อินเด็กซ์บัฟเฟอร์ ซึ่งเพียงพอกับการที่จะนำไปเรนเดอร์โดยใช้ฟังก์ชัน

DrawSubset(n)

พารามิเตอร์

n	คือ เลขของซับเซ็ตที่ต้องการให้แสดง ซึ่งในโปรแกรมนี้ไม่ได้มีการแบ่งซับเซ็ตเอาไว้จึงใช้ 0 ซึ่งหมายถึงวาดอันแรกซึ่งเป็นอันเดียว

หลังจากนั้นเวลาปิดโปรแกรมหากไม่มีการปลดการจองเนื้อที่ของอาร์เรย์ก็จะทำให้เกิดความผิดพลาดขึ้นได้จึงจำเป็นต้องทำการปลดการจองเนื้อที่โดยใช้ฟังก์ชันดังนี้

SAFE_DELETE(Mesh)

พารามิเตอร์

Mesh	คือ พอยต์เตอร์ที่ชี้ไปยังเมชที่ต้องการปลดล็อคการจองเนื้อที่

แล้วใช้ข้อมูลสามมิติของวัตถุนำมาบรรจุในเมชก็จะสามารถนำไปเรนเดอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.7.2 โหลดเอ็กซ์ไฟล์ (Load .x File)

ไดเร็กเอ็กซ์ได้สร้างฟังก์ชันขึ้นมาใช้ในการติดต่อกับเอ็กซ์ไฟล์ซึ่งมีหลายฟังก์ชัน หนึ่งในฟังก์ชันหลักก็คือ การโหลด เอ็กซ์ไฟล์ โดยในโปรแกรมจะใช้รูปแบบดังนี้

```
hr = D3DXLoadMeshFromX
(
        Filename,
        Options,
        Device,
        pAdjacency,
        pMaterial,
        pEffect,
        pNumMaterial,
        Mesh
);
```

พารามิเตอร์

Filename	คือ พอยต์เตอร์ไปยังสตริงที่ระบุไปยังไฟล์
Options	คือ ตัวเลือกในการสร้างเมช ให้กำหนดค่าเป็น D3DXMESH_MANAGED เป็นการสร้างเมชแบบมี เวอร์เท็กซ์บัฟเฟอร์ และ อินเด็กซ์บัฟเฟอร์
Device	คือ พอยต์เตอร์ของ IDirect3DDevice9 ที่เป็นดีไวซ์ที่ใช้กับเมช
pAdjacency	คือ พอยต์เตอร์ไปยังบัฟเฟอร์ซึ่งเก็บ แอดจาเซนซี่ดาต้า(adjacency data) ซึ่งระบุถึงหน้าที่ติดกับด้านทั้งสาม ซึ่งในโปรแกรมนี้เราจะไม่ใช้ ให้กำหนดเป็น 0 หรือ NULL
pMaterial	คือ พอยต์เตอร์ไปยังบัฟเฟอร์ที่เก็บ แมททีเรียลดาต้า (materials data) ให้กำหนดค่าเป็น 0 หรือ NULL เนื่องจากโปรแกรมนี้จะไม่ใช้ เนื่องจากไม่มีแมททีเรียลอย่างอื่น ใช้แค่ เวอร์เท็กซ์บัฟเฟอร์ และ อินเด็กซ์บัฟเฟอร์ก็เพียงพอแล้ว
pEffect	คือ พอยต์เตอร์ไปยังบัฟเฟอร์ที่เก็บอาร์เรย์ของ เอฟเฟกต์อินสแตนซ์ (effect instance) ซึ่งกำหนดเป็น 0 หรือ NULL เนื่องจากโปรแกรมนี้ไม่ใช้เนื่องจากไม่จำเป็น
pNumMaterial	คือ พอยต์เตอร์ไปยังจำนวนของแมททีเรียลใน pMaterial แต่เนื่องจากเราไม่มีแมททีเรียลเราจึงกำหนดค่าเป็น NULL
Mesh	คือ ที่อยู่ของพอยต์เตอร์ที่ชี้ไปยังเมชที่ได้โหลดมา
hr	คือ ตัวแปลแบบ HRESULT ซึ่งจะคืนค่า 0 เมื่อเกิดการผิดพลาดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.7.3เซฟเอ็กซ์ไฟล์ (Save .x File)

```
hr = D3DXSaveMeshToX
(
        Filename,
        Mesh,
        pAdjacency,
        pMaterial,
        pEffect,
        pNumMaterial,
        Format
);
```

