



การสร้างมัลติมีเดียในแบบเสมือนจริง
VIRTUAL REALITY MULTIMEDIA



โดย
นายพูนเพิ่ม สุวรรณรัฐภูมิ
น.ส.ศุคศิริ ขาวนาเจริญ
นายอมร ช่วยชู

วัน เดือน ปี...1 ตุลาคม 2541.....
เลขทะเบียน.....038294.....
เลขเรียกหนังสือ...T.313.14 N.252 f

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
สาขาวิศวกรรมคอมพิวเตอร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษา 2539 ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างมัลติมีเดียในแบบเสมือนจริง
VIRTUAL REALITY MULTIMEDIA

โดย

นายพูนเพิ่ม	สุวรรณรัฐภูมิ	37013303
น.ศ.ศุคศิริ	ชานาเจริญ	37013318
นายอมร	ช่วยชู	37013323



อาจารย์ที่ปรึกษา
อ.บรรจง ปิยธำรง

ปริญญาบัตรนี้สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาบัตรปีการศึกษา 2539

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การสร้างมัลติมีเดียในรูปแบบเสมือนจริง

ผู้จัดทำ

- | | | |
|---------------|---------------|----------|
| 1.นายพูนเพิ่ม | สุวรรณรัฐภูมิ | 37013303 |
| 2.น.ส.ศุคศิริ | ชานาเจริญ | 37013318 |
| 3.นายอมร | ช่วยชู | 37013323 |



.....อาจารย์ที่ปรึกษา
(อ.บรรจง ปิยะธำรง)

การสร้างมัลติมีเดียในรูปแบบเสมือนจริง

นายพูนเพิ่ม สุวรรณรัฐภูมิ

น.ส.สุคศิริ ชาวนาเจริญ

นายอมร ชำนาญ

อ.บรรจง ปิยธำรง อาจารย์ที่ปรึกษา

ปีการศึกษา 2539

บทคัดย่อ

ระบบมัลติมีเดียที่ใช้ในการนำเสนอข้อมูลภาพซึ่งมีอยู่ในปัจจุบัน ปกติแล้วจะให้ข้อมูลในรูปแบบของภาพ 2 มิติ ถึงแม้ว่าจะมีการจัดแสงเงาของภาพเพื่อให้ดูเหมือนมี 3 มิติ แต่จอภาพสามารถตอบสนองได้เพียง 2 มิติเท่านั้น โครงการนี้ได้นำทฤษฎีของระบบ Virtual Reality เข้ามาประยุกต์ใช้ เพื่อที่จะทำให้จอภาพที่ตอบสนองภาพได้เพียงแค่ 2 มิติเท่านั้น สามารถตอบสนองภาพให้ได้ถึง 3 มิติ โดยในโลกของ Virtual จะต้องนำเสนอผ่าน 3 D Stereo display ซึ่งทำให้ผู้ใช้สามารถมองเห็นภาพที่อยู่บนหน้าจอได้อย่างมีมิติจริง ๆ ซึ่งจะเป็นการสื่อความหมายของภาพออกมาให้ผู้ใช้เข้าใจได้ดีกว่าการนำเสนอข้อมูลภาพที่มีอยู่ในปัจจุบัน และในโครงการนี้ยังได้รวมถึงโปรแกรมที่ใช้สำหรับควบคุมลำดับของการนำเสนอภาพ เพื่อที่จะให้ภาพดำเนินเรื่องราวไปตามที่ผู้ใช้กำหนด ทำให้การเสนอภาพดูน่าสนใจมากขึ้น

VIRTUAL REALITY MULTIMEDIA

Mr.Poonperm Suwanarattaphoom

Miss Sudsiri Chaonacharoen

Mr.Amorn Choychoo

Mr.Bunjong Piyatamrong Advisor

1996

Abstract

In present time, the picture of multimedia system usually presents in form of two-dimension. Though we try to use many method to make the pictures look like three-dimension such as adjusting bright and shadow but it doesn't work. Because of the monitors can display only two-dimension, that is the problem. So we make this project. We apply the virtual reality theory to improve the quality of picture by using three-dimension stereo display. User will see the real dimension of picture that showed on the monitor. This project will reduce complex communication between user and picture. Three-dimension is more easy to communicate than two-dimension. This project includes program for controlling sequence of picture presentation. It use for controlling sequence of picture and attraction to user.

สารบัญ

หน้า

บทคัดย่อ

Abstract

สารบัญ

สารบัญภาพประกอบ

สารบัญตาราง

บทที่ 1 บทนำ.....	1
1.1 ความเป็นมา.....	1
1.2 วัตถุประสงค์.....	1
บทที่ 2 ความรู้เบื้องต้นของ virtual reality.....	2
2.1 virtual reality กับการสัมผัส.....	2
2.2 virtual reality กับการมองเห็น.....	3
2.2.1 ความรู้เบื้องต้นของการมองเห็น.....	3
2.2.2 การมองเห็นวัตถุเป็น 2 ภาพ(stereocopy).....	5
บทที่ 3 หลักการและการทำงานในระบบคอมพิวเตอร์กราฟฟิกเพื่อสร้างภาพ 3 มิติ.....	7
3.1 การสร้างงานด้วยระบบคอมพิวเตอร์กราฟฟิก 3 มิติ.....	8
3.1.1 การสร้างโครงวัตถุที่เป็นลายเส้น(Model Object).....	8
3.1.2 การกำหนดคุณสมบัติต่าง ๆ ให้กับวัตถุ(Assign Materials).....	11
3.1.3 การจัดองค์ประกอบภายในฉาก.....	11
3.1.4 การกำหนดการเคลื่อนไหวให้กับวัตถุ (Animation).....	12
3.1.5 การทำ special effect.....	13
3.1.6 การทำ Render.....	13
3.2 วิธีการเข้าสู่โปรแกรม.....	14
3.3 เข้าใจส่วนการทำงานหลักภายในโปรแกรม.....	15
3.3.1 2D SHAPER.....	15
3.3.2 3D LOFTER.....	16
3.3.3 3D EDITOR.....	16

สารบัญ (ต่อ)

3.3.5 MATERIAL EDITOR.....	18
3.4 หลักการสร้างภาพจาก 3D Studio.....	19
บทที่ 4 พื้นฐานการแสดงผล.....	21
4.1 การเลือกส่วนแสดงผล.....	22
4.2 โหมดการทำงานมาตรฐานของ EGA.....	24
4.3 โหมดการทำงานเพิ่มเติมสำหรับ VGA.....	31
4.4 สถาปัตยกรรม EGA/VGA.....	32
4.4.1 การอัปเดตข้อมูลจุดภาพและการใช้เพดลีสตี.....	32
4.4.2 โหมดข้อความและโหมดกราฟฟิก.....	33
4.4.3 สถาปัตยกรรมของ EGA และ VGA.....	34
4.4.4 การทำงานของจอ CRT.....	35
4.4.5 หน่วยความจำแสดงผล.....	36
4.4.6 ตัวควบคุมกราฟฟิก.....	47
บทที่ 5 หลักการทำงานของแวนดา LCD และรูปแบบของภาพเคลื่อนไหวและภาพนิ่ง.....	53
5.1 หลักการทำงานของแวนดา LCD Shutter.....	53
5.2 การทำงานของไฟล์ FLI.....	54
5.2.1 Header ของรูปแบบ FLI.....	55
5.2.2 กลุ่มข้อมูล FLI_COLOR.....	56
5.2.3 กลุ่มข้อมูล FLI_LC.....	57
5.2.4 กลุ่มข้อมูล FLI_BLACK.....	58
5.2.5 กลุ่มข้อมูล FLI_BRUN.....	58
5.2.6 กลุ่มข้อมูล FLI_COPY.....	58
5.3 การทำงานของไฟล์ PCX.....	58
5.3.1 256 color PCX ไฟล์.....	60
5.3.2 Run length encoding.....	60
บทที่ 6 การออกแบบและการสร้าง.....	63
6.1 ขั้นตอนในการดำเนินงาน.....	63
6.2 การออกแบบทางด้านฮาร์ดแวร์(Hardware design).....	64

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

6.2.1	ภาคควบคุมวงวนตา LCD.....	64
6.2.2	ภาคควบคุม.....	67
6.2.3	ภาครับส่งข้อมูลกับคอมพิวเตอร์.....	68
6.3	การออกแบบทางด้านซอฟต์แวร์ (Software design).....	71
6.3.1	Graphic editor.....	72
6.3.2	Covert ข้อมูลภาพ.....	72
6.3.3	Capture input.....	73
6.3.4	Generate script file.....	73
6.3.5	การเรียกดูข้อมูล.....	75
6.3.6	การ Run script ไฟล์.....	75
6.3.7	โปรแกรมเปิดไฟล์ภาพ.....	76
บทที่ 7	บทวิจารณ์และสรุป.....	79
7.1	สรุปผลการทดลอง.....	79
7.2	แนวทางในการพัฒนาต่อ.....	82

กิตติกรรมประกาศ

หนังสืออ้างอิง

สารบัญภาพประกอบ

หน้า

รูปที่ 2.1	ภาพแสดงส่วนต่าง ๆ ของตามนุษย์.....	3
รูปที่ 2.2	ภาพแสดงรายละเอียดส่วนต่าง ๆ ของเรติน่าและเส้นประสาท.....	4
รูปที่ 2.3	ภาพแสดงระยะห่างระหว่างศูนย์กลางของม่านตา.....	4
รูปที่ 2.4	ภาพแสดงการทำงานของสมองในการมองภาพของตาซ้ายและตาขวา.....	5
รูปที่ 2.5	มุมมองของตาซ้ายและตาขวาในการมองภาพกล้อง.....	6
รูปที่ 2.6	แสดงการมองวัตถุระยะใกล้,ระยะกลางและระยะไกล.....	6
รูปที่ 3.1	ตัวอย่างโครงรูป 3 มิติ.....	8
รูปที่ 3.2	หลักการและวิธีที่ใช้ในการสร้างแจกัน.....	9
รูปที่ 3.3	(ก) ลักษณะของวัตถุและภาพตัดขวางของวัตถุ.....	9
	(ข) ลักษณะโครงรูป 2 มิติ.....	9
รูปที่ 3.4	แสดงการกำหนดทิศทางการสร้างวัตถุและนำโครงรูปมาจัดวางในเส้นทางที่กำหนด.....	10
รูปที่ 3.5	แสดงเส้นทางการหมุนเข้าสู่จุดศูนย์กลาง.....	10
รูปที่ 3.6	ลักษณะของวัตถุรูปแจกันเมื่อนำเอาคุณสมบัติแบบต่าง ๆ มาใส่บนผิววัตถุ.....	11
รูปที่ 3.7	แสดงการเคลื่อนที่ของลูกบอลในแต่ละเฟรม.....	12
รูปที่ 3.8	ลักษณะของวัตถุที่เป็นลายเส้นและทำการจัดองค์ประกอบในฉากเรียบร้อยแล้ว.....	13
รูปที่ 3.9	ลักษณะของภาพที่เกิดขึ้นหลังจากการทำ Render.....	14
รูปที่ 3.10	ลักษณะหน้าจอของส่วนการทำงานแรกเมื่อเข้าสู่โปรแกรม Autodesk 3D Studio....	14
รูปที่ 3.11	ลักษณะหน้าจอของส่วนการทำงานหลัก 2D SHAPER.....	15
รูปที่ 3.12	ลักษณะหน้าจอของส่วนการทำงานหลัก 3D LOFTER.....	16
รูปที่ 3.13	ลักษณะหน้าจอของส่วนการทำงานหลัก 3D EDITOR.....	17
รูปที่ 3.14	ลักษณะหน้าจอของส่วนการทำงานหลัก KEY FRAMER.....	17
รูปที่ 3.15	ลักษณะหน้าจอของส่วนการทำงาน Materials Editor.....	18
รูปที่ 3.16	การกำหนดให้ดูแล้ววัตถุหมุนออกมาจากจอ.....	19
รูปที่ 3.17	การกำหนดให้ดูแล้ววัตถุลึกลงไปในจอภาพ.....	19
รูปที่ 3.18	การกำหนดให้ภาพมีทั้งหมุนและลึกลงในภาพเดียวกัน.....	20
รูปที่ 4.1	เพจการแสดงผลในโหมด 0 และ 1.....	26
รูปที่ 4.2	ข้อมูลสีที่เก็บแบบอ็อดจคุณภาพ.....	32

สารบัญภาพประกอบ (ต่อ)

รูปที่ 4.3 ข้อมูลสีที่เก็บแบบแยกเฟลนสี.....	33
รูปที่ 4.4 การแสดงผลโหมดข้อความและโหมดกราฟฟิก.....	33
รูปที่ 4.5 บล็อกไดอะแกรมของ EGA/VGA.....	34
รูปที่ 4.6 จอแสดงผล.....	35
รูปที่ 4.7 การทำงานของจอ CRT.....	35
รูปที่ 4.8 หน่วยความจำแสดงผล(Display memory).....	36
รูปที่ 4.9 หน่วยความจำแสดงผลในโหมดข้อความ.....	37
รูปที่ 4.10 แอดเดรสอ้างอิงโดยตัวประมวลผลและเฟลนบน EGA.....	37
รูปที่ 4.11 การใช้รหัสอักษรตัวชี้ตัวสร้างอักษร.....	38
รูปที่ 4.12 รูปแบบการเก็บตัวสร้างอักษร.....	39
รูปที่ 4.13 หน้าที่ของแต่ละบิตในไบต์แอดทริบิวต์.....	40
รูปที่ 4.14 การแมปหน่วยความจำในการแสดงกราฟฟิกแบบ CGA โหมด 6.....	42
รูปที่ 4.15 การแมปหน่วยความจำในการแสดงกราฟฟิกแบบ CGA โหมด 4 และ 5.....	42
รูปที่ 4.16 การแมปหน่วยความจำในการแสดงกราฟฟิกโหมด F.....	44
รูปที่ 4.17 การแมปหน่วยความจำในการแสดงกราฟฟิกโหมด F.....	45
รูปที่ 4.18 การแมปหน่วยความจำในการแสดงกราฟฟิกโหมด 11.....	46
รูปที่ 4.19 การแมปหน่วยความจำในการแสดงกราฟฟิกโหมด 12.....	46
รูปที่ 4.20 ตัวควบคุมกราฟฟิก.....	47
รูปที่ 4.21 การทำงานของตัวควบคุมกราฟฟิก.....	48
รูปที่ 4.22 ตัวควบคุมแอดทริบิวต์.....	49
รูปที่ 4.23 การทำงานของตารางค้นหาสีในตัวควบคุมแอดทริบิวต์.....	50
รูปที่ 4.24 ตัวควบคุม CRT.....	51
รูปที่ 4.25 ซีเควนเซอร์.....	51
รูปที่ 4.26 การเลือกเฟลนสีของซีเควนเซอร์.....	52
รูปที่ 5.1 แสดงถึง Negative Parallax.....	53
รูปที่ 5.2 แสดงถึง Zero Parallax.....	53
รูปที่ 5.3 แสดงถึง Positive Parallax.....	54
รูปที่ 5.4 แสดงถึงโครงสร้างของ FLI file.....	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพประกอบ (ต่อ)

รูปที่ 5.5 แสดงถึงการ ใช้ delta ring frame ในการกลับไปเริ่มต้นที่ frame ที่ 1.....	55
รูปที่ 5.6 Format ของ Pallete Chunk (FLI_COLOR).....	57
รูปที่ 5.7 Format ของ FLI_LC image chunk.....	57
รูปที่ 5.8 Format ของ FLI_BRUN image chunk.....	58
รูปที่ 5.9 Structure ของ PCX graphic file.....	60
รูปที่ 5.10 Flowchart แสดงการ decode row เพื่อ compressed pixel จาก file .PCX.....	61
รูปที่ 5.11 แสดง function unpack line in PCX file.....	61
รูปที่ 6.1 แสดงแผนผังการดำเนินงาน.....	63
รูปที่ 6.2 บล็อกไดอะแกรมแสดงการทำงานทั้งหมดของวงจร.....	64
รูปที่ 6.3 บล็อกไดอะแกรมแสดงส่วนของการควบคุมแวนดา.....	64
รูปที่ 6.4 วงจรควบคุมการทำงานของแวนดา LCD.....	65
รูปที่ 6.5 รูปแสดงการสแกนของอิเล็กทรอนิกส์บนหน้าจอ.....	66
รูปที่ 6.6 (ก) รูปของสัญญาณที่ป้อนเข้า LCD ที่ยังไม่ได้ทำการมอดูเลท.....	67
(ข) รูปของสัญญาณที่ความถี่ 1.2 kHz ที่จะทำการมอดูเลท.....	67
(ค) รูปของสัญญาณที่ป้อนเข้า LCD ที่ทำการมอดูเลทแล้ว.....	67
รูปที่ 6.7 บล็อกไดอะแกรมแสดงภาคควบคุม.....	68
รูปที่ 6.8 แสดงวงจรภาคควบคุม.....	68
รูปที่ 6.9 บล็อกไดอะแกรมแสดงส่วนของภาค buffer.....	68
รูปที่ 6.10 แสดงวงจรของภาค buffer.....	69
รูปที่ 6.11 รูปแสดงวงจรรวม.....	70
รูปที่ 6.12 บล็อกไดอะแกรมแสดงการทำงานทางด้านซอฟต์แวร์โดยรวม.....	71
รูปที่ 6.13 แสดงการทำงานของส่วน Graphic editor.....	72
รูปที่ 6.14 แสดงการ Convert ข้อมูลภาพ.....	73
รูปที่ 6.15 แสดงหลักการการทำงานในการ Capture input.....	74
รูปที่ 6.16 แสดงหลักการการทำงานในการ Generate script file.....	74
รูปที่ 6.17 แสดงการเรียกดูข้อมูลภาพ.....	75
รูปที่ 6.18 แสดงหลักการการทำงานในการ run script ไฟล์.....	76
รูปที่ 6.19 แสดงหลักการการทำงานในการเปิดภาพเคลื่อนไหว.....	77

สารบัญภาพประกอบ (ต่อ)

รูปที่ 6.20 แสดง subroutine ในการอ่านเฟรมต่อไป.....	78
รูปที่ 7.1 (ก) ในช่วงเวลาแรก (ข) ช่วงเวลาต่อมา.....	79
รูปที่ 7.2 แสดงระยะห่างของกล้องและจุดตัดการวางกล้อง.....	80
รูปที่ 7.3 การมองวัตถุที่มีฉากหลัง.....	80
รูปที่ 7.4 แสดงตำแหน่งของจุดตัดมุมกล้องใน 3D Studio เมื่อปรากฏบนหน้าจอ.....	81
รูปที่ 7.5 แสดงการวางมุมกล้องตั้งฉากกับแนววางกล้อง.....	81



สารบัญตาราง

	หน้า
ตารางที่ 4.1 จอแสดงผลที่คอมแพคทีเบลกับ ไอพีเอ็มพีซี.....	21
ตารางที่ 4.2 โหมดการแสดงผลตามมาตรฐาน ไอพีเอ็ม.....	24
ตารางที่ 4.3 แอตทริบิวต์ข้อความมาตรฐาน.....	26
ตารางที่ 4.4 ชุดสีมาตรฐานโหมด 4 และ 5.....	27
ตารางที่ 4.5 สีมาตรฐานสำหรับโหมดการแสดงผล 16 สี.....	30
ตารางที่ 4.6 ตำแหน่งของตัวสร้างอักษรที่ฝังตัวบนแรม.....	39
ตารางที่ 4.7 แอตทริบิวต์สีมาตรฐาน.....	40
ตารางที่ 4.8 แอตทริบิวต์ข้อความโมนโครม.....	41
ตารางที่ 4.9 สีมาตรฐานโหมด 4 และ 5.....	43
ตารางที่ 5.1 FLI header ขนาด 128 ไบต์.....	55
ตารางที่ 5.2 ค่าที่หักจาก header คอนตันจะเป็นส่วนของ Frame header.....	56
ตารางที่ 5.3 หลังจาก frame header จะเป็นส่วน header ของแต่ละชุดข้อมูล(chunk header). 56	
ตารางที่ 5.4 ตารางแสดงค่าของรูปแบบของแต่ละกลุ่มข้อมูล.....	56
ตารางที่ 5.5 แสดงส่วนของ header ของไฟล์ในรูปแบบ PCX.....	59