พารามิเตอร์

Filename	คือ พอยต์เตอร์ไปยังสตริงที่ระบุไปยังไฟล์
Mesh	คือ ที่อยู่ของพอยต์เตอร์ที่ชี้ไปยังเมชที่จะนำไปบันทึก
pAdjacency	คือ พอยต์เตอร์ไปยังบัฟเฟอร์ซึ่งเก็บ แอดจาเซนซี่ดาต้า(adjacency data) ซึ่งระบุถึงหน้าที่ติดกับด้านทั้งสาม ซึ่งในโปรแกรมนี้เราจะไม่ใช้ ให้กำหนดเป็น 0 หรือ NULL
pMaterial	คือ พอยต์เตอร์ไปยังบัฟเฟอร์ที่เก็บ แมททีเรียลดาต้า (materials data) ให้กำหนดค่าเป็น 0 หรือ NULL เนื่องจากโปรแกรมนี้จะไม่ใช้ เนื่องจากไม่มีแมททีเรียลอย่างอื่น ใช้แค่ เวอร์เท็กซ์บัฟเฟอร์ และ อินเด็กซ์บัฟเฟอร์ก็เพียงพอแล้ว
pEffect	คือ พอยต์เตอร์ไปยังบัฟเฟอร์ที่เก็บอาร์เรย์ของ เอฟเฟกต์อินสแตนซ์ (effect instance) ซึ่งกำหนดเป็น 0 หรือ NULL เนื่องจากโปรแกรมนี้ไม่ใช้เนื่องจากไม่จำเป็น
pNumMaterial	คือ พอยต์เตอร์ไปยังจำนวนของแมททีเรียลใน pMaterial แต่เนื่องจากเราไม่มีแมททีเรียลเราจึงกำหนดค่าเป็น NULL
Format	คือ รูปแบบของไฟล์ที่จะบันทึกลงในเอ็กซ์ไฟล์ ซึ่งสามารถที่จะเลือกได้ว่าให้บันทึกแบบ ไบนารี่ หรือ เท็กซ์ ให้กำหนดค่าเป็น DXFILEFORMAT_TEXT คือการบันทึกเป็นเท็กซ์ เพื่อให้ง่ายแก่การอ่านรายละเอียดในไฟล์โดยตรง
hr	คือ ตัวแปลแบบ HRESULT ซึ่งจะคืนค่า 0 เมื่อเกิดการผิดพลาดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
การทดลอง

7.1 การทดลอง เพื่อกำหนดค่าเทรชโฮลด์

ระบบสร้างภาพคอมพิวเตอร์แบบสามมิติโดยอาศัยข้อมูลระยะห่างของกล้องกับวัตถุ นั้นคือ ระบบที่นำวัตถุที่ต้องการจะสร้างโมเดลมาผ่านกระบวนการของระบบ โดยที่กระบวนการของระบบ คือ การฉายแถบแสงลงบนวัตถุ แล้วจึงบันทึกภาพโดยกล้องแล้วจึงนำไปประมวลผล ซึ่งการประมวลผลภาพนั้น คือ การนำเอาแถบแสงที่ฉายลงบนวัตถุออกมาเป็นข้อมูล ซึ่งแสงที่ฉายลงบนวัตถุนั้นจะแตกต่างไปในแต่ละวัตถุ ซึ่งเพื่อประสิทธิภาพในการประมวลผลภาพเราจำเป็นที่จะต้องปรับค่าเทรชโฮลด์ก็คือ ค่าความเข้มข้องแสง(intensity of line) ในแต่ละวัตถุ ซึ่งการทดลองค่าเทรชโฮลด์นั้นเพื่อหาค่าที่ น้อยที่สุด และมากที่สุดในการปรับค่า เพื่อให้เกิดประสิทธิภาพมากที่สุดนั้นเอง โดยที่ไม่มีสิ่งรบกวนต่างๆปนมาด้วย

ขั้นตอนการทดลอง

การทดลองนี้จะใช้วัตถุที่มีลักษณะเป็นทรงกระบอกดังรูปที่ 7-1 ก. และแสงสามารถส่องเห็นได้อย่างชัดเจน มีความละเอียดที่ 160x120 พิกเซลแล้วทำการเปลี่ยนค่าเทรชโฮลด์ให้เป็นค่าต่างๆ ซึ่งจากการทดลองเบื้องต้น ค่าเทรชโฮลด์ที่น้อยนั้นจะอยู่ช่วง 30-50 และ ค่าเทรชโฮลด์ที่มากนั้นจะอยู่ที่ช่วง 110-130 จึงทำการทดลองต่อว่าค่าเทรชโฮลด์ใดเป็นค่าที่น้อยที่สุดที่ควรปรับ และค่าใดเป็นค่ามากที่สุดที่ควรปรับ

ผลการทดลอง

จากรูปที่ 7-1 จะเห็นได้ว่า

รูป ค. มีค่าเทรชโฮลด์30 นั้นจะเห็นได้ว่ามีข้อมูลของแถบแสงมีขนาดกว้างมากและมีข้อมูลที่ผิดพลาดอยู่เยอะ(ในวงกลม) ซึ่งมีทั้งข้างบนของรูปด้วย

รูป จ. มีค่าเทรชโฮลด์40 นั้นจะเห็นได้ว่ามีข้อมูลของแถบแสงมีขนาดกว้างแต่น้อยลงกว่า รูป ค. และมีข้อมูลที่ผิดพลาดอยู่ (ในวงกลม) แต่น้อยกว่ารูป ค. และไม่มีข้อมูลที่ผิดพลาดข้างบน

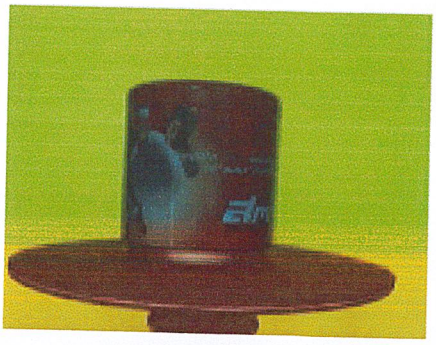
รูป ช. มีค่าเทรชโฮลด์50 นั้นจะเห็นได้ว่ามีข้อมูลของแถบแสงมีขนาดกว้างแต่น้อยลงกว่า รูป จ. และมีข้อมูลที่ผิดพลาดอยู่ (ในวงกลม) แต่น้อยกว่ารูป จ. และไม่มีข้อมูลที่ผิดพลาด