บทที่ 1

บทนำ

1.1 ความเป็นมา

เนื่องจากในปัจจุบันระบบคอมพิวเตอร์ได้เข้ามามีบทบาทอย่างมาก ทั้งในด้านการศึกษา ด้านโฆษณา ด้านอุตสาหกรรม และอื่น ๆ แรกเริ่มเราจะใช้คอมพิวเตอร์ในการเก็บข้อมูลและแสดงผลข้อมูลในรูปแบบของตัวอักษร (text) จากนั้นได้พัฒนาต่อมาจนสามารถแสดงผลเป็นแบบกราฟฟิก (graphic) ทำให้มีรูปแบบที่สวยงาม น่าใช้มากขึ้น และในปัจจุบันได้มีการใช้เทคโนโลยีทางด้านมัลติมีเดีย (Multimedia) เพื่อช่วยให้ทำงานได้สะดวก รวดเร็วยิ่งขึ้น ซึ่งได้รับความนิยมแพร่หลายเป็นอย่างมาก ระบบมัลติมีเดียที่ใช้กันโดยทั่วไปจะเป็นการนำเสนอ ภาพในสองมิติแบบธรรมชาติ ซึ่งจะขาดความสมจริง ทำให้ระบบมัลติมีเดียแบบเดิมนั้นไม่ดึงดูดความสนใจหรือน่าติดตาม ดังนั้นผู้จัดทำโครงการนี้จึงได้มีแนวคิดที่จะจัดทำระบบมัลติมีเดียให้เป็นระบบสามมิติ (3 Dimension) โดยการนำเทคโนโลยีระบบเสมือนจริง (Virtual Reality) มาใช้ เพื่อที่จะให้ระบบมัลติมีเดียที่ใช้มีความสมจริง ซึ่งจะต้องใช้อุปกรณ์ประกอบ คือ แว่นตาสามมิติ (LCD Shutter Glasses) และตัวอ้างอิงตำแหน่งระบบสามมิติ เพื่อที่จะให้สามารถแสดงภาพนิ่งและภาพเคลื่อนไหวได้อย่างสมจริง ดังนั้นในการสร้างภาพที่จะนำมาทำให้เป็น 3 มิติ จะต้องใช้ editor สำหรับสร้างภาพเสมือนจริงได้ ที่สามารถสร้างได้ง่ายและสะดวก ฉะนั้นโครงการนี้จึงได้เลือกโปรแกรม 3D Studio มาใช้ในการสร้างภาพ ซึ่งตัวโปรแกรม 3D Studio นี้มีความสามารถและมีความสะดวกในการใช้งานเป็นอย่างมาก

ปัจจุบันเทคโนโลยีทางด้านระบบเสมือนจริง ในประเทศไทยยังไม่ได้รับการศึกษาเท่าที่ควร จึงทำให้เทคโนโลยีทางด้านนี้ไม่ได้รับการพัฒนาเพื่อนำมาใช้งานอย่างเต็มที่ แนวทางต่อไปในอนาคตเมื่อเทคโนโลยีระบบเสมือนจริงได้รับการศึกษาและพัฒนาให้มากขึ้นแล้ว ก็จะสามารถนำไปประยุกต์ใช้ให้เกิดประโยชน์อย่างเต็มประสิทธิภาพ

1.2 วัตถุประสงค์

1. เพื่อศึกษาการทำงานของ Virtual Reality
2. เพื่อศึกษาการประยุกต์ใช้ Multimedia เข้ากับ Virtual Reality
3. เพื่อศึกษาวิธีการสร้างภาพ 3 มิติเสมือนจริง
4. เพื่อศึกษาการประยุกต์ใช้ระบบ Multimedia และ Virtual Reality เข้ากับงาน Presentation

บทที่ 2

ความรู้เบื้องต้นของ Virtual Reality

Virtual reality (VR) คือระบบเสมือนจริง ซึ่งสามารถแบ่งออกเป็นประเภทได้ดังนี้คือ การได้ยิน, การมองเห็น, การสัมผัส กลิ่นและรสชาติ แต่ระบบ virtual reality นี้จะทำการพิจารณาโดยปราศจากสิ่งที่เป็นตัวตนจริง ๆ นั่นก็คือเป็นการเลียนแบบพฤติกรรมของมนุษย์โดยใช้คอมพิวเตอร์มาสร้างสภาวะแวดล้อมและมีการโต้ตอบกับผู้ที่ใช้ virtual reality ประสาทสัมผัสไม่ใช่ความรู้สึกที่เกิดขึ้นจริงโดยตรงกับตัวเรา เพราะจริงๆ แล้วจะรับความรู้สึกเข้ามาและจะถูกเชื่อมต่อกับสมอง ซึ่งสมองจะเป็นตัวสร้างความรู้สึกจริง ๆ ที่เกิดขึ้น ดังนั้นภายในสมองจะมีการทำปฏิกิริยาเคมีซึ่งเกิดจากไฟฟ้า (electrochemical) แต่จริงแล้วสมองไม่ได้ทำด้วยตัวมันเองทั้งหมด ตามหลักชีววิทยาสมองจะสร้างสิ่งต่าง ๆ ขึ้นมาเหนือสิ่งอื่น ๆ (superstructure) เพื่อสนับสนุนการทำ bubbling cauldron ของการสร้างปฏิกิริยาเคมีที่เกิดจากไฟฟ้าที่จะเรียกว่าความรู้สึก (consciousness) เมื่อเนื้อเยื่อและโครงสร้างทางชีววิทยา (texture) ของเรสัมผัสและรู้สึกถึงวัตถุ เราไม่ได้สัมผัสมันด้วยตัวเอง แต่ด้วยประสาทสัมผัสซึ่งจะถูกเปลี่ยนแปลงเป็นความรู้สึกที่สัมผัสได้โดยสมองเรา ดังนั้น virtual reality คือการสังเคราะห์ (synthetic), 3 มิติ (three dimensional), การใช้คอมพิวเตอร์สร้างสภาพแวดล้อมการทำงานร่วมกับผู้ใช้, การรู้สึก, การได้ยิน, และทำงานร่วมกันในการสังเคราะห์สภาวะแวดล้อม การใช้งานของ virtual reality มีหลายประเภทดังนี้

- Entertainment
- VR meets wall street
- Architecture
- Computer aided design (CAD)
- Computer aided engineering (CAE)
- Flight Simulation
- Space Simulation
- Virtual navy
- Intelligent agents
- VR and surgery
- Telepresence - taking your body where your mind dares to travel

2.1 Virtual Reality กับการสัมผัส

การสัมผัสเป็นการสื่อสารที่ทำให้มนุษย์สามารถเข้าใจถึงแรงกระทบ ทำให้มนุษย์สามารถประมาณแรงที่กระทำกับวัตถุได้ ดังนั้นในด้าน virtual reality แล้วจะเป็นการสร้างสภาวะแวดล้อมเพื่อจำลองแรงกระทบหรือแรงต้านเพื่อให้มนุษย์สามารถรับรู้ได้ ซึ่งปัจจุบันมักนำไปใช้ในระบบของ Robotics เช่นการหยิบวัตถุด้วยแขนกล แต่ต้องการด้วยแขนและมือของมนุษย์ และคอมพิวเตอร์จะทำการควบคุมแรงต้านให้กับมือและแขนของมนุษย์เหมือนได้ไปจับวัตถุนั้นจริง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

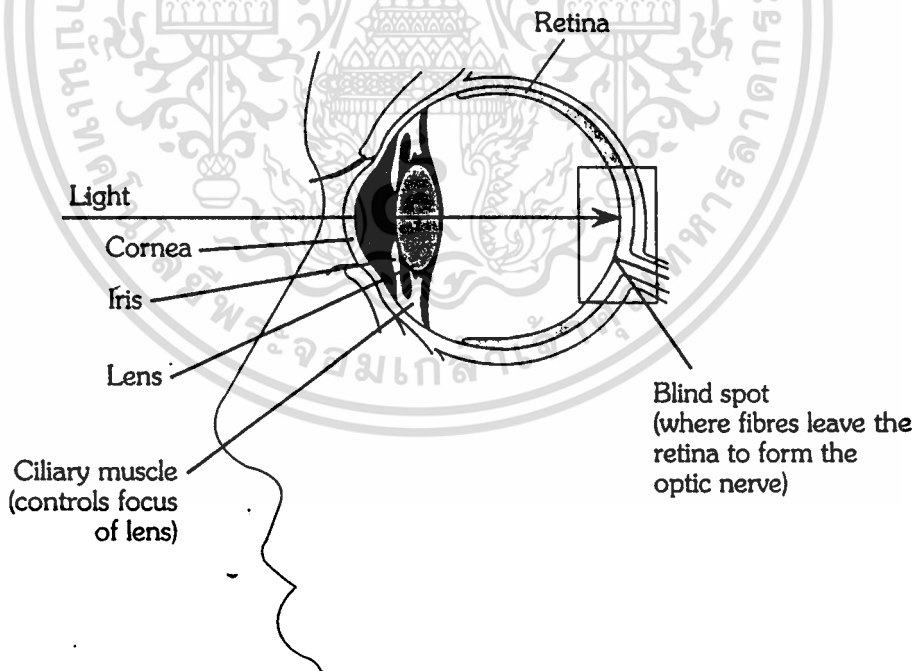
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Virtual Reality กับการมองเห็น

การมองเห็นเป็นการสื่อสารที่ทำให้มนุษย์สามารถที่จะเข้าใจสิ่งต่าง ๆ ได้ดีที่สุด สามารถแยกแยะได้ว่ามีความชัด,ลึก ทำให้มนุษย์สามารถประมาณระยะห่างของสิ่งที่กำลังดูได้ ดังนั้นในด้าน virtual reality แล้วจะเป็นการสร้างสภาวะแวดล้อม เพื่อจำลองให้เป็น 3 มิติ ในลักษณะเดียวกันกับที่มนุษย์มองเห็น ซึ่งปัจจุบันมีการนำไปใช้ในหลาย ๆ ด้าน เช่น จำลองการผ่าตัด ซึ่งตัวแพทย์จะลงมือทดสอบกับคนไข้จริงจะทำให้เกิดความเสี่ยง แต่ด้วยเทคโนโลยี virtual reality แล้วทำให้แพทย์ลงมือกระทำการผ่าตัดได้เหมือนกับการผ่าตัดคนไข้จริง ๆ จึงทำให้เกิดความเสี่ยงลดลง

2.2.1 ความรู้เบื้องต้นของการมองเห็น

ตา คือ optical device ประกอบด้วย lenses โดย lenses ของตาจะทำหน้าที่ปรับระยะชัดของภาพที่อยู่ภายนอกผ่านไปยัง pupil และตกกระทบบน retina ม่านตา(Iris) รูของสี(pigmented), เส้นใยกล้ามเนื้อที่รวมกันอยู่เป็นวงกลมและทิศทางของรัศมี เมื่อทำการเปิดดูศูนย์กลางของม่านตาเราจะเรียกว่า pupil ม่านตาจะทำการปรับจำนวนของแสงที่เข้ามายังดวงตาให้ลดลง(relaxes) หรือทำการทำให้สั้นลง(contract) เพื่อให้ได้ระดับที่ดี (optimum level) สำหรับการประมวลผลภาพโดย retina

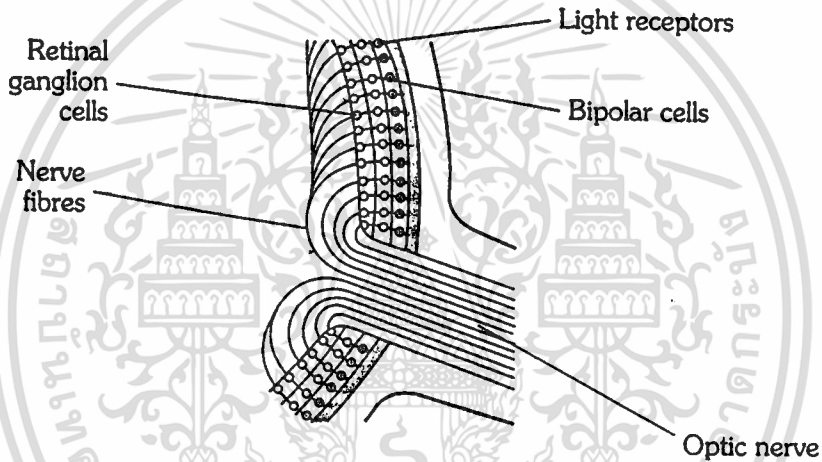


รูปที่ 2.1 ภาพแสดงส่วนต่าง ๆ ของตามนุษย์

retina คือ เนื้อเยื่อบาง ๆ ของเขตที่ไวแสง (photosensitive cell) ลักษณะเป็นเส้นที่อยู่ภายในดวงตา ภาพต่าง ๆ จะตกกระทบบน cell ซึ่งมันจะทำการเปลี่ยนเป็นแรงดันไฟฟ้า (electrical impulse) และจะส่งเอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

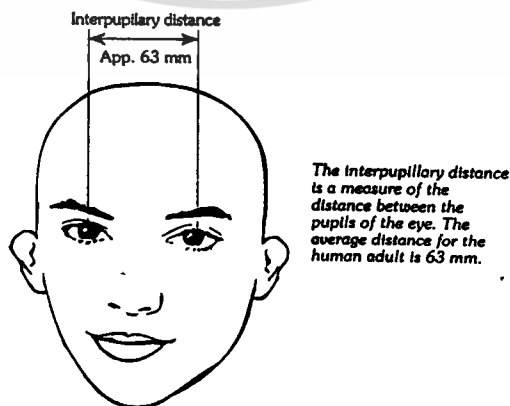
ไปยังสมองโดยผ่านเส้นประสาท (optical nerve) สมองจะทำการแปลงแรงดันไฟฟ้าเป็นรูปแบบของภาพที่เราเข้าใจ

color vision เซลล์ที่ไวแสงที่สร้างใน retina จะถูกเรียกว่า receptor cells และยังสามารถแบ่งได้เป็น 2 ชนิดคือ rods และ cones โดย rod cell จะไม่มีความไวต่อสี แต่จะมีความไวต่อแสงมาก โดยปกติแล้ว rod cell มักจะอยู่รอบๆ นอกของ retina คุณภาพของภาพจะอยู่ที่รูปแบบของ rod cell ด้วย ถ้า rod cell หยิบจะทำให้เรามองเห็นภาพไม่ชัดคือ มัว cone cell จะมีความไวต่อสีและรายละเอียดต่าง ๆ ของภาพ ปกติแล้ว cone cell จะอยู่ที่ส่วนตรงกลางของ retina cone cell มีอยู่ 3 ประเภทแต่ละตัวจะมีความไวเพียง 1 สี ดังนั้นตาจะมีความไวต่อสีพื้นฐาน (primary colors) อยู่ 3 สีเท่านั้น แม้ว่าจะมีสีเป็นล้านสี ล้าน shade นั้นคือมันจะนำข้อมูลที่ได้จากแต่ละชนิดของ cone cell มาผสมกันกับข้อมูลของแสงที่สว่างและมีที่รับมาจาก rod cell



รูปที่ 2.2 ภาพแสดงรายละเอียดส่วนต่างๆ ของเรตินาและเส้นประสาทตา

มุมมองของตาแต่ละตาจะมีมุมมองประมาณ 90° สองตาารวมกันก็จะมีมุมมอง 180° ตายังสามารถเคลื่อนที่ได้อีก 45° ไปทางซ้ายและทางขวา ซึ่งจะทำให้มีมุมมองเป็น 270°

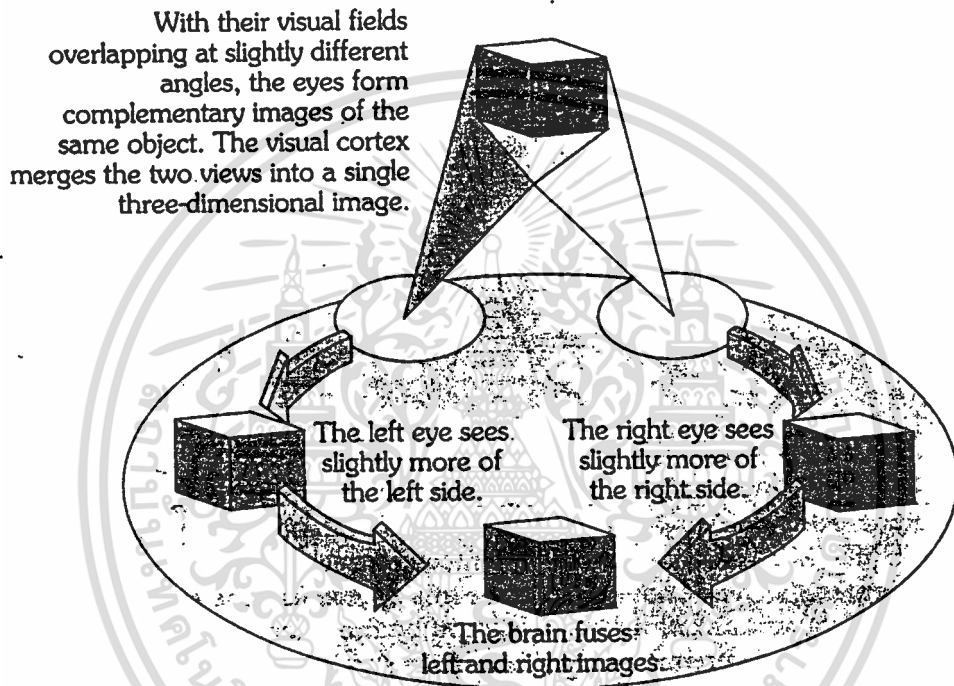


รูปที่ 2.3 ภาพแสดงระยะห่างระหว่างศูนย์กลางของม่านตา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การมองเห็นวัตถุเป็น 2 ภาพ (stereoscopy)

ตาเราจะมองเห็นในแบบ 3 มิติ เพราะเรามีตา 2 ตา แต่ละตาดูจะมองเห็นในมุมมองที่ต่างกัน สมองจะผสมภาพทั้งสองรวมกัน ซึ่งจะทำให้มองเห็นความชัดลึกได้

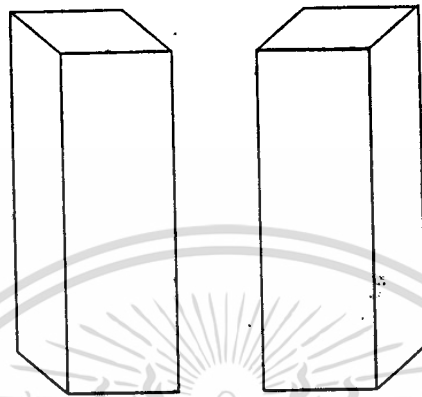


รูปที่ 2.4 ภาพแสดงการทำงานของสมองในการมองภาพของตาซ้ายและตาขวา

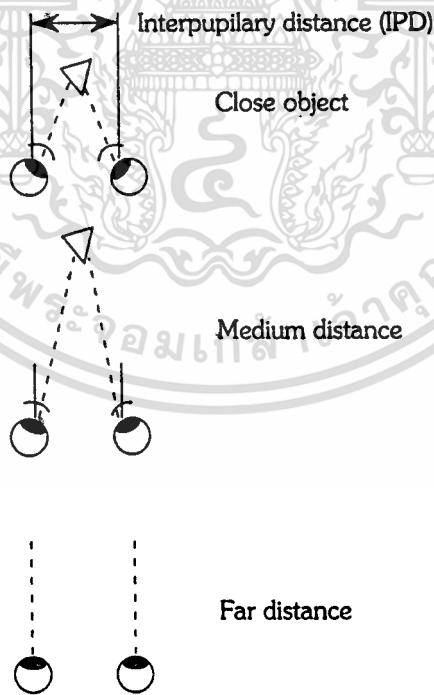
ระยะห่างระหว่างดวงตา (interpupillary distance : IPD) มีค่าระหว่าง 53 ถึง 73 มิลลิเมตร ซึ่งเป็นของทั้งผู้หญิงและผู้ชาย ซึ่งมีค่าเฉลี่ยประมาณ 63 มิลลิเมตร (ประมาณ 2.5 นิ้ว) สำหรับเด็กจะมี IPD น้อย IPD จะเป็นตัวกำหนดระยะห่างของมุมมองภาพที่รับได้ทั้งตาซ้ายและขวา

ในระบบ virtual reality ต้องมีการกำเนิด object อยู่ในแบบ 3 มิติ ในลักษณะของ stereoscopic image ซึ่งคอมพิวเตอร์จะทำหน้าที่สร้าง stereoscopic image แต่ละภาพจะถูกนำเสนอไปยังแต่ละตา ในมุมมองที่ต่างกัน การเปลี่ยนมุมมองของภาพบนหน้าจอก็จะต้องจัดมุมตามระยะห่างของดวงตา สมองจะทำการรวมภาพ 2 ภาพ เข้าด้วยกันเพื่อให้เป็นภาพแบบ 3 มิติ ดังนั้นความลึกของภาพที่ใช้ stereographic จะอยู่ที่ความเอนเอียงของภาพ เพราะว่ามันเอนเอียงมรววมอยู่ที่จุดเดียวกันไม่ถูกต้อง ความแตกต่างของภาพทั้ง 2 ภาพ เรียกว่า eye's parallax หรือ stereoscopic image

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 มุมมองของตาซ้ายและตาขวาในการมองภาพล่อง



รูปที่ 2.6 แสดงการมองวัตถุระยะใกล้,ระยะกลางและระยะไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการและการทำงานในระบบคอมพิวเตอร์กราฟฟิกเพื่อสร้างภาพ 3 มิติ

คอมพิวเตอร์กราฟฟิกเป็นหัวข้อที่น่าสนใจหัวข้อหนึ่ง เนื่องจากกลไกในการสร้างและควัด 3 มิติ โดย ไม่จำเป็นที่ต้องมีวัตถุอาจจะเป็น โลโก้บริษัท, รถยนต์, อาคารใหม่ที่ยังไม่ถูกสร้าง, ร่างกายมนุษย์, โมเดล ฯลฯ ซึ่งไม่มีข้อจำกัดในการสร้างวัตถุ ขึ้นอยู่กับจินตนาการของผู้สร้าง โดยผู้สร้างต้องจำลองแบบลงไปบนเครื่องคอมพิวเตอร์ กำหนดข้อมูลต่าง ๆ ให้กับวัตถุ หลังจากนั้นเครื่องคอมพิวเตอร์จะทำการคำนวณ และแสดงผลทางด้านภาพออกมา นอกจากนี้ผู้สร้างอาจกำหนดการเคลื่อนไหวต่าง ๆ ให้กับวัตถุได้ ซึ่งขั้นตอนเหล่านี้ขึ้นอยู่กับความสามารถของโปรแกรม และความเข้าใจในการใช้โปรแกรมของผู้สร้างงาน

คอมพิวเตอร์กราฟฟิกถูกนำมาใช้อย่างกว้างขวาง เช่น ในวงการโฆษณาหรือทำภาพยนตร์ โดยสื่อที่สร้างจากเครื่องคอมพิวเตอร์ สามารถสร้างภาพพจน์และความเชื่อถือให้กับลูกค้าได้มาก ยิ่งไปกว่านั้น ในบางครั้งการสร้างงานด้วยเครื่องคอมพิวเตอร์ ยังสามารถสร้างงานได้สมจริงสมจังมากกว่าวิธีอื่น ๆ ตัวอย่างเช่น หนังสือเรื่องจูราสสิกปาร์ค ภาพไดโนเสาร์ที่สร้างจากเครื่องคอมพิวเตอร์สามารถทำได้เหมือนจริง จนรู้สึกว่าได้โนเสาร์ยังมีชีวิตอยู่บนโลกนี้จริง ๆ

ในงานทางด้านอุตสาหกรรม เช่น การทำแบบจำลองรูปรถ, การจำลองส่วนประกอบต่าง ๆ ของรถ, การทดสอบระบบ Airo-Dynamic ก่อนที่จะลงมือสร้างรถจริง ๆ , การจำลองรูปแบบโครงสร้างของอาคาร, การออกแบบระบบการเดินท่อ เป็นต้น ในปัจจุบันถึงต่าง ๆ เหล่านี้ได้มีการนำเอาคอมพิวเตอร์ไปประยุกต์ใช้เพื่อแก้ปัญหาในการสร้างงาน ทำให้ขั้นตอนในการทำงานต่าง ๆ สั้นลงและสะดวกขึ้น สำหรับงานบางประเภทเมื่อนำเอาคอมพิวเตอร์กราฟฟิกเข้าไปใช้แล้ว สามารถประหยัดได้ทั้งเงิน และความเสี่ยงที่อาจเกิดขึ้นได้ ตัวอย่างเช่น งานทางด้านการศึกษา อย่างที่การบินไทยมีห้องสำหรับฝึกสอนการบินด้วยระบบ Flight Simulators คอมพิวเตอร์จะจำลองสภาพสนามบินต่าง ๆ เพื่อให้ผู้ฝึกได้ฝึกในสภาพแวดล้อมและสถานการณ์ต่าง ๆ จนชำนาญ แล้วค่อยไปฝึกบนเครื่องบินจริง ทำให้ลดความเสี่ยงที่อาจเกิดขึ้นได้อย่างมากมาย ทั้งในรูปแบบของทรัพย์สิน ชีวิตผู้สอน และผู้ฝึกหัด

สำหรับการศึกษานั้น คอมพิวเตอร์ได้เข้ามามีบทบาทอย่างมาก เนื่องจากการเรียนในหัวข้อบางหัวข้อมีแต่ทฤษฎีและตัวเลขทางด้านคณิตศาสตร์เท่านั้น เช่น สูตรโครงสร้างทางด้านเคมี เช่น น้ำ (H_2O) ถ้ามีการจำลองรูปแบบให้เกิดเป็นภาพขึ้นมาแล้ว ความสามารถในการทำความเข้าใจของนักเรียนก็เพิ่มมากขึ้น นอกเหนือจากการนำเอาข้อมูลจากการทดลองมาสื่อออกมาในรูปแบบของกราฟเพื่อตีความ รวมไปถึงโปรแกรมสำหรับสอนการเรียนรู้ อย่างเช่น โปรแกรมสอนความเข้าใจทางภาษาสำหรับเด็ก ก็เป็นภาพทางด้านกราฟฟิกต่าง ๆ ที่สวยงามที่เกี่ยวข้องกับการใช้คำ จะเห็นได้ว่าบทบาทของคอมพิวเตอร์กราฟฟิกได้เข้ามามีส่วนช่วยงานทางด้านการศึกษาในสาขาต่าง ๆ เป็นจำนวนมาก

จากที่กล่าวมาข้างต้น คอมพิวเตอร์กราฟฟิกได้เข้ามามีบทบาทอย่างมากในด้านต่าง ๆ ไม่ว่าจะเป็นงานทางด้าน ทิว, โฆษณา, อุตสาหกรรม, การศึกษา, การแพทย์, งานทางด้านก่อสร้างทำแบบจำลองต่าง ๆ และอื่น ๆ อีกมากโดยในรายงานเล่มนี้เน้นการใช้งานคอมพิวเตอร์กราฟฟิก 3 มิติ (Three Dimensional Computer Graphic) โดยเน้นสำหรับงานทางด้านสื่อนำเสนอ (Presentation), การทำโฆษณา (Advertising) รวมถึงการประยุกต์ใช้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประยุกต์ใช้สำหรับงานออกแบบทางด้านโครงสร้างอาคาร แต่การใช้งานทางด้านคอมพิวเตอร์กราฟฟิก 3 มิติ ไม่ง่ายเสียทีเดียว ผู้ใช้ต้องมีความรู้ ความเข้าใจในเรื่องต่าง ๆ ไม่ว่าจะเป็นการสร้างโครงร่าง (Wire Frame), การกำหนดคุณสมบัติของผิววัตถุ (Materials), การควบคุมสภาพแวดล้อมต่าง ๆ ที่อยู่รอบข้าง (Environments) ของวัตถุเช่น แสง, กล้อง และการจัดเตรียมอุปกรณ์ต่าง ๆ เพื่อนำมาใช้ในระบบคอมพิวเตอร์กราฟฟิก

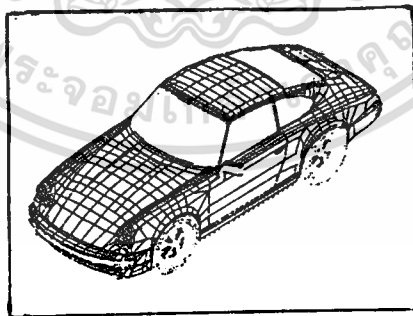
3.1 การสร้างงานด้วยระบบคอมพิวเตอร์กราฟฟิก 3 มิติ

ในส่วนนี้เน้นการปรับพื้นฐานของผู้อ่านที่ยังไม่เคยใช้งานหรือทำงานทางด้านนี้มาก่อนเลย โดยอธิบายถึงขั้นตอนต่าง ๆ ที่ใช้ในการสร้างงานด้วยระบบคอมพิวเตอร์กราฟฟิก 3 มิติตั้งแต่เริ่มต้น เพื่อให้ผู้อ่านเข้าใจขั้นตอนต่าง ๆ ที่ใช้ในการสร้างงาน โดยทั่ว ๆ ไปขั้นตอนที่ใช้ในการสร้างงานด้วยระบบคอมพิวเตอร์กราฟฟิก 3 มิติ แบ่งออกเป็น 6 วิธีคือ

- การสร้างโครงวัตถุที่เป็นลายเส้น (Model Object)
- การกำหนดคุณสมบัติต่าง ๆ ให้กับผิววัตถุ (Assign Materials)
- การจัดองค์ประกอบภายในฉาก
- การกำหนดการเคลื่อนไหวให้กับวัตถุ (Animation)
- การทำ Special Effect
- การ Render

สำหรับรายละเอียดในแต่ละวิธีมีดังต่อไปนี้

3.1.1 การสร้าง โครงวัตถุที่เป็นลายเส้น (Model Object)



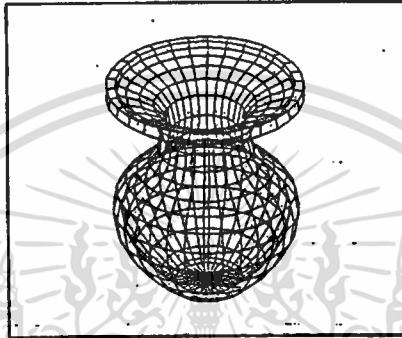
รูปที่ 3.1 ตัวอย่างโครงรูป 3 มิติ

เป็นจุดเริ่มต้นในการสร้างวัตถุ 3 มิติ ลักษณะวัตถุที่สร้างมีลักษณะเป็น โครงเส้น (Wireframe) เพื่อเป็นการบอกขนาดและลักษณะของรูปทรงของวัตถุการสร้างสรรค์สามารถเริ่มต้นสร้างจากโครงรูป 2 มิติก่อนแล้วค่อยดึงรูปด้วยวิธีการต่าง ๆ เพื่อให้เป็นวัตถุ 3 มิติ (Model Object) หรือสร้างวัตถุ 3 มิติจากรูปทรงพื้นฐาน เช่น วัตถุรูปทรงกลม, วัตถุรูปทรงสี่เหลี่ยม ฯลฯ แล้วมาทำการเปลี่ยนแปลงแก้ไขรูปทรงของวัตถุ ความซับซ้อนของวัตถุมีผลต่อการคำนวณของเครื่องคอมพิวเตอร์ ยิ่งวัตถุมีความซับซ้อนมากขึ้นเท่าไร

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ย่อมมากขึ้น รวมไปถึงประสิทธิภาพของเครื่องทั้งในแง่ของการคำนวณและจำนวนหน่วยความจำที่ใช้สำหรับการทำงานต้องมากขึ้นตามไปด้วย เพราะฉะนั้นการออกแบบและสร้างต้นแบบที่ดีสามารถลดเวลาในการสร้างและทำงานในขั้นตอนอื่น ๆ เป็นอย่างมาก

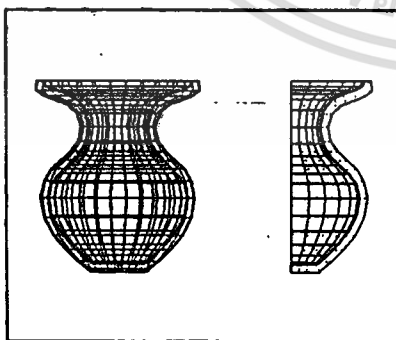
การสร้างวัตถุด้วยคอมพิวเตอร์มีอยู่หลายวิธีหลายเทคนิค ขึ้นอยู่กับลักษณะของวัตถุรวมถึงประสบการณ์ของผู้สร้าง และความสามารถของโปรแกรม เพื่อความเข้าใจในการสร้างวัตถุจะยกตัวอย่างวิธีที่ใช้ในการสร้างวัตถุรูปแจกันที่แสดงดังรูป



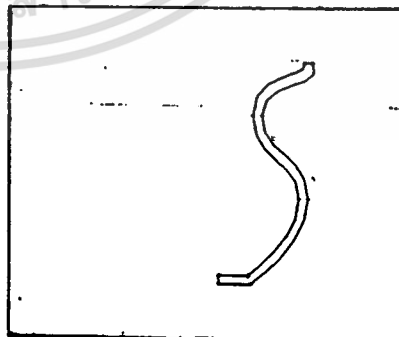
รูปที่ 3.2 หลักการและวิธีที่ใช้ในการสร้างแจกันดังนี้

• สร้างรูปตัดขวางของวัตถุ 2 มิติ

ในการสร้างวัตถุที่มีลักษณะสมมาตร เรามักจะสร้างโครงรูปตัดขวาง 2 มิติของวัตถุขึ้นมา แล้วนำมาหมุนในเส้นทาง (Path) ที่เป็นวงกลม เราก็จะได้วัตถุนั้น ในรูปข้างล่างจะแสดงลักษณะของวัตถุและภาพตัดขวางของวัตถุ พร้อมกับโครงรูป 2 มิติที่ทำการสร้าง



(ก)



(ข)

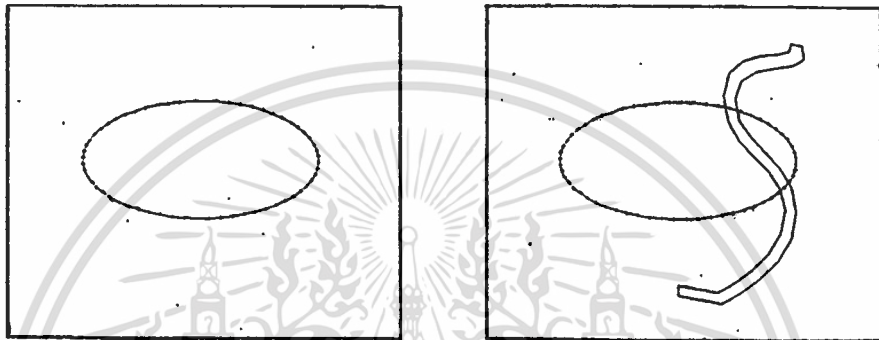
รูปที่ 3.3 (ก) ลักษณะของวัตถุและภาพตัดขวางของวัตถุ

(ข) ลักษณะของโครงรูป 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สร้างเส้นทางที่ใช้ในการหมุน และจัดวางวัตถุในเส้นทางที่กำหนด

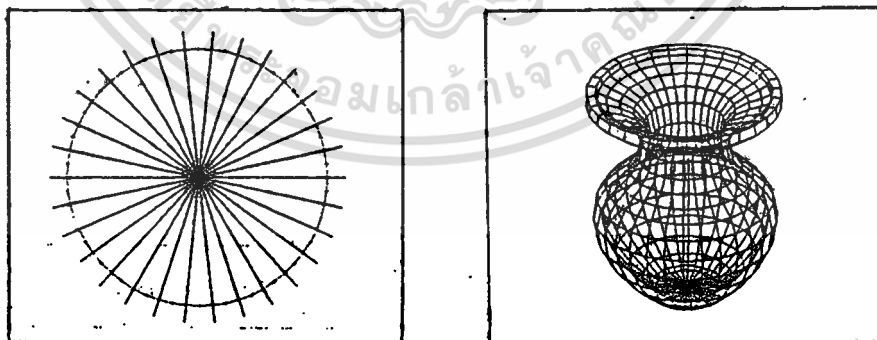
เมื่อเราสร้างโครงรูป 2 มิติ ขึ้นตอนต่อไปก็คือการดึงรูปจาก 2 มิติให้เป็น 3 มิติ ในการดึงรูปเราต้องสร้างเส้นทาง (Path) สำหรับกำหนดทิศทางในการดึงรูป ซึ่งเส้นทางอาจดึงขึ้นในแนวเส้นตรงหรือเส้นโค้งขึ้นอยู่กับลักษณะของวัตถุที่ต้องการสร้าง สำหรับเจกกันเราสร้างเส้นทางเป็นวงกลม และนำโครงรูปมาจัดวางในเส้นทางที่กำหนดขึ้น ในการจัดวางโครงรูปในเส้นทางก็มีผลต่อรูปทรงของวัตถุ รายละเอียดในส่วนนี้จะกล่าวถึงในบทของ 3D Lofting



รูปที่ 3.4 แสดงการกำหนดทิศทางสร้างวัตถุและนำโครงรูปมาจัดวางในเส้นทางที่กำหนด

- ทำการหมุนวัตถุ

ในขั้นตอนนี้เป็นการสร้างวัตถุขึ้นมา โดยการหมุนโครงรูปในเส้นทางที่กำหนด สำหรับการหมุนโครงรูป โครงรูปต้องหมุนเข้าสู่จุดศูนย์กลางของเส้นทางดังแสดงในภาพข้างล่าง ผู้ใช้ก็จะได้วัตถุ 3 มิติรูปแจกันตามที่ต้องการ



รูปที่ 3.5 แสดงเส้นทางการหมุนเข้าสู่จุดศูนย์กลาง

จากวิธีที่กล่าวมาข้างต้น เป็นวิธีหนึ่งที่นิยมใช้ในการสร้างวัตถุ ซึ่งวิธีหรือเทคนิคที่ใช้ในการสร้างวัตถุ ผู้ใช้สามารถเรียนรู้จากตัวอย่างของงานที่แถมมาให้กับตัวโปรแกรมนั้น ๆ