รูป ง. มีค่าเทรชโฮลด์110 นั้นจะเห็นได้ว่ามีข้อมูลของแถบแสงมีขนาดแคบและไม่เต็มเส้นแต่เส้นในแนวดิ่งนั้นยังไม่ขาดช่วง

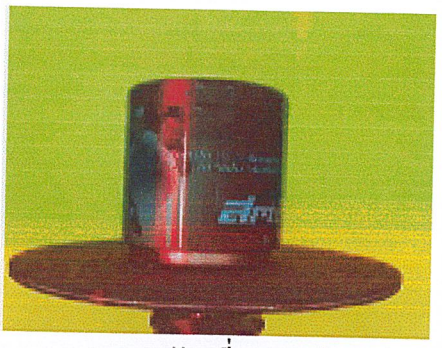
รูป ฉ. มีค่าเทรชโฮลด์120 นั้นจะเห็นได้ว่ามีข้อมูลของแถบแสงมีขนาดแคบและไม่เต็มเส้น มากกว่า รูป ง. แต่เส้นในแนวดิ่งนั้นยังไม่ขาดช่วงและมีข้อมูลที่หายไปมากขึ้น

รูป ซ. มีค่าเทรชโฮลด์130 นั้นจะเห็นได้ว่ามีข้อมูลของแถบแสงมีขนาดแคบและไม่เต็มเส้น อีกทั้งเส้นในแนวดิ่งนั้นขาดช่วงไป (ในวงกลม)

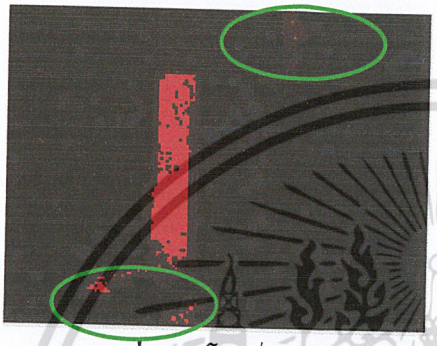
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก. ภาพของวัตถุ



ข. ภาพของวัตถุที่ถูกฉายส่อง

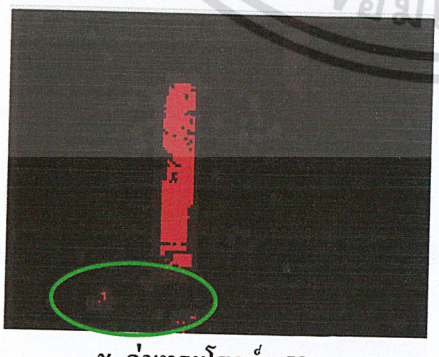


ค. ค่าเทรชโฮลด์= 30

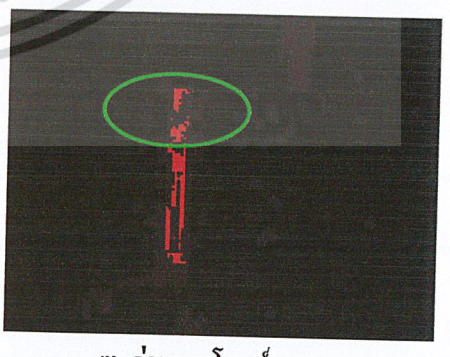
ง. ค่าเทรชโฮลด์= 110

จ. ค่าเทรชโฮลด์= 40

ฉ. ค่าเทรชโฮลด์= 120



ช. ค่าเทรชโฮลด์= 50



ซ. ค่าเทรชโฮลด์= 130

รูปที่ 7-1 รูปผลการทดลองค่าเทรชโฮลด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปการทดลอง

จากการทดลองเมื่อปรับค่าเทรชโฮลด์ให้มีค่าน้อยมากๆ จะทำให้เห็นแถบแสงที่มีความเข้มน้อยได้ แต่ทำให้เกิดเส้นแสงที่มีขนาดกว้างมาก และทำให้เกิดข้อมูลที่มีความผิดพลาดมาก แต่ถ้าทำการปรับค่าเทรชโฮลด์มากๆ ก็จะทำให้เห็นเฉพาะแถบแสงที่มีความเข้มมาก ซึ่งหากความเข้มของแถบแสงน้อยก็อาจจะทำให้เกิดช่วงข้อมูลที่ขาดหายไป

ถึงแม้การปรับค่าเทรชโฮลด์จะทำให้สามารถเห็นแถบแสงที่มีความเข้มน้อยและกำจัดสิ่งรบกวนออกไปได้ แต่ก็ไม่ควรปรับให้มากหรือน้อยไปกว่า 40-120 เพราะจะทำให้เกิดข้อมูลที่ผิดพลาดหรือขาดหายไปได้

7.2 การทดลอง เพื่อหาเวลาที่ใช้ในการสแกนวัตถุ

การทำงานของระบบสร้างภาพคอมพิวเตอร์แบบสามมิติโดยอาศัยข้อมูลระยะห่างกล้องกับวัตถุแต่ละครั้งในการทำงานเมื่อมีการเปลี่ยนแปลงค่าของระบบให้แตกต่างจากเดิม ก็จะทำให้ระยะเวลาที่ใช้ในการสแกนวัตถุแตกต่างกันไปด้วย ซึ่งการทดลองนี้จะทำการทดลองเพื่อหาปัจจัยที่มีต่อระยะเวลาที่ใช้ในการสแกนวัตถุ

ขั้นตอนการทดลอง

จะทำการทดลองโดยการเปลี่ยนแปลงค่าของระบบ: โดยจะเริ่มจับเวลาตั้งแต่ใส่วัตถุและกดตกลงจนเมื่อแสดงผลภาพโมเดลของวัตถุ ทำการเปลี่ยนค่าของระบบดังนี้

1) โหมดการทำงาน ปกติ กับ ดีบัก โดยใช้ความละเอียด 160x120 พิกเซล จำนวนสเตป 50 สเตป ค่าเทรชโฮลด์80

2) ค่าเทรชโฮลด์40,80,120 โดยใช้ความละเอียด 160x120 พิกเซลจำนวนสเตป 50 สเตป และโหมดการทำงานทั้งปกติ และ ดีบัก

3) ความละเอียด 160x120,320x240,640x480 พิกเซลจำนวนสเตป 50 สเตป ค่าเทรชโฮลด์80และโหมดการทำงานทั้งปกติ และ ดีบัก

4) จำนวนสเตป 25,50,100,200 ความละเอียด 160x120 พิกเซลค่าเทรชโฮลด์80 และโหมดการทำงานทั้งปกติ และ ดีบัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

1) เมื่อมีการเปลี่ยนแปลงโหมดการทำงาน

ความละเอียด(พิกเซล)	จำนวนสเตป	เทรชโฮลด์	โหมดการทำงาน	เวลา(วินาที)
160x120	50	80	ปกติ	139
160x120	50	80	ดีบัก	170

ตารางที่ 7-1 ผลการทดลองเมื่อเปลี่ยนแปลงโหมดการทำงาน

จะเห็นว่าการเปลี่ยนแปลงโหมดการทำงาน จะมีผลเป็นทำให้ระยะเวลาในการทำงานเพิ่มขึ้นเป็นอย่างมาก คือการทำงานในโหมดดีบักนั้น จะมีระยะเวลาที่ใช้ในการสแกนวัตถุที่มากกว่าการทำงานในโหมดปกติ

2) เมื่อมีการเปลี่ยนแปลงค่าของเทรชโฮลด์

ความละเอียด(พิกเซล)	จำนวนสเตป	เทรชโฮลด์	โหมดการทำงาน	เวลา(วินาที)
160x120	50	40	ปกติ	135
160x120	50	80	ปกติ	138
160x120	50	120	ปกติ	136
160x120	50	40	ดีบัก	172
160x120	50	80	ดีบัก	170
160x120	50	120	ดีบัก	173

ตารางที่ 7-2 ผลการทดลองเมื่อเปลี่ยนแปลงค่าเทรชโฮลด์

จะเห็นว่าการปรับเปลี่ยนค่าเทรชโฮลด์นั้นไม่ว่าจะเปลี่ยนแปลงมากขึ้น หรือ น้อยลง ก็ไม่มีผลต่อระยะเวลาที่ใช้ในการสแกนวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) เมื่อทำการเปลี่ยนแปลงค่าความละเอียด

ความละเอียด(พิกเซล)	จำนวนสเตป	เทรชโฮลด์	โหมดการทำงาน	เวลา(วินาที)
160x120	50	80	ปกติ	137
320x240	50	80	ปกติ	139
640x480	50	80	ปกติ	142
160x120	50	80	ดีบัก	168
320x240	50	80	ดีบัก	173
640x480	50	80	ดีบัก	176

ตารางที่ 7-3 ผลการทดลองเมื่อเปลี่ยนแปลงความละเอียด

จะเห็นว่าการเปลี่ยนแปลงค่าความละเอียดนั้น จะมีผลเพียงเล็กน้อยต่อระยะเวลาที่ใช้ในการสแกนวัตถุ คือเมื่อมีความละเอียดมากขึ้นทำให้ระยะเวลาที่ใช้ในการสแกนวัตถุเพิ่มขึ้นเล็กน้อย

4) เมื่อมีการเปลี่ยนแปลงจำนวนสเตป

ความละเอียด(พิกเซล)	จำนวนสเตป	เทรชโฮลด์	โหมดการทำงาน	เวลา(วินาที)
160x120	200	80	ปกติ	256
160x120	100	80	ปกติ	181
160x120	50	80	ปกติ	137
160x120	25	80	ปกติ	118
160x120	200	80	ดีบัก	388
160x120	100	80	ดีบัก	242
160x120	50	80	ดีบัก	169
160x120	25	80	ดีบัก	126

ตารางที่ 7-4 ผลการทดลองเมื่อเปลี่ยนแปลงจำนวนสเตป

จะเห็นว่าการเปลี่ยนแปลงจำนวนสเตป จะมีผลเป็นทำให้ระยะเวลาที่ใช้ในการสแกนวัตถุเพิ่มขึ้นเป็นอย่างมาก คือเมื่อจำนวนสเตปมากขึ้นนั้นจะทำให้ระยะเวลาในการทำงานเพิ่มขึ้นมาก