สำหรับงานที่เกี่ยวข้องกับงานทางด้านคอมพิวเตอร์กราฟฟิกมักเกี่ยวข้องกับเวลา ถ้าต้องการทำงานดีมีรายละเอียดมาก ๆ เวลาในการทำงานต้องมากตามไปด้วย ดังนั้น ประสิทธิภาพของเครื่อง, ด้านการคำนวณ, ไม่ว่าจะเร็วแค่ไหน, ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพของโปรแกรม และประสบการณ์ของผู้ใช้มีผลมาก เพราะสามารถที่จะทำให้งานออกมาดีได้โดยไม่ต้องใช้เวลา

3.1.2 การกำหนดคุณสมบัติต่าง ๆ ให้กับผิววัตถุ (Assign Materials)

หลังจากได้วัตถุรูปทรง 3 มิติ (Model Object) ซึ่งเป็นโครงลายเส้น ก็ถึงขั้นตอนที่ผู้สร้างต้องกำหนดคุณสมบัติ (Material) ต่าง ๆ ให้กับผิววัตถุ เช่น สี, ความเงา, ความโปร่งใส, ความขรุขระของผิววัตถุ หรือลักษณะลายบนผิววัตถุ ทำให้ผู้สร้างสามารถกำหนดรูปแบบวัตถุได้จำนวนมาก เช่น ผู้ใช้สามารถสร้างคันแอปเปิ้ล ให้ออกลูกแอปเปิ้ลมีผิวเป็นลายอิฐได้ ซึ่งในความเป็นจริงคงไม่มีเหตุการณ์เช่นนี้เกิดขึ้น ในโปรแกรมต่าง ๆ มีวิธีการกำหนดคุณสมบัติของวัตถุแตกต่างกันออกไป ขึ้นอยู่กับความสามารถของโปรแกรมนั้น ๆ หลังจากที่ได้อุปกรณ์ของวัตถุ ก็ถึงขั้นตอนการนำเอาคุณสมบัติของวัตถุไปใส่บนผิวของวัตถุ เพื่อให้เกิดภาพตามที่ต้องการ



รูปที่ 3.6 ลักษณะของวัตถุรูปแฉกกัน เมื่อนำเอาคุณสมบัติแบบต่าง ๆ มาใส่บนผิววัตถุ

3.1.3 การจัดองค์ประกอบภายในฉาก

ในส่วนนี้ผู้ใช้ต้องสมมติตัวเองเป็นผู้กำกับ คือผู้ใช้ต้องทำสิ่งต่าง ๆ ดังต่อไปนี้

1. จัดตำแหน่งต่าง ๆ ของวัตถุ เนื่องจากในการทำงานชิ้นหนึ่ง ๆ อาจจะมียุทธวิธีอยู่จำนวนมาก ผู้ใช้ต้องทำการจัดตำแหน่งของวัตถุตามที่ต้องการ
2. การสร้างแสง, กำหนดสีของแสง, กำหนดความสว่างของแสง, รวมไปถึงการเลือกประเภทของแสง เช่น แสงแบบเปล่งออกรอบตัวของมันเอง (Omni Light) หรือแสงแบบ Spotlight
3. กำหนดตำแหน่งของแสง ซึ่งจะมีผลกับเฉดสีและเงาของวัตถุ
4. การสร้างกล้อง และกำหนดขนาดของเลนส์ที่ต้องการใช้ ขนาดของเลนส์ที่กำหนดจะมีผลต่อลักษณะของภาพที่เกิดขึ้น
5. การกำหนดตำแหน่งของกล้อง, ระยะและมุมมองของกล้อง เพื่อให้ได้รูปตามที่ต้องการ

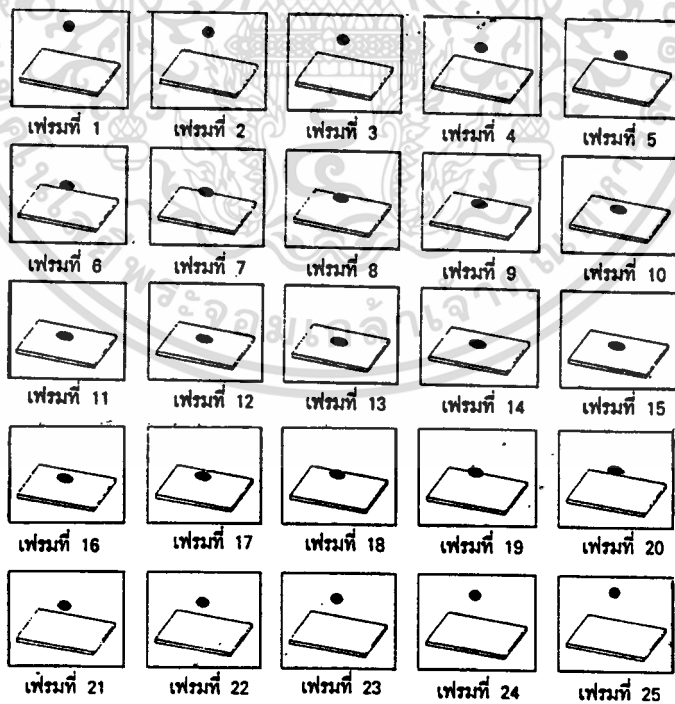
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 การกำหนดการเคลื่อนไหวให้กับวัตถุ (Animation)

หลังจากที่ผู้ใช้จัดองค์ประกอบต่าง ๆ ภายในฉากเรียบร้อยแล้ว ก็ถึงขั้นตอนในการทำการเคลื่อนไหว โปรแกรมที่ใช้สร้างภาพ 3 มิติ ส่วนใหญ่จะมีส่วนสำหรับการเคลื่อนไหวให้กับวัตถุ สำหรับการเคลื่อนไหวที่สร้างด้วยคอมพิวเตอร์กำหนดอยู่ในหน่วยของเฟรม (Frame) ซึ่งทำให้สามารถควบคุมเกี่ยวกับเวลาได้ โดย

1. การเคลื่อนไหวที่ใช้ในระบบทีวีบ้านเราในระบบ PAL กำหนดจำนวนเฟรม 25 เฟรม เท่ากับ 1 วินาที
2. การเคลื่อนไหวที่ใช้ในระบบทีวีในระบบ NTSC กำหนดจำนวนเฟรม 30 เฟรม เท่ากับ 1 วินาที

ถ้าผู้ใช้รับงานทางด้านการทำโฆษณาจำนวน 10 วินาที ผู้ใช้ก็สามารถกำหนดจำนวนเฟรมที่ต้องทำงานได้คือ 250 เฟรม และการควบคุมการเคลื่อนที่ของวัตถุที่เกิดขึ้นในระหว่างเฟรม ผู้ใช้ก็สามารถควบคุมเวลาได้ ในการทำงานบางครั้งผู้ใช้สามารถเปลี่ยนแปลงค่านี้น้อยลงได้ แต่ลักษณะของภาพเมื่อนำมาแสดงแบบต่อเนื่องจะกระตุก ในรูปข้างล่างจะแสดงการเคลื่อนไหวของลูกบอลที่เกิดขึ้นในแต่ละเฟรมเป็นจำนวน 1 วินาที (25 เฟรม)



รูปที่ 3.7 แสดงการเคลื่อนที่ของลูกบอลในแต่ละเฟรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญูญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะการเคลื่อนไหวของลูกบอลที่เกิดขึ้นเราไม่ได้กำหนดตำแหน่งของลูกบอลลงไปทุกเฟรมสำหรับตำแหน่งที่กำหนดของลูกบอลมีเพียง 3 เฟรม คือ

เฟรมที่ 1 กำหนดจุดสูงสุดของลูกบอล

เฟรมที่ 13 กำหนดจุดต่ำสุดของลูกบอล

เฟรมที่ 25 กำหนดจุดสูงสุดของการสะท้อนกลับของลูกบอล (ในที่นี้ลูกบอลกลับไปอยู่ในตำแหน่งเดิม)

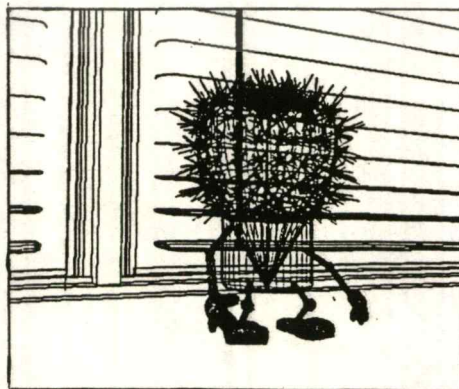
ส่วนเฟรมที่เหลือคอมพิวเตอร์จะเป็นผู้ที่กำหนดระยะของลูกบอลให้ ทำให้การกำหนดการเคลื่อนไหวของวัตถุง่ายขึ้น เพียงแต่กำหนดเฟรมเริ่มต้น เฟรมสุดท้ายของการเคลื่อนไหว ในระหว่างนั้นเครื่องจะทำให้ ถ้าในส่วนที่เครื่องทำให้ผู้ใช้ไม่พอใจก็สามารถทำการแก้ไขได้

3.1.5 การทำ Special Effect

ในส่วนนี้เป็นการใส่ลูกเล่นให้กับชิ้นงาน เช่น การใส่แสงเรือง, การทำให้วัตถุแตกกระจาย ออกเป็นอนุภาคแล้วเคลื่อนไหวไปในทิศทางต่าง ๆ ซึ่งในส่วนนี้มักจะใช้โปรแกรมเสริมจากภายนอก และผู้ใช้ต้องจัดหาซื้อเพิ่มเติมเข้ามา สำหรับโปรแกรม Autodesk 3D Studio ตั้งแต่ Release 3 ก็จะมีโปรแกรมเสริมที่ช่วยในการทำเทคนิคต่าง ๆ ซึ่งผู้ใช้สามารถหาซื้อเพิ่มเติมได้

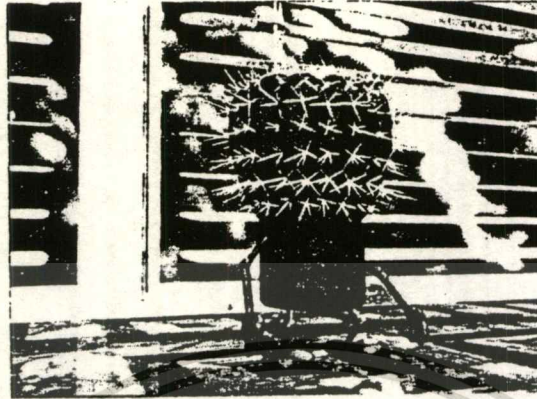
3.1.6 การ Render

เป็นขั้นตอนที่ผู้สร้างสั่งให้เครื่องนำข้อมูลทางด้านโครงเส้น 3 มิติของวัตถุ มาใส่พื้นผิวตามที่กำหนดไว้ แล้วคำนวณสีของวัตถุตามสภาพแวดล้อมเพื่อให้เกิดเป็นภาพ ถ้าเปรียบเทียบกับกรวาดภาพแล้ว การทำ Model Object เหมือนกับการร่างภาพ และการ Render คือการลงสีลงเงา เพื่อให้ภาพดูมีชีวิตชีวาเหมือนจริง ซึ่งเทคนิคหรือวิธีการ Render ที่ใช้ในเครื่องคอมพิวเตอร์เป็นเรื่องที่ซับซ้อน เพราะเกี่ยวข้องกับปริมาณทางคณิตศาสตร์เป็นส่วนใหญ่ เนื่องจากเครื่องต้องคำนวณลักษณะผิวที่ทำการ Mapping, มุมสะท้อนของแสง และการหักเหของแสงที่เกิดขึ้นว่ามีผลกับวัตถุต่าง ๆ อย่างไรบ้าง ดังนั้นผู้ที่ศึกษาเบื้องต้นควรศึกษาในโหมดต่าง ๆ ของการ Render รวมถึงผลของการ Render ในแต่ละโหมด



รูปที่ 3.8 ลักษณะของวัตถุที่เป็นลายเส้นและทำการจัดองค์ประกอบในฉากเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับว่าผิดเงื่อนไขใบลิขสิทธิ์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ลักษณะของภาพที่เกิดขึ้นหลังจากการทำ Render

3.2 วิธีการเข้าสู่โปรแกรม

สำหรับการใช้งานโปรแกรม Autodesk 3D Studio อุปกรณ์ที่สำคัญที่ขาดไม่ได้คือ อุปกรณ์สำหรับซี ซึ่งอาจจะเป็นเมาส์หรือแท็บเล็ตก็ได้ ผู้ใช้ต้องตรวจสอบว่าได้มีการติดตั้งอุปกรณ์นี้แล้วหรือไม่ และมีการติดตั้งไดรเวอร์ของอุปกรณ์เหล่านี้แล้วหรือยัง ถ้ายังผู้ใช้ควรติดตั้งอุปกรณ์และไดรเวอร์ให้เรียบร้อยก่อน ไมเช่นนั้นเมื่อผู้ใช้เรียกใช้โปรแกรม ตัวซีในโปรแกรมผู้ใช้จะไม่สามารถเคลื่อนที่ได้ทำให้ไม่สามารถทำงานได้ ซึ่งผู้ใช้ต้องออกจากโปรแกรมและติดตั้งให้เรียบร้อยก่อนการใช้งาน

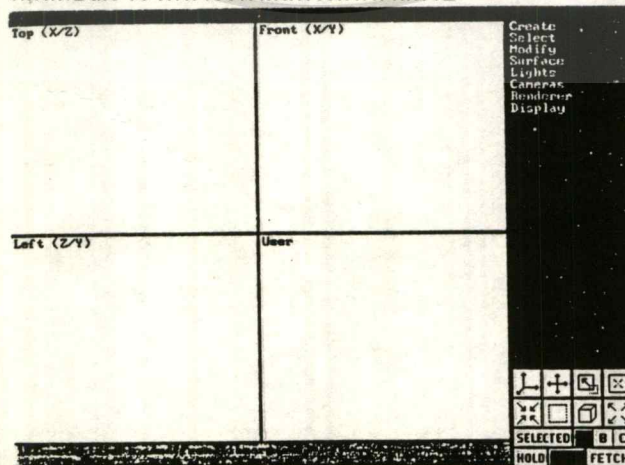
ขั้นตอนการเรียกใช้งานของโปรแกรม ให้ผู้ใช้เปลี่ยนไปยังไดเรกทอรีที่เก็บโปรแกรม Autodesk 3D studio ด้วยคำสั่ง

C:\> CD 3DS4

เรียกการทำงานของโปรแกรมด้วยการพิมพ์คำสั่ง

C:\3DS4> 3DS

หลังจากนั้นจะเข้าสู่การทำงานของโปรแกรม Autodesk 3D Studio โดยที่หน้าจอแสดงส่วนการทำงานหลัก 3D Editor ที่มีลักษณะของหน้าจอ ดังแสดงในหน้าถัดไป



รูปที่ 3.10 ลักษณะหน้าจอของส่วนการทำงานแรกเมื่อเข้าสู่โปรแกรม Autodesk 3D Studio

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อที่หน้าจอแสดงส่วนการทำงานหลัก 3D Editor นั้นหมายถึงว่า โปรแกรมพร้อมที่จะทำงานผู้ใช้สามารถเริ่มต้นการทำงานจากส่วนการทำงานหลัก 3D Editor นี้ได้เลย หรือเปลี่ยนไปทำงานในส่วนการทำงานหลักอื่น ๆ ได้

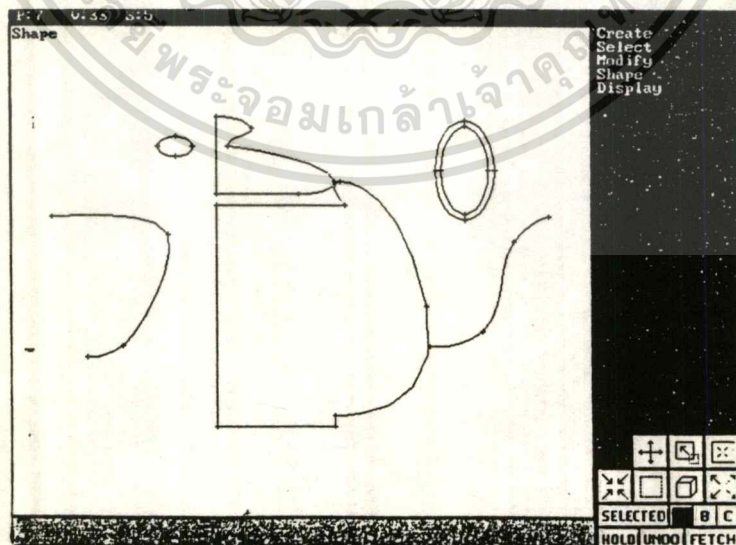
3.8 เข้าใจส่วนการทำงานหลักภายในโปรแกรม

ส่วนการทำงานหลักของโปรแกรมประกอบไปด้วยส่วนต่าง ๆ 5 ส่วนคือ

- 2D SHAPER
- 3D LOFTER
- 3D EDITOR
- KEYFRAMER
- MATERIALS EDITOR

3.3.1 2D SHAPER

ในการสร้างวัตถุ 3 มิติที่ซับซ้อนในบางครั้งผู้ใช้ต้องเริ่มต้นตั้งแต่การสร้างโครงเส้น 2 มิติ เพื่อนำไปใช้เป็นส่วนประกอบชิ้นพื้นฐานสำหรับการขึ้นรูป โปรแกรม Autodesk 3D Studio ได้จัดชุดคำสั่งสำหรับการสร้างโครงสร้างเส้น 2 มิติไว้ในส่วนของ 2D Shaper เพื่อเป็นส่วนสำหรับเตรียมโครงรูป 2 มิติสำหรับนำไปใช้ในส่วนการทำงานอื่น ๆ ผู้ใช้สามารถเข้าสู่ส่วนการทำงาน 2D Shaper ได้โดยการกดฟังก์ชันคีย์ F1 หรือเรียกเมนู "Program/2D Shaper" สำหรับรูปข้างล่างแสดงรายละเอียดภายในหน้าจอของส่วนการทำงานหลัก 2D Shaper โดยมีส่วนของโครงเส้น 2 มิติที่สร้างไว้สำหรับนำไปขึ้นรูปเป็นกาน้ำชา

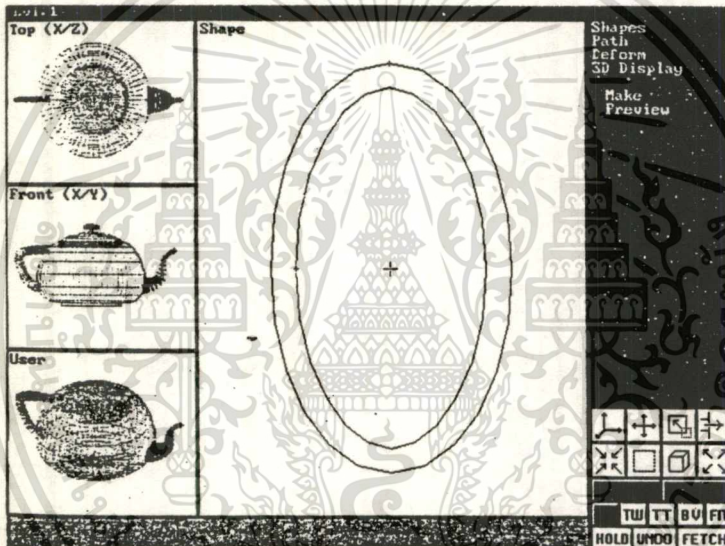


รูปที่ 3.11 ลักษณะหน้าจอของส่วนการทำงานหลัก 2D SHAPER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 3D LOFTER