สรุปการทดลอง

จากการทดลองที่ผ่านมา จะเห็นได้ว่าปัจจัยที่มีผลต่อระยะเวลาที่ใช้ในการสแกนเป็นส่วนใหญ่ มีอยู่สองปัจจัย คือ จำนวนสเตปในการทำงาน และโหมดการทำงานของระบบ ซึ่งเมื่อมีค่ามากขึ้นจะทำให้ระยะเวลาที่ใช้ในการสแกนวัตถุมากขึ้น และมีผลเล็กน้อยก็คือค่าความละเอียด ซึ่งเมื่อปรับมากขึ้นก็จะส่งผลให้ระยะเวลาในการสแกนวัตถุมากขึ้นเล็กน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 การทดลอง สแกนวัตถุที่มีลักษณะพื้นผิวแบบต่างๆ

ระบบสร้างภาพคอมพิวเตอร์แบบสามมิติโดยอาศัยกล้องวีดีโอเก็บภาพของวัตถุเป็นระบบที่ใช้สำหรับนำวัตถุมาผ่านกระบวนการเพื่อสร้างภาพสามมิติ ดังนั้นตัวแปรที่มีผลต่อระบบอย่างมากคือตัววัตถุเอง เนื่องจากวัตถุที่แตกต่างกันปัจจัยที่สำคัญที่ใช้ในการประมวลผลเพื่อสร้างภาพสามมิติก็จะแตกต่างกันไป ตามลักษณะของวัตถุ เช่น สีของวัตถุ ลักษณะพื้นผิว รูปทรงของวัตถุ ขนาดของวัตถุ

การทดลองนี้ทำเพื่อให้ทราบว่าระบบมีความสามารถที่จะทำงานได้กับวัตถุประเภทใดและได้ผลออกมาเป็นอย่างไรบ้าง

ขั้นตอนการทดลอง

นำวัตถุหลายๆแบบมาผ่านกระบวนการของระบบให้ได้ออกมาเป็นโมเดลสามมิติ โดยใช้วัตถุดังนี้

1) แก้วใสทรงกระบอก ขนาด เส้นผ่าศูนย์กลาง 7.5 เซนติเมตร สูง 8 เซนติเมตร

2) แก้วมีหูจับสีน้ำเงิน ขนาด เส้นผ่าศูนย์กลางด้านบน 9 เซนติเมตร ด้านล่าง 5 เซนติเมตร สูง 8.5 เซนติเมตร

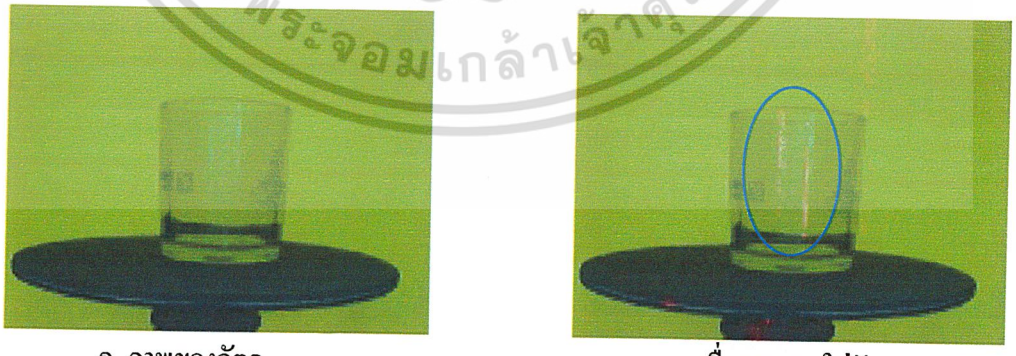
3) กล่องอะลูมิเนียมสีแดง ขนาด เส้นผ่าศูนย์กลาง 8 เซนติเมตร สูง 8 เซนติเมตร

4) ตะกร้าจักสานมีรูปร่างเว้า ขนาด เส้นผ่านศูนย์กลาง 10 เซนติเมตร สูง 6.5 เซนติเมตร

5) แก้วมีหูจับสีขาว ทรงกระบอก ขนาด เส้นผ่าศูนย์กลาง 8 เซนติเมตร สูง 10 เซนติเมตร

ผลการทดลอง

1) แก้วใสทรงกระบอก มีรูปร่างดังรูปที่ 7-2ก. นำมาผ่านกระบวนการ โดยมีค่าความละเอียด 160x120 พิกเซลจำนวนสเตป 25 สเตป ค่าเทรชโฮลด์80 แสงตกกระทบที่ตัววัตถุได้น้อยมาก หรือไปตกกระทบข้างหลังของวัตถุ ดังรูปที่ 7-2ข. ทำให้ไม่สามารถที่จะสร้างโมเดลออกมาได้



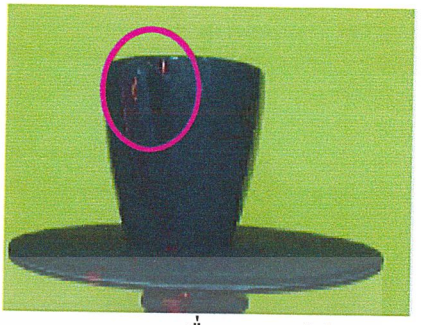
ก. ภาพของวัตถุ ข. เมื่อฉายแสงใส่วัตถุ

รูปที่ 7-2 ผลการทดลอง เมื่อนำแก้วใสมาสร้างโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) แก้วมีหูจับสีน้ำเงิน มีรูปร่างดังรูปที่ 7-3ก. นำมาผ่านกระบวนการ โดยมีค่าความละเอียด 160x120 พิกเซลจำนวนสเตป 25 สเตป ค่าเทรชโฮลด์80 แสงตกกระทบที่ตัววัตถุได้น้อยมาก ดังรูปที่ 7-3ข. ทำให้ไม่สามารถที่จะสร้างโมเดลออกมาได้

ก. ภาพของวัตถุ



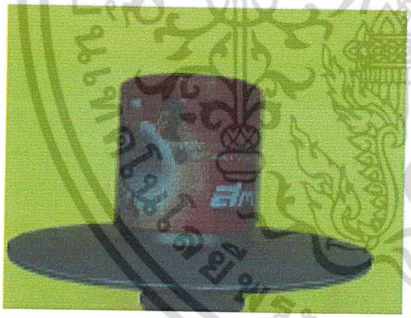
ข. ภาพเมื่อฉายแสงใส่วัตถุ

รูปที่ 7-3 ผลการทดลอง เมื่อนำแก้วมีหูจับสีน้ำเงินมาสร้างโมเดล

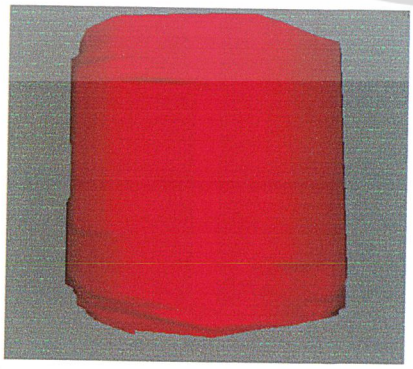
3) กล่องอะลูมิเนียมสีแดง ดังรูปที่ 7-4ก. นำมาผ่านกระบวนการ มีค่าเทรชโฮลด์80 โดยผ่านกระบวนการสองครั้ง

ครั้งที่ 1 ความละเอียด 160x120 พิกเซลจำนวนสเตป 25 สเตป จะได้โมเดลที่มีรูปร่างใกล้เคียง แต่พื้นผิวมีความผิดพลาดเล็กน้อย มีส่วนที่ไม่เรียบอยู่ ดังรูป 7-4ข.

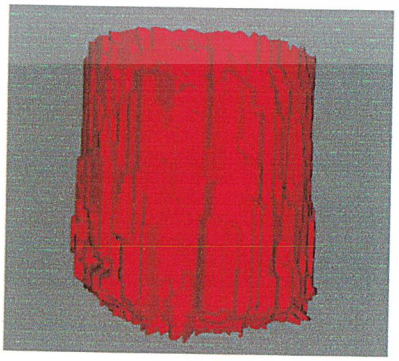
ครั้งที่ 2 ความละเอียด 320x240 พิกเซลจำนวนสเตป 200 สเตป จะได้โมเดลที่มีรูปร่างใกล้เคียง แต่พื้นผิวมีความผิดพลาดอยู่เยอะ มีส่วนที่ขรุขระจำนวนมาก ดังรูป 7-4ค.



ก. ภาพของวัตถุ



ข. โมเดลวัตถุ ความละเอียด 160x120 25สเตป



ค. โมเดลวัตถุ ความละเอียด 320x240 200สเตป

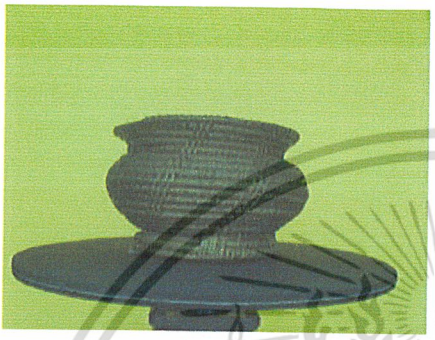
รูปที่ 7-4 ผลการทดลอง เมื่อนำกล่องอะลูมิเนียมสีแดงมาสร้างโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ตะกร้าจักสาน ดังรูปที่ 7-5ก. นำมาผ่านกระบวนการ มีค่าเทรชโฮลด์80 โดยผ่านกระบวนการสองครั้ง

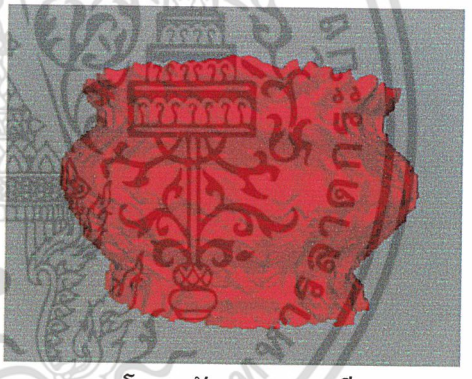
ครั้งที่ 1 ความละเอียด 160x120 พิกเซลจำนวนสเตป 25 สเตป จะได้โมเดลที่มีรูปร่างใกล้เคียง แต่พื้นผิวมีความผิดพลาดอยู่เยอะ คือจะไม่ค่อยมีความขรุขระ ดังรูป 7-5ข.

ครั้งที่ 2 ความละเอียด 320x240 พิกเซลจำนวนสเตป 200 สเตป จะได้โมเดลที่มีรูปร่างใกล้เคียงมาก พื้นผิวมีความผิดพลาดเล็กน้อย มีส่วนที่ขรุขระจำนวนมากคล้ายกับวัตถุ ดังรูป 7-4ค.