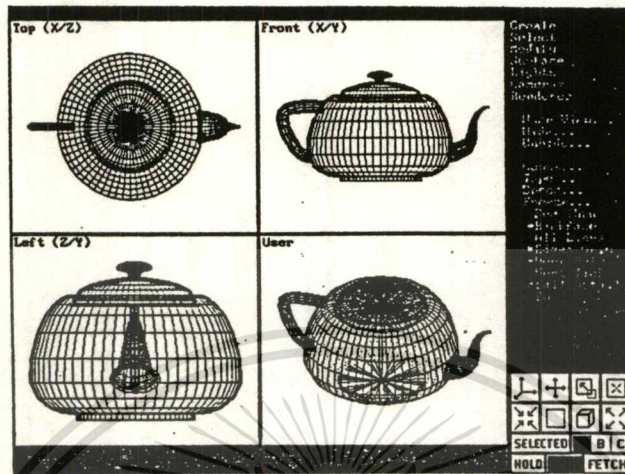
โปรแกรม Autodesk 3D Studio มีส่วนการทำงานสำหรับการดึงโครงรูปจาก 2 มิติเป็น 3 มิติ ซึ่งส่วนนี้จะมีชุดคำสั่งต่าง ๆ เพื่อใช้ในการดึงโครงรูปที่สร้างจาก 2D Shaper หรือนำโครงเส้นที่สร้างจากโปรแกรมอื่น เช่น โปรแกรม AutoCAD มาใช้งาน โดยโปรแกรม Autodesk 3D Studio ได้จัดการทำงานในส่วนนี้ไว้ในส่วนการทำงานหลัก 3D Loftter ผู้ใช้สามารถเข้าสู่ส่วนการทำงาน 3D Loftter ได้โดยการกดฟังก์ชันคีย์ F2 หรือเรียกจากเมนู “Program/3D Loftter” สำหรับรูปข้างล่างแสดงรายละเอียดภายในหน้าจอของส่วนการทำงานหลัก 3D Loftter โดยแสดงส่วนโครงสร้างเส้น 2 มิติจาก 2D Shaper เมื่อนำมาดึงให้เป็นวัตถุ 3 มิติ



รูปที่ 3.12 ลักษณะหน้าจอของส่วนการทำงานหลัก 3D LOFTER

3.3.3 3D EDITOR

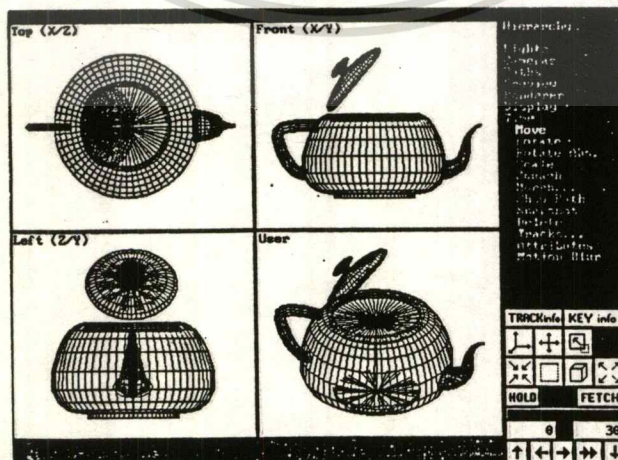
เป็นส่วนการทำงานที่เป็นหัวใจของโปรแกรม Autodesk 3D Studio ในส่วนนี้ผู้ใช้สามารถทำงานเกี่ยวกับการสร้างวัตถุพื้นฐานโดยใช้ชุดคำสั่งภายใน 3D Editor, การแก้ไขและตัดแปลงรูปทรงของวัตถุ, การรวมวัตถุเข้าด้วยกันด้วยวิธีทางคณิตศาสตร์ (Boolean), การจัดตำแหน่งต่าง ๆ ของวัตถุ, การสร้างแสงและกำหนดตำแหน่งของแสง, การสร้างกล้องและกำหนดตำแหน่งของกล้อง รวมไปถึงการกำหนดคุณสมบัติต่าง ๆ ให้กับผิวของวัตถุ (Assign Material) ผู้ใช้สามารถเข้าสู่ส่วนการทำงาน 3D Editor ได้โดยการกดฟังก์ชันคีย์ F3 หรือเรียกจากเมนู “Program/3D Editor” สำหรับรูปข้างล่างแสดงรายละเอียดภายในหน้าจอของส่วนการทำงานหลัก 3D Editor โดยแสดงวัตถุ 3 มิติรูปกาน้ำชาที่เริ่มต้นสร้างจาก 2D Shaper แล้วนำมาดึงทีละส่วนใน 3D Loftter และนำแต่ละส่วนมาประกอบเป็นรูปทรงใน 3D Editor



รูปที่ 3.13 ลักษณะหน้าจอของส่วนการทำงานหลัก 3D EDITOR

3.3.4 KEYFRAMER

โปรแกรม Autodesk 3D Studio ได้จัดชุดของคำสั่งสำหรับการเคลื่อนไหวให้กับวัตถุ (Animation) ในชุดคำสั่งเหล่านี้ผู้ใช้สามารถทำงานเกี่ยวกับการควบคุมการเคลื่อนไหวของวัตถุหรือกลุ่มของวัตถุที่มีความสัมพันธ์กัน, การควบคุมการเคลื่อนไหวของแสงและกล้อง, การเปลี่ยนขนาดของวัตถุ (Scale), การเปลี่ยนรูปทรงของวัตถุจากวัตถุหนึ่งไปยังอีกวัตถุหนึ่ง (Morph), การควบคุมการซ่อน (Hide) และแสดงวัตถุ (Unhide), การซ้อนภาพด้วยเทคนิคของ Video Post รวมไปถึงการทำ Motion Blur ให้กับวัตถุที่ทำการเคลื่อนไหว กลุ่มของชุดคำสั่งต่างๆ เหล่านี้อยู่ในส่วนการทำงานหลัก Keyframer ผู้ใช้สามารถเข้าสู่ส่วนการทำงาน Keyframer ได้โดยการกดฟังก์ชันคีย์ F4 หรือเรียกจากเมนู "Program/Keyframer" สำหรับรูปข้างล่าง แสดงรายละเอียดภายในหน้าจอของส่วนการทำงานหลัก Keyframer โดยแสดงวัตถุ 3 มิติรูปกาน้ำชาที่ทำการเคลื่อนไหวโดยการยกฝากาน้ำชาขึ้น

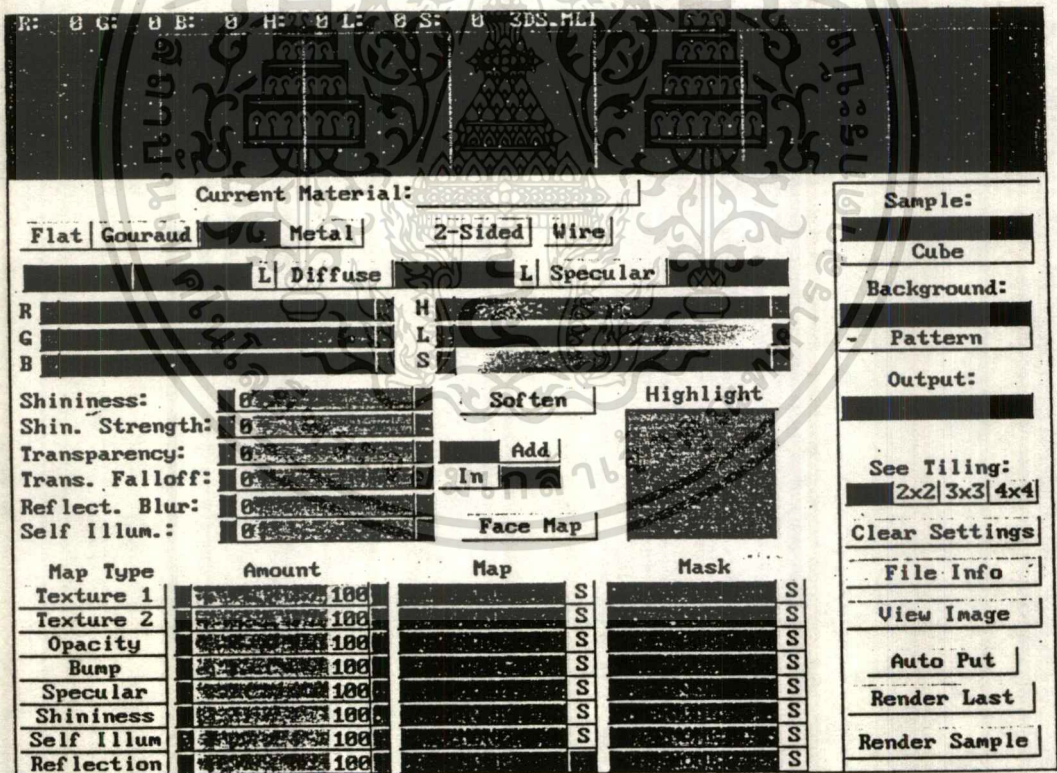


รูปที่ 3.14 ลักษณะหน้าจอของส่วนการทำงานหลัก Keyframer

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 MATERIALS EDITOR

สำหรับวัตถุที่สร้างขึ้นจาก 3D Loft (โดยการนำโครงรูป 2 มิติ จาก 2D Shaper มาคึง) หรือวัตถุที่สร้างจาก 3D Editor มีลักษณะเป็นโครงร่างทางลายเส้น ผู้ใช้ต้องทำการสร้างลักษณะของพื้นผิวเพื่อนำไปกำหนดให้เป็นผิวของโครงร่างทางลายเส้นของวัตถุนั้น ๆ ในส่วนที่ใช้สร้างคุณสมบัติของพื้นผิวของวัตถุ โปรแกรม Autodesk 3D Studio ได้จัดกลุ่มคำสั่งและกลุ่มทางเลือกต่าง ๆ ไว้ในส่วน ของ Materials Editor เพื่อให้ผู้ใช้กำหนดลักษณะพื้นผิวของวัตถุ เพื่อเป็นส่วนสร้างคุณสมบัติของพื้นผิวให้กับวัตถุ เพื่อนำไปกำหนดให้กับวัตถุโดยใช้ชุดคำสั่งในส่วน ของ 3D Editor ผู้ใช้สามารถเข้าสู่ส่วนการทำงาน Materials Editor ได้โดยการกดฟังก์ชันคีย์ F5 หรือเรียกจากเมนู "Program/Materials" สำหรับรูปที่แสดงข้างล่างนี้ได้แสดงรายละเอียดภายในหน้าจอของส่วนการทำงานหลัก Materials Editor



รูปที่ 3.15 ลักษณะหน้าจอของส่วนการทำงาน Materials Editor

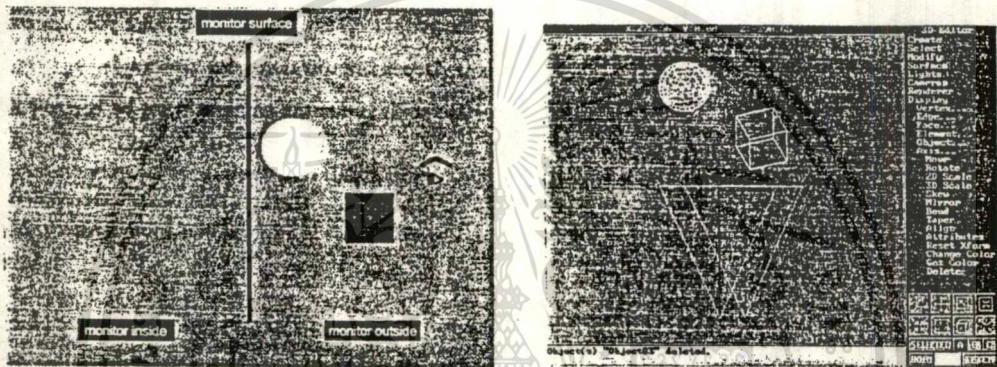
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาติให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



3.4 หลักการสร้างภาพจาก 3D Studio

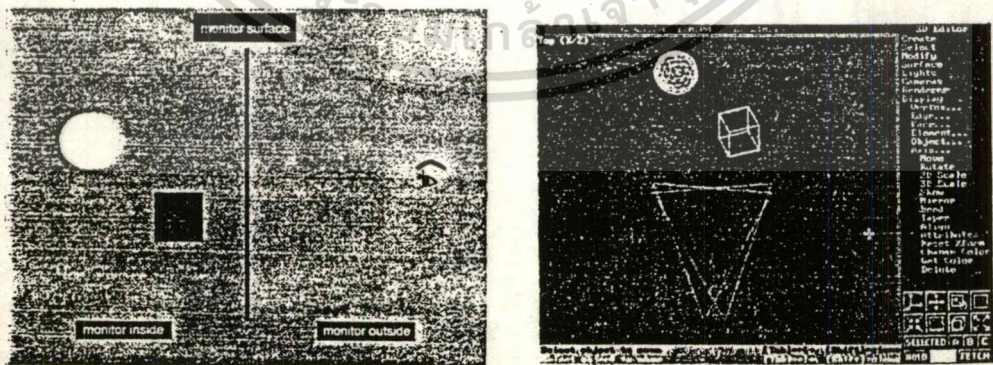
ในการสร้างภาพบนโปรแกรม Autodesk 3D Studio เป็นการกำหนดมุมและตำแหน่งของกล้อง โดยจะใช้กล้อง 2 กล้อง จับภาพทางด้านซ้ายกับทางด้านขวา แล้วจะนำภาพ 2 ภาพนี้ไปแสดงต่อไป สำหรับตำแหน่งควรอยู่ห่างกัน 6-7 เซนติเมตร เมื่อเทียบกับส่วนต่าง ๆ ซึ่งถ้าให้ความกว้างระดับนี้จะทำให้ขนาดของวัตถุต่าง ๆ มีขนาดใกล้เคียงของจริงมาก สำหรับมุมของกล้องจะกำหนดตำแหน่งความลึกของวัตถุ โดยกำหนดดังนี้

1. การกำหนดทำให้ดูแล้วรู้สึกว่วัตถุออกมาจากจอ โดยการกำหนดให้ กล้องวางตัวขนานกัน หรือ จุดตัดของแนวในการวางกล้องทั้งสองอยู่ด้านหลังวัตถุ



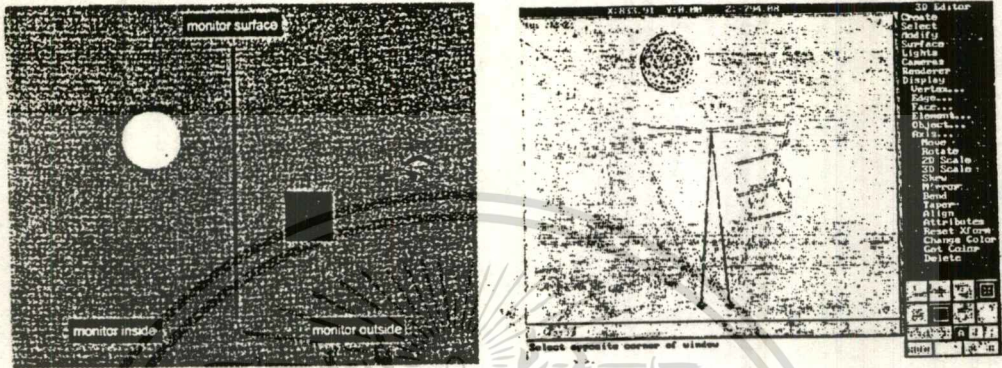
รูปที่ 3.16 การกำหนดทำให้ดูแล้ววัตถุออกมาจากจอ

2. การกำหนดทำให้ดูแล้วรู้สึกว่วัตถุลึกลงไปในจอภาพ ทำได้โดยการวางจุดตัดแนวในการวางกล้องทั้งสองให้อยู่ด้านหน้าวัตถุนั้น ถ้าวางก่อนหน้ามากจะทำให้ภาพดูแล้วรู้สึกว่วัตถุจะลึกลงไปในจอมาก



รูปที่ 3.17 การกำหนดทำให้ดูแล้ววัตถุลึกลงไปในจอภาพ

3. การกำหนดให้ภาพมีทั้งนูนและลึกในภาพเดียวกัน โดยการกำหนดจุดตัดแนววงกลองให้มีทั้งวัตถุอยู่ก่อนหน้าจุดตัดและหลังจุดตัด วัตถุที่อยู่หน้าจุดตัดจะทำให้ดูเหมือนวัตถุนูนออกมา ส่วนวัตถุที่อยู่หลังจุดตัดจะทำให้วัตถุนั้นลึกลงไปในจอภาพ



รูปที่ 3.18 การกำหนดให้ภาพมีทั้งนูนและลึกในภาพเดียวกัน

บทที่ 4

พื้นฐานการแสดงผล

การ์ด EGA ได้รับการพัฒนาโดยบริษัทไอบีเอ็ม สำหรับใช้ร่วมกับจอ ECD โดยให้ความละเอียดของการแสดงผลในแนวนอน 640 จุด และแนวตั้ง 350 จุด และสามารถแสดงได้พร้อมกัน 16 สี การ์ด EGA สามารถนำไปใช้กับจอแสดงผลอื่น ๆ บนเครื่องไอบีเอ็ม คอมแพทิเบิล (Compatible : เข้ากันได้ , ทำงานแทนกันได้) ทั่วไปส่งผลให้ผู้พัฒนาซอฟต์แวร์ส่วนใหญ่หันมาพัฒนาโปรแกรมให้ใช้งานร่วมกับการ์ด EGA คอมแพทิเบิลได้

การ์ด VGA ได้รับการพัฒนาขึ้นมาโดยบริษัทไอบีเอ็ม เช่นกัน โดยใช้เป็นอุปกรณ์มาตรฐานสำหรับเครื่องตระกูล PS/2 ซึ่งมีความละเอียดการแสดงผลในแนวนอน 640 จุด และแนวตั้ง 480 จุด และสามารถแสดงสีได้มากกว่าการ์ด EGA (โดยแสดงได้ 256 สีพร้อมกันแต่ความละเอียดการแสดงผลจะลดลง) การ์ด VGA ที่พัฒนาขึ้นมาไม่สามารถใช้งานกับจอทั่วๆ ไปได้ จึงได้พัฒนาจอสี และจอโมโนโครมสำหรับ VGA ขึ้นมารองรับ นอกจากนี้ยังได้มี การพัฒนาซอฟต์แวร์สำหรับใช้งานร่วมกับ VGA

การ์ด EGA และ VGA ไม่สามารถประมวลผลโดยตัวเองได้ ซึ่งเหมือนกับคุณสมบัติทั่วไปของอะแดปเตอร์ (Adapter) อื่น แต่จะใช้ตัวประมวลผลระบบในการจัดการเรื่องการแสดงผล โดยจะเขียนข้อมูลที่จะแสดงลงในหน่วยความจำส่วนแสดงผล (bit-mapped display memory) ผู้ใช้คอมพิวเตอร์มักจะเลือกข้อกำหนดและขอบเขตในการใช้งานส่วนแสดงผลตามโปรแกรมที่ใช้ (เช่น Microsoft Windows , Digital Research , GEM หรือ GKS) หรืออาจเขียนโปรแกรมสำหรับควบคุมส่วนนี้เอง