ก. ภาพของวัตถุ

ข. โมเดลวัตถุ ความละเอียด 160x120 พิกเซล 25สเตป

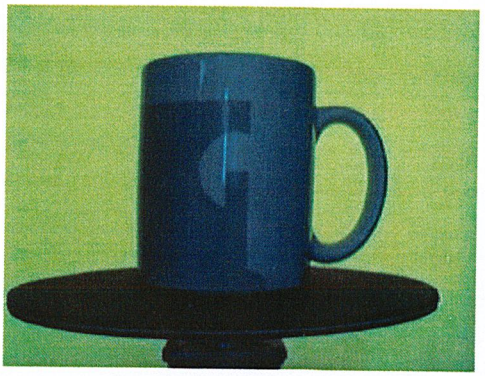


ค. โมเดลวัตถุ ความละเอียด 320x240 พิกเซล200สเตป

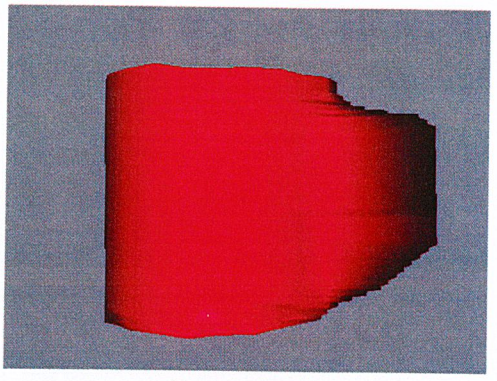
รูปที่ 7-5 ผลการทดลอง เมื่อนำตะกร้าจักสารมาสร้างโมเดล

5) แก้วมีหูจับสีขาว ดังรูปที่ 7-6ก. นำมาผ่านกระบวนการ มีความละเอียด 160x120 พิกเซล จำนวนสเตป 25 สเตป มีค่าเทรชโฮลด์80 โมเดลที่ได้จะมีส่วนที่ขาดหาย คือตรงบริเวณหูจับ ซึ่งวัตถุตรงหูจับจะมีรู แต่ตัวโมเดลที่ได้นั้นจะไม่มีรู แต่โมเดลที่ได้มีลักษณะใกล้เคียง และมีข้อมูลที่ผิดพลาดเพียงเล็กน้อยเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก. ภาพของวัตถุ



ข.โมเดลของวัตถุ

รูปที่ 7-6 ผลการทดลอง เมื่อนำแก้วมีหูจับสีขาวมาสร้างโมเดล

สรุปการทดลอง

จากการทดลองจะเห็นได้ว่าวัตถุบางประเภทไม่สามารถใช้กับระบบได้ คือ วัตถุที่มีลักษณะใส เนื่องจากทำให้แถบแสงไม่สามารถที่จะตกกระทบได้จึงทำให้ระบบไม่สามารถหาข้อมูลไปสร้างโมเดลได้ ส่วนวัตถุอีกประเภทก็คือ วัตถุที่ดูดกลืนแสง เนื่องจากแถบแสงที่ตกกระทบนั้นจะหายไปทำให้ตัวระบบไม่สามารถมองเห็นแถบแสงได้จึงทำให้ไม่มีข้อมูลไปสร้างโมเดล จากการทดลอง 1) และ 2) เพราะฉะนั้นวัตถุที่ใช้กับระบบได้ คือ วัตถุที่สามารถทำให้ระบบมองเห็นแถบแสงที่ฉายลงบนวัตถุได้อย่างชัดเจน เช่น พื้นผิวด้านสีขาว เป็นต้น

เมื่อใช้วัตถุที่ใช้กับระบบได้แล้ว ความละเอียดและจำนวนสเตป จะเป็นปัจจัยในการซุ่มข้อมูลบนผิวของวัตถุเพื่อสร้างโมเดล ซึ่งระบบนั้นยังมีความผิดพลาดของข้อมูลอยู่ จากการทดลอง 3) 4) เมื่อใช้การซุ่มข้อมูลที่มากๆกับตัววัตถุที่มีรูปร่างง่าย เช่น มีพื้นผิวเรียบ จะเกิดข้อมูลที่ผิดพลาดขึ้นมามาก จึงควรใช้การซุ่มข้อมูลน้อยๆกับวัตถุที่ทีรูปร่างง่ายๆ ส่วนวัตถุที่มีรูปร่างซับซ้อน เช่น ตะกร้าจักสาน ซึ่งมีพื้นผิวขรุขระ ควรใช้การซุ่มข้อมูลเยอะๆจึงจะได้โมเดลที่ออกมาดูเหมือนจริง กว่าการซุ่มข้อมูลน้อยๆ

วัตถุบางอย่างที่มีการบดบังนั้น จะไม่สามารถสร้างในส่วนที่ถูกบดบังได้ เนื่องจากส่วนที่ถูกบดบัง กล้องไม่สามารถที่จะบันทึกภาพนำไปคำนวนได้ จึงทำให้เกิดโมเดลที่ผิดพลาดในบริเวณที่มีการบดบัง จากการทดลองที่ 5) เช่น แก้วที่มีหูจับ จะไม่สามารถสร้างโมเดลออกมาเป็นหูจับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
บทสรุป

8.1 สรุป

สำหรับโครงงานระบบสร้างภาพคอมพิวเตอร์แบบสามมิติโดยอาศัยวีดีโอเก็บภาพของวัตถุนั้น ได้พยายามที่จะพัฒนาให้มีความสะดวกง่ายดาย และ ได้ผลที่มีความเหมือนกับวัตถุมากที่สุด ซึ่งผลที่ออกมานั้น เรื่องความสะดวก ความง่ายในการใช้งานเครื่องมือนั้น ถือว่าง่ายดายเป็นอย่างมาก แต่เรื่องความเหมือนของโมเดลของวัตถุที่ได้ออกมานั้นยังมีข้อจำกัดในเรื่องของขนาดรูปร่าง สีของวัตถุ รูปร่าง ความโค้งเว้าของวัตถุ ซึ่งการทำให้โมเดลที่ได้ออกมานั้น มีความผิดพลาดเกิดขึ้น