ตารางที่ 4.1 จอแสดงผลที่คอมแพทิเบิลกับไอบีเอ็มพีซี

จอแสดงผล	การ์ดแสดงผล	สี	ความละเอียดของตัวอักษร	ความละเอียดของกราฟิกส์	อัตรากรกวาดพิกเซล
จอโมโนโครม	เซอร์ทวิส EGA	2	MDA 80*25	640*350 720*350 720*348	แนวตั้ง -50 Hz แนวนอน -15.8 KHz
จอสีมาตรฐาน	CGA EGA	16	40*50 80*25	320*200 640*200	แนวตั้ง -60 Hz แนวนอน -15.8 KHz
จอ ECD	CGA EGA	16 จาก 64	40*25 80*25	320*200 640*200 640*350	แนวตั้ง -60 Hz แนวนอน -15.8 KHz หรือ 21.8 KHz
จอสีจอลดมิติดิจิทัล	CGA EGA	16 จาก 64	40*25	320*200 640*200 640*350	
จออะนะล็อกมิติดิจิทัล	VGA	256 จาก 256K	80*25	640*480 800*600	เปลี่ยนแปลงได้

ตารางที่ 4.1 (ต่อ) จอแสดงผลที่คอมแพททิเบิลกับไอพีเอ็มพีซี

จอแสดงผล	การ์ดแสดงผล	สี	ความละเอียด ของตัวอักษร	ความละเอียด ของกราฟิก	อัตราการ กวาดภาพ
จอสี VGA	VGA	256 จาก 256K	40*25	320*400	แนวตั้ง -70 Hz
จอโมโนโครม	VGA		80*25	640*400	แนวนอน -31.5 KHz
				320*350	
				640*350	
				720*350	
				720*400	
				640*480	

4.1 การเลือกส่วนแสดงผล

ตารางที่ 4.1 ได้รวบรวมจอชนิดต่าง ๆ ที่สามารถใช้งานร่วมกับการ์ด EGA และ VGA ได้ การแบ่งประเภทของจอภาพสามารถแบ่งได้ตามชนิดของการอินเตอร์เฟซ (Ingerface : การติดต่อระหว่างส่วนประมวลผลและส่วนอื่น) ที่ใช้

ประเภทของการอินเตอร์เฟซโดยทั่วไป

- **จอดิจิทัล (Digital (TTL) Displays)** โดยปกติประกอบด้วยเส้นอินพุตสีของจอ 6 เส้น เมื่อเส้นใด ON สัญญาณสีของเส้นนั้นก็ปรากฏบนจอภาพ ถ้าจอมีเส้นอินพุต n เส้น จำนวนสีที่จอดิจิทัลสามารถแสดงได้จะเท่ากับ 2^n สี จอส่วนใหญ่ที่ใช้ในคอมพิวเตอร์ส่วนตัวจะเป็นแบบดิจิทัล และการ์ด EGA ก็ต้องใช้กับจอดิจิทัล
- **จอแบบผสม (Composite Displays)** จะมีเส้นอินพุตวิดีโอเพียงเส้นเดียว ซึ่งอาจจะแสดงแบบโมโนโครมหรือแบบสีก็ได้ ข้อมูลสำหรับจอแบบผสมนี้จะผ่านการเข้ารหัสรวมกันมาบนเส้นอินพุตเดียวตามมาตรฐาน NTSC (National Television System Committee) จอแบบนี้ให้ความละเอียดต่ำและราคาค่อนข้างถูก CGA เป็นการ์ดแสดงผลของไอพีเอ็มพีซีชนิดเดียวที่สนับสนุนการใช้งานจอชนิดนี้ (CGA ยังสามารถใช้ร่วมกับจอดิจิทัลได้) จอแสดงผลแบบนี้ใช้โดยทั่วไปในอุตสาหกรรมโทรทัศน์ แต่ไม่นิยมใช้ในวงการคอมพิวเตอร์
- **จออะนาล็อก RGB (Analog RGB Displays)** ประกอบด้วยอินพุตสีแบบอะนาล็อก 3 เส้น คือ สีแดง , สีเขียว และสีน้ำเงิน ระดับแรงดันในแต่ละเส้นอินพุตจะเป็นตัวกำหนดระดับสีที่ปรากฏบนจอภาพ ตามทฤษฎีจอแบบนี้แสดงสีได้ไม่จำกัด แต่ในความเป็นจริงแล้วจำนวนสีที่แสดงได้ขึ้นอยู่กับความสามารถของอะแดปเตอร์

การ์ด VGA ใช้ได้กับจออะนาล็อกเท่านั้น โดยใช้ได้ทั้งจออะนาล็อกแบบโมโนโครมและแบบสีที่ออกแบบไว้ใช้สำหรับการ์ด VGA โดยเฉพาะ ซึ่งมีความยืดหยุ่นในการใช้งานมากกว่าจอดิจิทัล โดยจอเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นาล็อก ทั้งสองแบบสามารถทำงานกับการ์ด VGA ได้ทุกโหมด โดยถ้าใช้โหมดสีบนจอโมโนโครม การแสดงผลภาพจะออกมาในแบบระดับของสีเทา

การพิจารณาว่าจะใช้จอแสดงผลชนิดใด ขึ้นอยู่กับซอฟต์แวร์ที่จะนำมาใช้กับระบบนั้น จอแสดงผลที่ใช้กับเครื่องไอบีเอ็มคอมพิวเตอร์ชนิดในปัจจุบันแสดงดังตารางที่ 4.1

จอโมโนโครม

จอโมโนโครม (Monochrome displays : MD) และการ์ด MDA (Monochrome Display Adapter) เป็นอุปกรณ์แสดงผลสำหรับเครื่องไอบีเอ็มพีซีรุ่นแรก โดยการ์ด MDA ไม่สามารถแสดงกราฟฟิกได้ แต่ก็ยังคงใช้กันอย่างแพร่หลายในปัจจุบันสำหรับแสดงข้อความ (text) การ์ด MDA ให้ความละเอียดถึง 720 x 350 จุด ซึ่งสูงกว่า EGA ซึ่งมีความละเอียดเพียง 640 x 350 จุด นอกจากนี้ยังมีราคาถูกกว่าและเหมาะกับการแสดงตัวอักษรมากกว่าจอสีบางชนิด

หลังจากที่บริษัทไอบีเอ็ม ได้ออกเครื่องไอบีเอ็มพีซี บริษัท Hercules Computer Technology ก็ได้ผลิตการ์ดเซอร์วิคัลกราฟฟิก (Hercules Graphics) สำหรับเป็นอะแดปเตอร์ของจอ โมโนโครมที่แสดงกราฟฟิกได้

จอโมโนโครมที่คอมเพทิเบิลกับจอ โมโนโครมของไอบีเอ็ม จะต้องมียอตราการรีเฟรชจอภาพในแนวตั้งเท่ากับ 50 เฮิรตซ์ (การแสดงผลต้องทำการรีเฟรชภาพ 50 ครั้งต่อวินาที)

ปกติการทำงานในโหมดกราฟฟิกโมโนโครมของ EGA จะไม่คอมเพทิเบิลกับการทำงานของการ์ดเซอร์วิคัลกราฟฟิก ซึ่งความละเอียดการแสดงผลของการ์ดเซอร์วิคัล (720 x 348 จุด) จะสูงกว่าความละเอียดของการ์ด EGA (640 x 350 จุด) ทำให้มีการเพิ่มการทำงานในโหมดเซอร์วิคัลคอมเพทิเบิลเข้าในโหมดการทำงานของการ์ด EGA

จอสีมาตรฐาน

ไอบีเอ็ม ได้วางมาตรฐานสำหรับการแสดงผลสำหรับเครื่องพีซี โดยได้ผลิตจอสีมาตรฐาน (Color Displays : CD) และอะแดปเตอร์แสดงกราฟฟิกสี CGA ซึ่งสามารถแสดงกราฟฟิกสีได้ 4 สี และแสดงข้อความได้ 8 สี โดยจอสีมาตรฐานสามารถแสดงได้ถึง 16 สี อย่างไรก็ตาม จอสีมาตรฐานมีความละเอียดการแสดงผลต่ำเพียง 640 x 200 จุด ทำให้การแสดงผลยังมองเห็นเป็นจุด ๆ หรือไม่ต่อเนื่อง นอกจากนี้การ์ด CGA รุ่นแรกยังมีปัญหาเรื่องการรีเฟรชจอภาพถูกรบกวนในขณะที่เขียนหน่วยความจำแสดงผล ในการแก้ปัญหาที่ซอฟต์แวร์ที่ใช้งานร่วมกับการ์ด CGA จะหยุดการรีเฟรชจอภาพในขณะที่เขียนหน่วยความจำ แต่ก็ยังคงมีปัญหานี้เรื่องภาพกะพริบอยู่บ้าง จอสีมาตรฐานมียอตราการรีเฟรชภาพเท่ากับ 60 เฮิรตซ์

ถึงแม้ว่าการ์ด CGA จะมีข้อบกพร่องดังกล่าว แต่เมื่อพิจารณาในด้านราคาที่ต่ำและความสามารถในการแสดงกราฟฟิกก็ไม่นับได้ว่าการ์ด CGA ประสบความสำเร็จอย่างดีเยี่ยมและมีผู้ใช้กันอย่างแพร่หลาย

จอ ECD

จอ ECD (Enhanced Color Displays) และการ์ด EGA ได้ถูกผลิตออกมาใช้งานควบคู่กัน โดยจอ ECD ได้ปรับปรุงขึ้นมาจากจอสีมาตรฐานให้มีความละเอียดถึง 640 x 350 จุด และแสดงสีได้ 64 สี ส่วนในโหมดข้อความ (text mode) ได้ปรับปรุงชุดตัวอักษรใหม่ให้แต่ละหน่วยอักษร (character cell) มีความกว้างเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดข้อความ (text mode) ได้ปรับปรุงชุดตัวอักษรใหม่ให้แต่ละหน่วยอักษร (character cell) มีความกว้าง 8 จุด และสูง 14 จุด ซึ่งใกล้เคียงกับคุณภาพอักษรของการ์ด MDA และจัดการกะพริบของภาพดังที่พบในการใช้งานการ์ด CGA

การ์ด EGA มีอัตราการรีเฟรชภาพในแนวตั้งเท่ากับ 60 เฮิร์ตซ์ และมีอัตราการรีเฟรชตามแนวนอนเท่ากับ 15.75 กิโลเฮิร์ตซ์ หรือ 21.8 กิโลเฮิร์ตซ์

เมื่อได้พิจารณาถึงแนวต้นความต้องการจอที่มีความละเอียดสูง บริษัท NEC ได้เริ่มผลิตจอที่ซิงค์ (Sync) หลายความถี่ขึ้นมาเรียกว่า *มัลติซิงค์ (Multisync)* ซึ่งสามารถทำงานในช่วงความถี่ การซิงค์ตามแนวนอนและแนวตั้งที่กว้างและแสดงภาพได้หลายความละเอียด ซึ่งต่อมาบริษัทอื่น เช่น โซนี่ , ทอมสัน และมิตซูบิชิ ก็หันมาผลิตจอชนิดนี้ โดยจอแบบนี้ให้ความละเอียดตั้งแต่ระดับ EGA (640 x 350 จุด , 640 x 480 จุด , 800 x 600 จุด , 1024 x 768 จุด หรือมากกว่านั้น) นอกจากนี้จอที่ซิงค์หลายความถี่สามารถแสดงสีได้มากกว่าจอ ECD ซึ่งดำเนินงานในแบบดิจิตอลจะแสดงสีได้สูงสุด 64 สีเหมือนจอ ECD แต่ดำเนินงานในแบบอนาล็อก จอนี้จะทำงานแบบจอ RGB ซึ่งแสดงสีได้มากมายไม่จำกัด จอที่ซิงค์หลายความถี่ในปัจจุบันส่วนใหญ่จะคอมแพททิเบิลกับจอ VGA

จอ VGA

การ์ด VGA (Video Graphic Adapter) ได้รับการแนะนำเข้าสู่ตลาดพร้อมกับจอ VGA (VGA Displays) โดยให้ความละเอียดสูงและมีให้เลือกทั้งแบบสีและแบบโมโนโครม โดยแบบโมโนโครมสามารถแสดงรายละเอียดสีในรูปแบบระดับสีเทา และจอทั้งสองแบบสามารถใช้งานทดแทนกันได้ กล่าวคือ จอสีสามารถใช้งานแบบโมโนโครมได้และจอโมโนโครมก็ใช้งานแบบจอสีได้

4.2 โหมดการทำงานมาตรฐานของ EGA

ในการกำหนดมาตรฐานการติดต่อจอภาพสำหรับซอฟต์แวร์ใช้งาน บริษัทไอบีเอ็ม ได้วางมาตรฐานการทำงานในโหมดต่าง ๆ สำหรับ EGA และ VGA ขึ้น ดังต่อไปนี้

ตารางที่ 4.2 โหมดการแสดงผลตามมาตรฐานไอบีเอ็ม

โหมด	ชนิดของกรมแสดงผล	สี	ความละเอียด	จอแสดงผล
0,1	ข้อความสี	16	40*25 8*8 char cell	CD,ECD,VGA จอซิงค์หลายความถี่
0,1	ข้อความสี	16	40 * 25 8 * 14 char cell	ECD,VGA จอซิงค์หลายความถี่
0+,1+			40 * 25 9 * 16 char cell	VGA จอซิงค์หลายความถี่
2,3	ข้อความสี	16	80 * 25 8 * 8 char cell	CD,ECD,VGA จอซิงค์หลายความถี่

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสำนักงานเพื่อการพัฒนาเทคโนโลยีสารสนเทศและการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 (ต่อ) โหมดการแสดงผลตามมาตรฐานไอบีเอ็ม

โหมด	ชนิดจอแสดงผล	สี	ความละเอียด	จอแสดงผล
2,3	ข้อความสี	16	80 * 25 9 * 14 char cell	ECD,VGA จอซิงค์หลายความถี่
2+,3+	ข้อความสี	16	80 * 25 9 * 16 char cell	VGA จอซิงค์หลายความถี่
4,5	กราฟิกส์สี	4	320*200	CD,ECD,VGA จอซิงค์หลายความถี่
6	กราฟิกส์สี	2	640*200	CD,ECD,VGA จอซิงค์หลายความถี่
7	ข้อความโมโนโครม	2	80 * 25 9 * 14 char cell	จอโมโนโครม VGA
7+	ข้อความโมโนโครม	2	80 * 25 9 * 16 char cell	เฉพาะ VGA
8,9,A	เฉพาะ PC jr			
D	กราฟิกส์สี	16	320*200	CD,ECD,VGA จอซิงค์หลายความถี่
E	กราฟิกส์สี	16	640*200	CD,ECD,VGA
F	กราฟิกส์โมโนโครม		640*350	โมโนโครม VGA
10	กราฟิกส์สี	16	640*350	ECD,VGA จอซิงค์หลายความถี่
11	กราฟิกส์สี	2	640*480	VGA จอซิงค์หลายความถี่
12	กราฟิกส์สี	16	640*480	VGA จอซิงค์หลายความถี่
13	กราฟิกส์สี	256	320*200	VGA จอซิงค์หลายความถี่

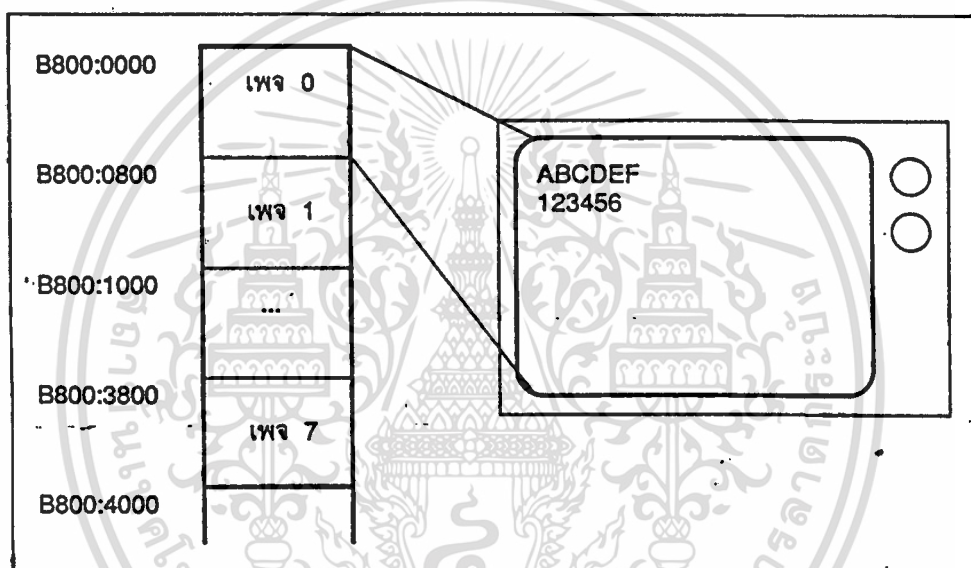
โหมด 0 และ 1 (ข้อความสี)

สำหรับการ์ด EGA และ VGA การทำงานใน โหมด 0 และ 1 จะไม่ต่างกัน โดยได้จำลองจากการทำงานของการ์ด CGA และจะให้ผลเหมือนกันเมื่อต่อที่ขั้ว CGA Composite Video Output ในโหมด 0 และ 1 จะแสดงข้อความสีได้ทั้งบนจอสีมาตรฐาน , จอ ECG , จอ VGA และจอที่ซิงค์หลายความถี่ โดยแสดงแบบ 25 บรรทัด 40 คอลัมน์อักษร และเป็นอักษรแบบการ์ด CGA คือแต่ละตัวอักษรใช้ 8 x 8 จุด ซึ่งให้คุณภาพตัวอักษรต่ำเช่นเดียวกับการใช้การ์ด CGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ์ด EGA และ VGA สามารถใช้เพจ (page) ในการแสดงผลได้ถึง 8 เพจ การเลือกเพจที่จะแสดงผลในโหมดนี้ไม่นิยมใช้การเรียกโดยไบออส (BIOS cell) มากกว่าการเปลี่ยนค่ารีจิสเตอร์แอดเดรสเริ่มต้น (Start Address Register) ในตัวควบคุม CRT ถึงแม้ว่าจะให้ผลเหมือนกัน เพจในการแสดงผลสามารถข้ามผ่านแอดเดรสต่อไปนี้

เพจ 0 - B800:0000	เพจ 4 - B800:2000
เพจ 1 - B800:0800	เพจ 5 - B800:2800
เพจ 2 - 0800:1000	เพจ 6 - B800:3000
เพจ 3 - 0800:1800	เพจ 7 - B800:3800



รูปที่ 4.1 เพจการแสดงผลในโหมด 0 และ 1

โหมดนี้สามารถแสดงได้ 8 สีมาตรฐานและ 8 สีสว่าง แต่ละอักษรสามารถโปรแกรมสีพื้นและสีตัวอักษรได้อย่างอิสระ ในตารางที่ 4.3 แสดงแอดทริบิวต์ (Attribute) ข้อความสีมาตรฐาน สำหรับรายละเอียดจะกล่าวต่อไป

ตารางที่ 4.3 แอดทริบิวต์ข้อความสีมาตรฐาน

แอดทริบิวต์	สีมาตรฐาน	สีสว่าง
000	Black	Gray
001	Blue	Light Blue
010	Green	Light Green
011	Cyan	Light Cyan
100	Red	Light Red
101	Magenta	Light Magenta
110	Brown	Yellow
111	Gray	White

การกวาดซ้ำ

การทำงานในโหมดที่คอมแพทิเบิลกับ CGA บนจอ VGA ซึ่งมีความละเอียดสูง (เส้นการสแกน 400 เส้น) จะใช้เทคนิคการกวาดซ้ำ (Double Scanning) ในการแสดงภาพความละเอียดต่ำแบบ CGA (เส้นการสแกน 200 เส้น) โดยในการกวาดภาพตามแนวนอนจะใช้ข้อมูลเพิ่มเติมในการกวาดซ้ำกัน 2 เส้นการสแกนที่อยู่ติดกัน ซึ่งเป็นการเพิ่มความละเอียดในการกวาดภาพจาก 200 เส้น เป็น 400 เส้น ทำให้คุณภาพของการแสดงภาพดีขึ้น เทคนิคการกวาดภาพซ้ำนี้จะใช้ในโหมด 0, 1, 2, 3, 4, 5, 6, D และ E

โหมด 0* และ 1* (ข้อความดี)

โหมด 0* และ 1* พัฒนาเพิ่มเติมจาก โหมด 0 และ 1 โดยสามารถแสดงตัวอักษรได้ละเอียด 8 x 14 จุดตามแบบ EGA ในแบบ 25 บรรทัด 40 คอลัมน์ แต่ไม่สามารถใช้กับจอสีมาตรฐานได้โดยต้องใช้กับจอ EGA, จอ VGA และจอที่ซิงค์หลายความถี่เท่านั้น ในโหมดนี้จะแสดงตัวอักษรได้ดีกว่าในโหมด 0 และ 1 เนื่องจากใช้ตัวอักษรต่างชุดกัน แต่ยังคงมีจำนวนเพจและการแสดงสีเหมือนกับโหมด 0 และ 1

โหมด 2 และ 3 (ข้อความดี)

ลักษณะการทำงานและแสดงผลเหมือนกับโหมด 0 และ 1 ซึ่งคอมแพทิเบิลกับ CGA แต่ต่างกันที่จำนวนคอลัมน์ในการแสดงอักษร โดยในโหมด 2 และ 3 จะแสดงแบบ 25 บรรทัด 80 คอลัมน์ สำหรับโหมด 2 และ 3 นี้ ถ้ามีหน่วยความจำแสดงผลเพียง 64 กิโลไบต์ จะสร้างเพจได้เพียง 4 เพจเท่านั้น ซึ่งปกติใช้ได้ถึง 8 เพจ

โหมด 2* และ 3* (ข้อความดี)

โหมดนี้จะทำงานในลักษณะเดียวกับโหมด 0* และ 1* โดยโหมด 2* และ 3* จะใช้จำนวนคอลัมน์ในการแสดงอักษรแตกต่างออกไป โดยแสดงแบบ 25 บรรทัด 80 คอลัมน์และในการทำงานโหมด 2* และ 3* นี้ ถ้ามีหน่วยความจำแสดงผลเพียง 64 กิโลไบต์ จะสร้างเพจได้เพียง 4 เพจเท่านั้น

โหมด 4 และ 5 (กราฟฟิก 4 สี 320 x 200 จุด)

โหมด 4 และ 5 เป็นโหมดการแสดงผลกราฟฟิกแบบ CGA ที่ EGA และ VGA จำลองมา โดยข้อแตกต่างระหว่าง 2 โหมดนี้จะหมดไป เมื่อต่อจอกับการ์ดที่ซิว CGA Composite Video Output โหมดนี้สามารถแสดงผลได้ละเอียด 320 x 200 จุด 4 สี โดยจะเป็นชุดสีมาตรฐาน หรือชุดสีเพิ่มเติมก็ได้

ตารางที่ 4.4 ชุดสีมาตรฐาน - โหมด 4 และ 5

สีมาตรฐาน	สีพิเศษ
Black	Black
Light Cyan	Green
Light Magenta	Red
White	Brown

การทำงานของ EGA และ VGA ในโหมดนี้ไม่คอมแพทibileกับการทำงานของ CGA อย่างสมบูรณ์ และเมื่อใช้โหมดนี้กับซอฟต์แวร์สำหรับ CGA ที่มีการเขียนรีจิสเตอร์ของ CGA โดยตรงอาจเกิดข้อผิดพลาดได้ แต่ถ้าเขียนรีจิสเตอร์โดยการเรียกใช้ฟังก์ชันไบออสจะยังคงให้ผลถูกต้อง

ในโหมดนี้ใช้หน่วยความจำแสดงผลเพียงเพจเดียว เริ่มต้นที่แอดเดรส B800:0000 ข้อมูลทีแต่ละจุดภาพจะเก็บแบบ 2 บิตต่อ 1 จุดภาพ

การทำงานในโหมด 4 และ 5 กับการ์ด VGA นี้ VGA จะใช้เทคนิคการกวาดซ้ำในการแสดงผล

โหมดกราฟฟิกแบบ CGA จะมีการทำงานที่ต่างไปจากการทำงานของตัวควบคุม CRT ที่ใช้บนการ์ด CGA (โมโตโรล่า 6845) ซึ่งจำกัดการกวาดภาพตามแนวนอนไว้เพียง 128 เส้นในโหมดกราฟฟิก แต่ในการแสดงผลให้มีความละเอียดในการกวาดภาพตามแนวนอนเป็น 200 เส้น ตัวควบคุม CRT ต้องทำงานในโหมดข้อความและกำหนดโปรแกรมให้แสดงอักษร 100 บรรทัด แต่ละอักษรสูง 2 จุดภาพ และใช้หน่วยความจำแสดงผลส่วนที่เป็นแบบบิตแมป ทำหน้าที่เหมือนตัวสร้างอักษร (Character Generator) จึงต้องแปลงข้อมูลที่เป็นตำแหน่งจุดภาพบนหน้าจอไปยังตำแหน่งในหน่วยความจำแสดงผลที่ถูกต้อง

โหมด 6 (กราฟฟิก 2 สี 640 x 200 จุด)

โหมด 6 ของ EGA และ VGA เป็นโหมดการทำงานที่แสดงกราฟฟิกความละเอียดสูงสุดตามแบบ CGA ให้ความละเอียด 640 x 200 จุด แต่แสดงได้เพียง 2 สี และใช้ได้กับจอสีมาตรฐาน, จอ ECD, จอ VGA และจะทิ้งหลายความถี่ ส่วนรายละเอียดและการทำงานทั่วไปเหมือนกับการทำงานในโหมด 4 และ 5 ต่างกันเพียงความละเอียดที่เพิ่มขึ้นและจำนวนสีที่แสดงได้ลดลง

โหมด 7 (ข้อความโมนोโครม)

โหมดนี้เป็นการแสดงผลเลียนแบบการแสดงผลของ MDA (Monochrome Display Adapter) ใช้ได้กับจอโมนิโครม หรือจอ VGA เท่านั้น โหมดนี้จะแสดงข้อความแบบ 25 บรรทัด 80 คอลัมน์ และใช้ชุดตัวอักษรขนาดกว้าง 8 จุด สูง 14 จุด และเมื่อแสดงผลจะขยายความกว้างอักษรเป็น 9 จุด เพื่อให้มีความละเอียดแบบจอโมนิโครม คือ 720 จุดตามแนวนอน

การแสดงผลที่ 9 เพิ่มไปในแต่ละอักษรเป็นสิ่งจำเป็น เพราะถ้าไม่มีจุดที่ 9 จะเกิดช่องว่างในการแสดงตัวอักษรกราฟฟิกที่ต้องการความต่อเนื่อง หรือขาดความต่อเนื่องในรูปกราฟฟิกง่าย ๆ ที่ทำได้ เช่น เส้นกรอบ เส้นตรง เป็นต้น ในการแสดงผลข้อความในแบบกราฟฟิกนี้ จะทำการจำลองบิตที่ 9 จากข้อมูลของบิตที่ 8 ของแต่ละอักษร ในการควบคุมการจำลองบิตที่ 9 นี้ จะต้องอินาเบิ้ล (enable) หรือดิสเอบิ้ล (disable) ผ่านรีจิสเตอร์ควบคุม รหัสแอสกีสำหรับชุดตัวอักษรกราฟฟิกจะอยู่ระหว่างรหัส C0 กับ DF ในฐานสิบหก

ปกติการแสดงผลในโหมดนี้ใช้ 8 เพจ ยกเว้นในกรณีที่มีหน่วยความจำแสดงผลบนการ์ด EGA เพียง 64 กิโลไบต์ จะใช้เพียง 4 เพจ การเลือกเพจที่จะแสดงผลสามารถเรียกผ่านไบออส หรืออาจจะเลือกโดยการตั้งค่าที่รีจิสเตอร์แอดเดรสเริ่มต้นในตัวควบคุม CRT เพจสำหรับการแสดงผลโหมดนี้ที่แอดเดรสดังต่อไปนี้

เพจ 0 - B000:0000

เพจ 4 - B000:4000

เพจ 1 - B000:1000

เพจ 5 - B000:5000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพจ 0 - B000:0000	เพจ 4 - B000:4000
เพจ 1 - B000:1000	เพจ 5 - B000:5000
เพจ 2 - B000:2000	เพจ 6 - B000:6000
เพจ 3 - B000:3000	เพจ 7 - B000:7000

โหมดนี้แอดทริบิวต์อักษรจะไม่ควบคุมลักษณะสี แต่จะทำหน้าที่ควบคุมลักษณะอักษรอื่น ๆ แทน เช่น ความสว่างของอักษร การกะพริบ การขีดเส้นใต้ และลักษณะสัญญาณภาพตรงข้าม (Reverse Video)

โหมด D (กราฟฟิก 16 สี 320 x 200 จุด)

โหมดนี้พัฒนาขึ้นมาจากการแสดงกราฟฟิก 4 สีแบบ CGA โดยแสดงสีได้มากขึ้น (16 สี) แต่ยังมี ความละเอียดต่ำ (320 x 200 จุด) จึงไม่เป็นที่นิยม โหมดนี้ทำงานได้บนจอสีมาตรฐาน , จอ ECD , จอ VGA และจอที่ซิงค์หลายความถี่ ซอฟต์แวร์ส่วนใหญ่จะไม่สามารถใช้งานในโหมดนี้ได้

ถ้าการ์ด EGA ที่ใช้มีหน่วยความจำ 256 กิโลไบต์ จะใช้การแสดงผลได้ 8 เพจ ถ้ามี 128 กิโลไบต์ จะใช้ได้ 4 เพจ หรือถ้ามี 64 กิโลไบต์ จะใช้ได้เพียง 2 เพจ โดยแต่ละเพจอยู่ที่แอดเดรสดังต่อไปนี้

เพจ 0 - A000:0000	เพจ 4 - A000:4000
เพจ 1 - A000:1000	เพจ 5 - A000:5000
เพจ 2 - A000:2000	เพจ 6 - A000:6000
เพจ 3 - A000:3000	เพจ 7 - A000:7000

การแสดงกราฟฟิกในโหมด D จะส่งตำแหน่งจุดที่จะแสดงแต่ละจุด ไปยังหน่วยความจำแสดงผลที่ สมนัยกับตำแหน่งนั้นโดยตรง สีมาตรฐานที่แสดงได้ในโหมด D แสดงในตารางที่ 5 ในโหมดนี้การ์ด VGA จะแสดงโดยใช้เทคนิคการกวาดซ้ำ

โหมด E (กราฟฟิก 16 สี 640 x 200 จุด)

การทำงานของโหมด E โดยทั่วไปคล้ายกับโหมด 6 คือ แสดงได้ถึง 16 สี แต่ยังมีข้อจำกัดในเรื่อง ความละเอียด (640 x 200 จุด) ที่ต่ำอยู่ ส่งผลให้โหมด E ไม่เป็นที่นิยมและใช้งานในวงจำกัด และแทบไม่พบเห็นการใช้งาน

ถ้าการ์ด EGA ที่ใช้มีหน่วยความจำ 256 กิโลไบต์ จะใช้การแสดงผลได้ 4 เพจ ถ้ามี 128 กิโลไบต์ จะใช้ได้ 2 เพจ หรือถ้ามี 64 กิโลไบต์ จะใช้ได้เพียง 1 เพจ โดยแต่ละเพจอยู่ที่แอดเดรสดังต่อไปนี้

เพจ 0 - A000:0000
เพจ 1 - A000:4000
เพจ 2 - A000:8000
เพจ 3 - A000:C000

สำหรับการจัดการด้านกราฟฟิก จะแมปตำแหน่งที่จะแสดงกับตำแหน่งในหน่วยความจำแสดงผลอย่าง ตรงไปตรงมาเช่นเดียวกับในโหมด D และแสดงสีมาตรฐานได้เหมือนกันดังตารางที่ 5

เมื่อทำงานในโหมด E การ์ด VGA จะแสดงผลโดยใช้เทคนิคการกวาดซ้ำ

โหมด F (กราฟฟิกโมโนโครม 640 x 350 จุด)

โหมด F เป็นโหมดการทำงานเฉพาะสำหรับการ์ด EGA และ VGA โดยใช้กับจอโมโนโครม หรือ จอ VGA เท่านั้น มีความละเอียด 640 x 350 จุด ซึ่งยังน้อยกว่าความละเอียดของการ์ดเซอร์คิวลิส (720 x 348 จุด)

โหมด F จะทำการแมปโดยตรงระหว่างตำแหน่งบนจอกับตำแหน่งในหน่วยความจำแสดงผล การแสดงผลใช้ 2 เพจ โดยอยู่ที่แอดเดรสดังต่อไปนี้

เพจ 1 - A000:0000

เพจ 2 - A000:8000

นอกจากนี้ยังใช้เพลนสี 2 เพลน ทำให้แต่ละจุดภาพแสดงตามแอดทริบิวต์ได้ 4 แบบ ดังนี้

00 - สีดำ

01 - สีขาว

10 - กะพริบ

11 - สว่าง (Intensified)

การเลือกเพลนบนหน่วยความจำแสดงผลสามารถทำได้โดยอิสระ โดยการกำหนดที่รีจิสเตอร์เลือกเพลนในส่วนซีแควนเซอร์

โหมด 10 (กราฟฟิกแบบ Enhanced 640 x 350 จุด)

ตารางที่ 4.5 สีมาตรฐานสำหรับโหมดการแสดงผล 16 สี

เพลนสี 3210	สี (หน่วยความจำ 128K+)	สี (หน่วยความจำ 64K)
0000	Black	Black
0001	Blue	Blue
0010	Green	Black
0011	Cyan	Blue
0100	Red	Red
0101	Magenta	White
0110	Brown	Red
0111	White	White
1000	Dark Gray	Black
1001	Light Blue	Blue
1010	Light Green	Black
1011	Light Cyan	Blue
1100	Light Red	Red
1101	Light Magenta	White
1110	Yellow	Red
1111	Intens. White	White

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ของบุคลากรศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 10 เป็นโหมดที่นิยมใช้กันมากในการแสดงกราฟฟิกให้ความละเอียด 640 x 350 จุด โหมด 10 นี้สามารถใช้ได้กับจอ EGA , จอ VGA และจอที่ซิงค์หลายความถี่ แต่ไม่สามารถใช้ได้กับจอมาตรฐาน การแสดงผลใช้เฟรม 4 เฟรม ทำให้แสดงได้ถึง 16 สีพร้อมกัน (โดยถ้ามีหน่วยความจำแสดงผล 64 กิโลไบต์ จะสามารถใช้ได้เพียง 2 เฟรม) และการเลือกเฟรมนี้จะควบคุมผ่านรีจิสเตอร์เลือกเฟรมในส่วนซีเคานเซอร์

การทำงานจะใช้ 2 เฟจ (ยกเว้นถ้าหน่วยความจำมี 64 กิโลไบต์ จะใช้เฟจเดียว) โดยอยู่ที่ ตำแหน่ง A000:0000 และ A000:8000

สีมาตรฐานที่ใช้ในโหมด 10 แสดงในตารางที่ 4.5 โดยสีเหล่านี้สามารถเปลี่ยนแปลงได้โดยการโปรแกรมรีจิสเตอร์สี่ของตัวคุมแอดทริบิวต์

4.3 โหมดการทำงานเพิ่มเติมสำหรับ VGA

โหมดการทำงานเหล่านี้ใช้ได้เฉพาะกับการ์ด VGA และจอ VGA และจอที่ซิงค์หลายความถี่เท่านั้น

โหมด 0+ , 1+ (ข้อความสี)

โหมดนี้ปรับปรุงขึ้นมาจากการทำงานโหมด 0 , 1 ซึ่งเขียนแบบการแสดงผลของการ์ด CGA มาตรฐาน โหมดนี้ทำการแสดงผลข้อความแบบ 25 บรรทัด 40 คอลัมน์ ใช้จุดตัวอักษรกว้าง 9 จุด สูง 16 จุด โหมด 0+ , 1+ ยังคงมีปัญหาเช่นเดียวกับโหมด 0* , 1*

โหมด 2+ , 3+ (ข้อความสี)

โหมด 2+ , 3+ พัฒนารับขึ้นมาจากการทำงานในโหมด 2 และ 3 โดยแสดงข้อความแบบ 25 บรรทัด 80 คอลัมน์ และใช้จุดตัวอักษรขนาดกว้าง 9 จุด สูง 16 จุด โหมด 2+ , 3+ ยังคงมีปัญหาเช่นเดียวกับ 2* , 3*

โหมด 7+ (ข้อความโมโนโครม)

โหมดนี้เป็นการเขียนแบบการสร้างอักษรตามแบบของ MDA แต่ใช้จุดอักษรชนิด 9 x 16 จุด โหมด 7+ ยังคงมีปัญหาเหมือนกับโหมดอื่น ๆ ที่ใช้จุดอักษรแบบนี้

โหมด 11 (กราฟฟิก 2 สี 640 x 480 จุด)

โหมด 11 เป็นโหมดสนับสนุนการใช้งานการ์ด VGA โดยใช้แสดงกราฟฟิกให้ความละเอียดสูงสุดถึง 640 x 480 จุด แต่แสดงสีได้เพียง 2 สีพร้อมกัน โหมดนี้ยังสามารถแสดงข้อความแบบ 30 บรรทัด 80 คอลัมน์ สำหรับหน่วยความจำแสดงผลในโหมดนี้เริ่มที่แอดเดรส A000:0000

โหมด 12 (กราฟฟิก 16 สี 640 x 480 จุด)

โหมด 12 ใช้กับการ์ด VGA ให้ความละเอียดสูงสุด 640 x 480 จุด และสามารถแสดงได้ 16 สีพร้อมกัน สีมาตรฐานที่แสดงได้ในโหมดนี้แสดงในตารางที่ 3 หน่วยความจำแสดงผลจะเริ่มจากแอดเดรส

A000:0000 เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พร้อมกัน สามารถแสดงได้ในโหมดนี้แสดงในตารางที่ 3 หน่วยความจำแสดงผลจะเริ่มจากแอดเดรส A000:0000

โหมด 13 (กราฟฟิก 256 สี 320 x 200 จุด)

โหมดนี้สามารถแสดงได้ถึง 256 สีพร้อมกัน แต่มีความละเอียดต่ำ 320 x 200 จุด ส่วนหน่วยความจำแสดงผลจะเริ่มจากแอดเดรส A000:0000

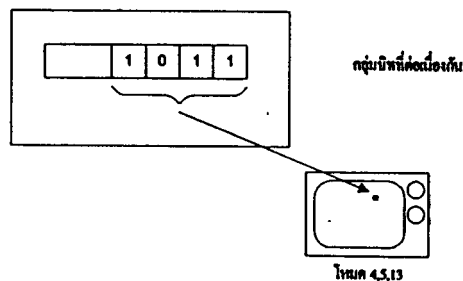
4.4 สถาปัตยกรรม EGA / VGA

ในการแสดงผลของการ์ด EGA และ VGA การ์ดทั้งสองชนิดไม่มีความสามารถในการประมวลผลด้วยตัวเอง ต้องอาศัยการประมวลผลจากตัวประมวลผลหลัก แล้วส่งข้อมูลไปให้หน่วยความจำแสดงผลเพื่อแสดงแต่ละจุดภาพออกหน้าจอ วงจรส่วนใหญ่ของการ์ด EGA และ VGA จะถูกใช้ในการส่งข้อมูลจากหน่วยความจำแสดงผลออกจอภาพ โดยเรียกการทำงานลักษณะนี้ว่า *การรีเฟรชจอภาพ (Display Refresh)* ซึ่งจะทำการแสดงผลซ้ำ ๆ กันด้วยอัตรา 60 ครั้งต่อวินาทีสำหรับ EGA และ 70 ครั้งต่อวินาทีสำหรับ VGA