8.2 ปัญหาและอุปสรรค

1) เกิดความผิดพลาดของ ข้อมูลที่นำไปแปลงเป็นค่าทางสามมิติเกิดความผิดพลาด

เนื่องจากแหล่งกำเนิดแสง LED ที่ใช้นั้นให้แสงลักษณะเป็นแถบกว้างจึงทำให้การหาข้อมูลเป็นจุดนั้นต้องหา ณ จุดกึ่งกลางของแถบแสง ซึ่งทำให้เกิดความคลาดเคลื่อนของข้อมูลที่ได้มา หรือถ้าหากวัตถุที่นำมาใช้งานนั้นมี ลักษณะสะท้อนแสง ดูดซับแสง หรือมีสีเหมือนกับแหล่งกำเนิดแสงที่ใช้ ทำให้ไม่สามารถเห็นแถบแสง ทำให้ตัวโปรแกรมไม่สามารถประมวลผลแยกแยะแถบแสงออกมาได้ทำให้ข้อมูลนั้นหายไป ทำให้ไม่สามารถที่จะสร้างโมเดลได้ หรือ มีความผิดพลาดของโมเดลเกิดขึ้น

2) โมเดลที่ได้มีความผิดพลาดไปจากตัววัตถุเนื่องจากมีส่วนที่ถูกบดบัง

เนื่องจากโครงงานนี้เราได้ใช้กล้องเพียงหนึ่งตัวเท่านั้น จึงทำให้ภาพของวัตถุที่ได้เพื่อนำไปเปลี่ยนเป็นข้อมูลทางด้านสามมิตินั้น เห็นเพียงด้านเดียวทำให้เกิดการบดบังขึ้น ซึ่งเมื่อนำไปแปลงเป็นข้อมูลทางสามมิติแล้วทำการแสดงโมเดลออกมา จะทำให้เกิดความผิดพลาดเกิดขึ้น

3) ข้อมูลทางสามมิติที่ได้มีความผิดพลาด

เนื่องจากการตั้งค่าเริ่มต้นของกล้องอาจมีความผิดพลาด หรือ การจัดวางตัวฮาร์ดแวร์มีความผิดพลาดไม่ตรงกับระยะทางหรือมุมที่ใช้ในการคำนวน จึงทำให้โมเดลที่ออกมามีความผิดพลาดเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3 แนวทางในการพัฒนาต่อ

1) จากความผิดพลาดของข้อมูลที่จะนำไปแปลงเป็นค่าสามมิติ สามารถที่จะทำการแก้ไขได้โดยการเปลี่ยนแหล่งกำเนิดแหล่งแสงใหม่ จากแหล่งกำเนิดแสง LED ให้เป็น แหล่งกำเนิดแสงแบบเลเซอร์ ซึ่งทำให้ได้แถบแสงที่มีความแคบมาก ทำให้ได้จุดบนวัตถุที่นำมาหาค่าสามมิติมีความคลาดเคลื่อนน้อยลง และยังทำให้กำหนดค่าเทรชโฮลด์เพื่อลดสิ่งรบกวนออกจากระบบได้ง่ายขึ้น

2) จากความผิดพลาดของโมเดลในส่วนที่ถูกบดบังนั้น สามารถที่จะแก้ไขในบางส่วนได้ด้วยวิธีการ เพิ่มจำนวนกล้องขึ้นมา เพื่อทำให้เพิ่มความกว้างของมุมมอง ซึ่งหากเกิดมีการบดบังกล้องหนึ่ง อีกกล้องหนึ่งก็จะไม่ถูกบดบัง ทำให้ได้ข้อมูลในส่วนนั้นที่กล้องแรกถูกบดบังได้

3) หากต้องการเพิ่มความละเอียดของโมเดล สามารถที่จะทำได้ โดยการซุ่มข้อมูลของจุดบนวัตถุให้มากขึ้นกว่าเดิม ทำได้โดยการใช้กล้องที่มีความละเอียดในการบันทึกภาพที่มากขึ้น และ ใช้สเตปมอเตอร์ที่มีความละเอียดในการหมุนมากขึ้น

4) หากต้องการลดสิ่งรบกวนให้น้อยลง ก็สามารถที่จะทำได้โดยการจำกัดสิ่งแวดล้อมให้ไม่มีผลกระทบจากภายนอก เช่นการปิดกล่องสนิทเพื่อกันแสงจากภายนอก เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1]**Frank D. Luna :** "Introduction to 3D programming with DirectX 9.0",Wordware Publishing , Inc.

[2] **Peter Walsh :** "Advance 3D Game programming with DirectX 9.0",Wordware Publishing, Inc.

[3]**ยุทธนา ลีลาศวัฒนกุล:** "Visual C++ 6.0 " ,info press,2544

[4] **www.msdn.com**

[5] **www.csse.uwa.edu.au/~cheng/**

[6] **www.muellerr.ch/engineering/laserscanner/**

[7]**www.adisak51.com**

[8]**www.thaiio.com**

[9]**www.codeproject.com**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้