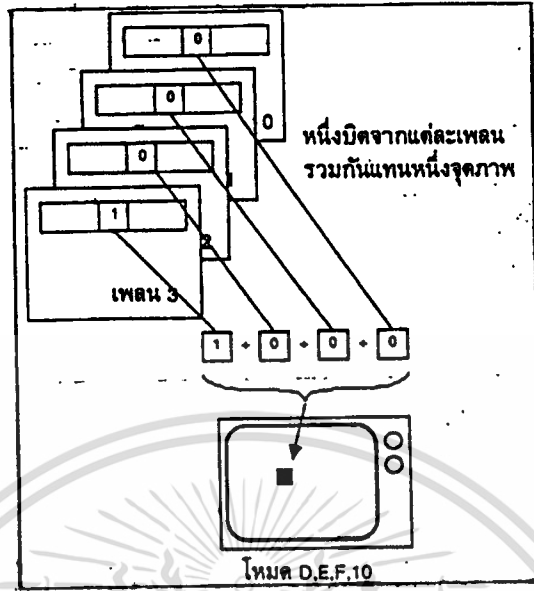
ในการแสดงผลบนจอสี จำนวนที่แสดงได้จะถูกจำกัดโดยจำนวนบิตของหน่วยความจำแสดงผล ถ้าจำนวนบิตที่ใช้ต่อจุดภาพเท่ากับ n จะแสดงสีได้ทั้งหมด 2^n สี เช่น EGA ใช้ 1 ถึง 4 บิตต่อจุดภาพ จะแสดงสีได้ 2^4 หรือ 16 สี ส่วน VGA แสดงสีได้ถึง 2^8 สี หรือ 256 สี

4.4.1 การอัดข้อมูลจุดภาพและการใช้เพลนสี

เทคนิคที่ใช้ในการเก็บข้อมูลสีมี 2 วิธี คือ วิธีการอัดข้อมูลจุดภาพ (Packed Pixels) และวิธีการใช้เพลนสี (Color Plane) สำหรับการ์ด EGA และ VGA จะใช้วิธีการใช้เพลนสี แต่การ์ดทั้งสองชนิดก็สามารถจำลองการทำงานแบบอัดข้อมูลจุดภาพได้ ซึ่งจะทำการเก็บข้อมูลจุดภาพแต่ละจุดด้วยขนาดต่าง ๆ กันไป เช่น ถ้าต้องการแสดงสีไม่มากก็อาจใช้หน่วยความจำเพียงครั้งไบต์ แต่ถ้าต้องการแสดงหลายสีก็อาจต้องใช้หน่วยความจำหลายไบต์ รูปที่ 4.2 แสดงการอัดข้อมูลจุดภาพแบบ 4 บิตต่อจุดภาพสำหรับวิธีการใช้เพลนสีจะแบ่งหน่วยความจำแสดงผลออกเป็นหลายเพลนที่เป็นอิสระต่อกัน โดยแต่ละเพลนจะใช้สำหรับควบคุมสีแต่ละสีและแต่ละบิตบนเพลนจะแทนจุดภาพบนจอแต่ละจุด แล้วรวมข้อมูลบิตที่ตรงกันของทุกเพลนเป็นข้อมูลการแสดงผลหนึ่งจุดภาพ ดังในรูปที่ 4.3



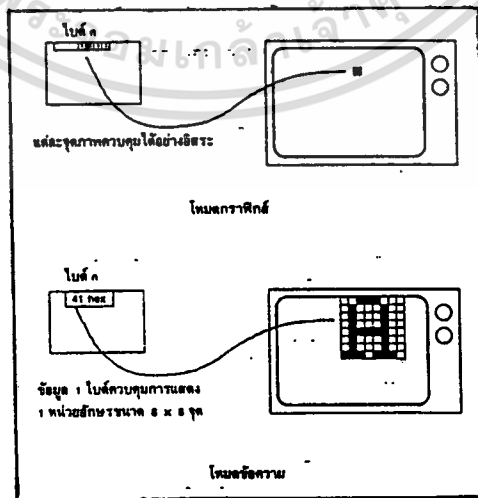
รูปที่ 4.2 ข้อมูลสีที่เก็บแบบอัดจุดภาพ



รูปที่ 4.3 ข้อมูลที่เก็บแบบแยกเฟรมลิส

4.4.2 โหมดข้อความและโหมดกราฟฟิก

ทั้งสองโหมดนี้เป็นโหมดการแสดงผลพื้นฐานของ EGA และ VGA ในโหมดกราฟฟิกแต่ละบิตของหน่วยความจำแสดงผลจะแทนแต่ละจุดภาพบนจอ ส่วนในโหมดข้อความ แต่ละไบต์ของหน่วยความจำแสดงผลจะแทนรหัสแอสกีของแต่ละอักษร ซึ่งการแสดงผลในโหมดนี้ต้องการหน่วยความจำน้อยและใช้การประมวลผลน้อยมาก แต่มีข้อจำกัดคือ แสดงได้โดยแสดงรูปแบบของตัวอักษรและรูปภาพที่ง่าย ๆ ดังรูปที่ 4.4

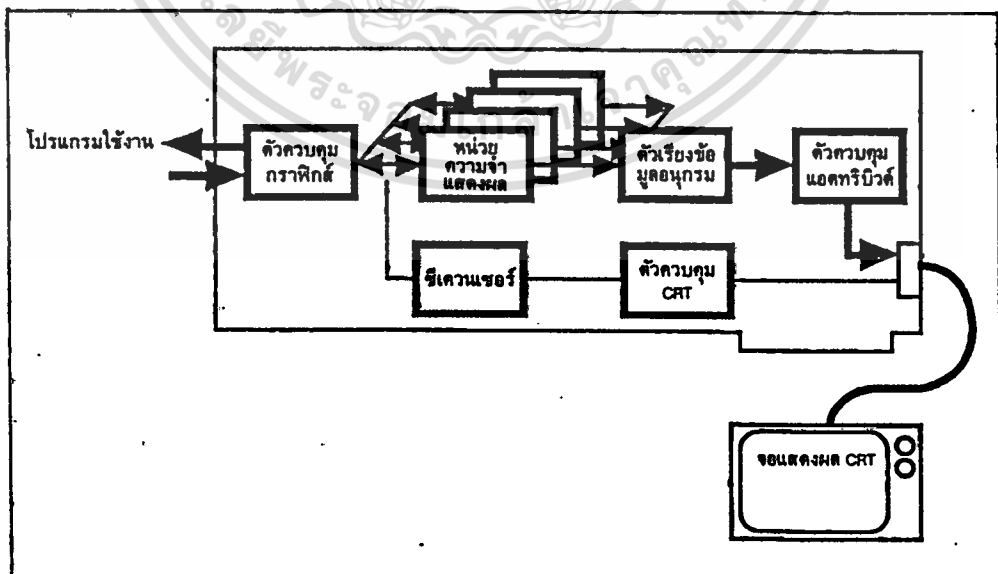


รูปที่ 4.4 การแสดงผลโหมดข้อความและโหมดกราฟฟิก

4.4.3 สถาปัตยกรรมของ EGA และ VGA

สถาปัตยกรรมพื้นฐานของ EGA แสดงดังรูปที่ 4.5 ซึ่งประกอบด้วยส่วนสำคัญ 6 ส่วนคือ

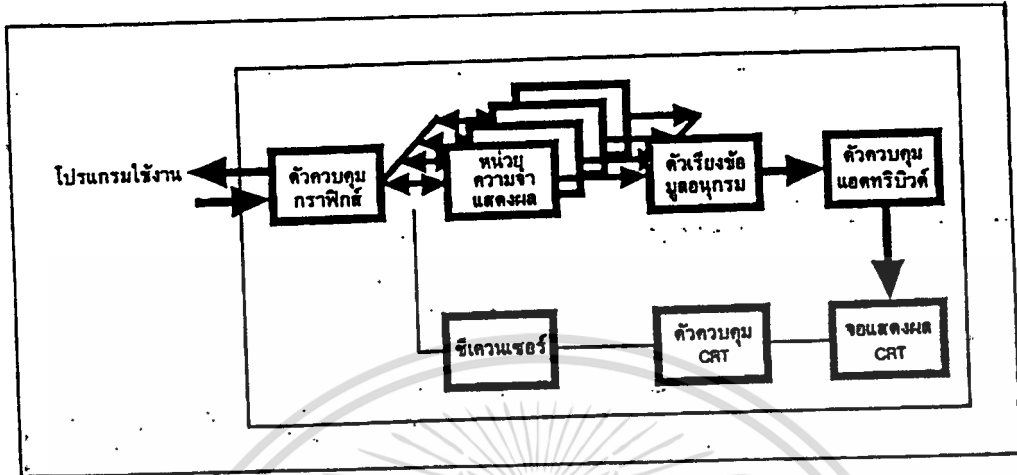
1. หน่วยความจำแสดงผล (Display Memory) ประกอบด้วยไดนามิกแรม (Dynamic Access Memory : DRAM) ขนาด 256 กิโลไบต์ และแบ่งเป็น 4 เฟลน ซึ่งใช้ในการเก็บข้อมูลที่จะนำออกแสดงผล
2. ตัวควบคุมกราฟฟิก (Graphics Controller) มีความสามารถในการทำงานทางตรรกะ (AND, OR, XOR และ ROTATE) กับข้อมูลที่จะเขียนลงบนหน่วยความจำแสดงผล
3. ตัวควบคุม CRT (CRT Controller) เป็นตัวสร้างสัญญาณทางเวลาที่ใช้ควบคุมการทำงานของ CRT และการรีเฟรชจอภาพ
4. ตัวเรียงข้อมูลอนุกรม (Data Serializer) เป็นตัวนำข้อมูลทีละบิตส่งมาจากหน่วยความจำแสดงผลในแต่ละครั้ง มาทำการแปลงข้อมูลให้อยู่ในรูปอนุกรมเพื่อส่งต่อไปยังจอ CRT
5. ตัวควบคุมแอตทริบิวต์ (Attribute Controller) ประกอบด้วยตารางค้นหาสี (Color Look-up Table : LUT) ทำหน้าที่แปลงข้อมูลสีจากหน่วยความจำแสดงผลไปเป็นข้อมูลสีที่จะส่งไปยังจอ CRT โดยทั่วไปแล้วจอสีสามารถแสดงสีได้มากกว่าความสามารถของการ์ดแสดงผล
6. ซีควเอนเซอร์ (Sequencer) เป็นตัวควบคุมเวลาในการทำงานแสดงผลทั้งหมด และยังประกอบด้วยฟังก์ชันทางตรรกะที่ใช้ในการเลือกใช้เฟลนสี



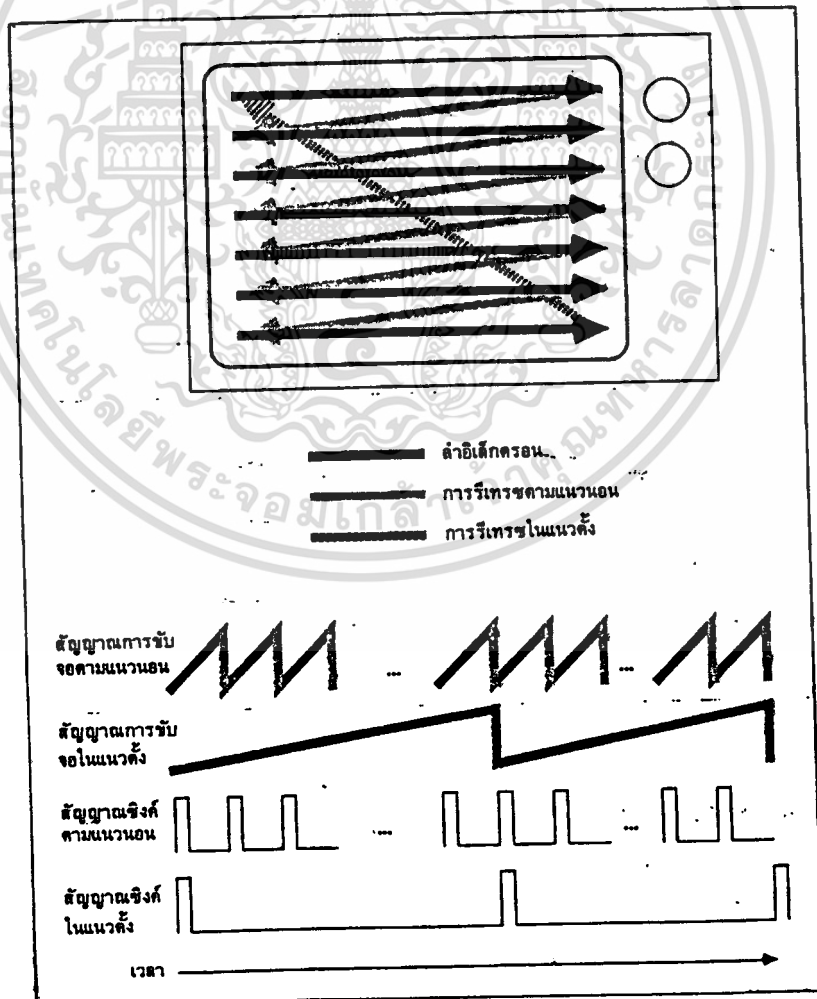
รูปที่ 4.5 บล็อกไดอะแกรมของ EGA / VGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.4 การทำงานของจอ CRT



รูปที่ 4.6 จอแสดงผล



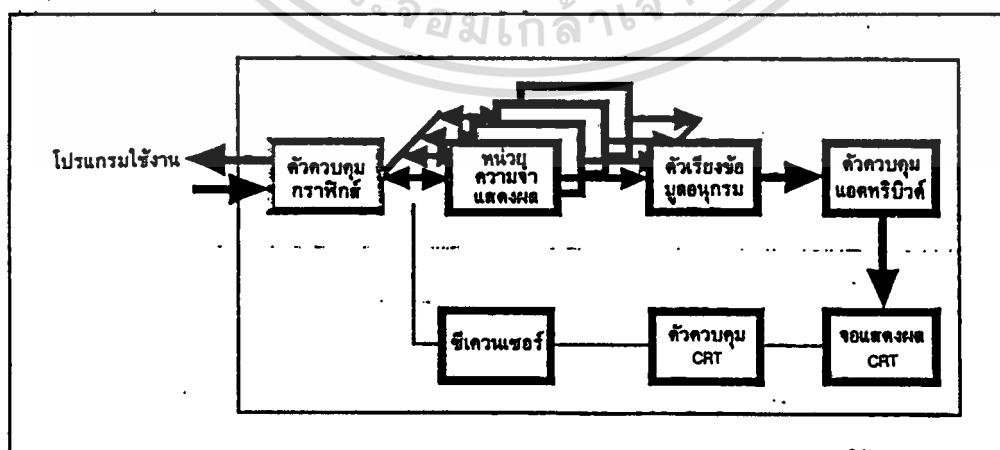
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้รูปที่ 4.7 การทำงานของจอ CRT
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของจอแบบหลอดภาพแคโทด (Cathode Ray Tube Display) จุดภาพจะถูกสร้างโดยการยิงอิเล็กตรอนไปกระทบกับผิวจอที่ฉาบด้วยสารฟอสฟอรัส ซึ่งจะเกิดการเรืองแสงขึ้นถ้าของอิเล็กตรอนจะกวาดจากซ้ายไปขวาตามแนวระดับ ในขณะที่เดียวกันความเข้มของลำอิเล็กตรอนจะเป็นตัวกำหนดรูปแบบการแสดงผล ลำอิเล็กตรอนจะถูกกวาดเข้าไปมาด้วยอัตราการกวาดภาพประมาณ 50 , 60 หรือ 70 ครั้งต่อวินาที ขึ้นอยู่กับชนิดของจอ (ดูตารางที่ 4.1) ซึ่งเรียกขบวนการนี้ว่า การรีเฟรชจอภาพ (Display Refresh) หรือ การรีเฟรชภาพ (Screen Refresh) และเรียกการกวาดของลำอิเล็กตรอนบนจอว่า *แรสเตอร์ (Raster)* ซึ่งเป็นการกวาดลำอิเล็กตรอนจากมุมบนซ้ายแล้วกวาดไปทางขวา เมื่อกวาดถึงขวาสุดก็จะทำการปิดลำอิเล็กตรอน (Blanking) จากนั้นจะไปเริ่มต้นทางซ้ายใหม่ (Retrace) เพื่อกวาดเส้นการสแกนในแนวระดับเส้นถัดไป เมื่อกวาดภาพตามแนวระดับถึงมุมขวาล่างจะปิดลำอิเล็กตรอน จากนั้นจะไปเริ่มต้นที่มุมบนซ้ายใหม่ ดังรูปที่ 4.7

วงจรรภายในจอ CRT จะสร้างลำอิเล็กตรอนเพื่อยิงไปบนผิวจอ และจะถูกควบคุมการทำงานโดยการ์ดแสดงผล โดยการ์ดแสดงผลจะส่งสัญญาณซิงค์ในแนวนอนและแนวตั้งให้กับจอ CRT

4.4.5 หน่วยความจำแสดงผล

การ์ด EGA และ VGA ประกอบไปด้วยหน่วยความจำหลัก 256 กิโลไบต์ ซึ่งแบ่งออกเป็น 4 ชุด ชุดละ 64 กิโลไบต์ แต่ละชุดเรียกว่า *เพลนสี (Color Plane)* ดังรูปที่ 4.8 โดยทั้ง 4 เพลนจะอ้างผ่านแอดเดรสหน่วยความจำเดียวกัน ในการเขียนหรือการอ่านเพลนจะถูกกำหนดโดยการตั้งค่ารีจิสเตอร์ I/O และเมื่อตัวประมวลผลเขียนข้อมูลแต่ละครั้งก็จะเขียนพร้อมกันทั้ง 4 เพลน ซึ่งความสามารถนี้จะช่วยให้การวาดรูปบางอย่างทำได้เร็วขึ้น แต่ในบางครั้งการวาดรูปต้องการใช้เพียงสีเดียวก็จะใช้เพียง 1 เพลน ดังนั้นเราสามารถกำหนดเพลนสีที่จะใช้งานได้โดยการตั้งค่ารีจิสเตอร์บางตัว



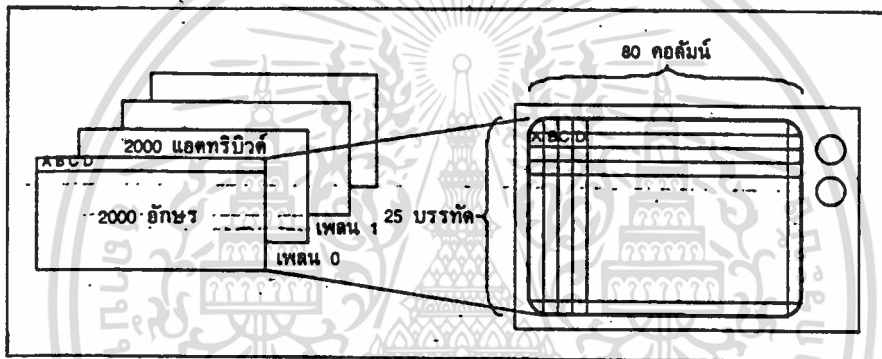
รูปที่ 4.8 หน่วยความจำแสดงผล (Display Memory)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

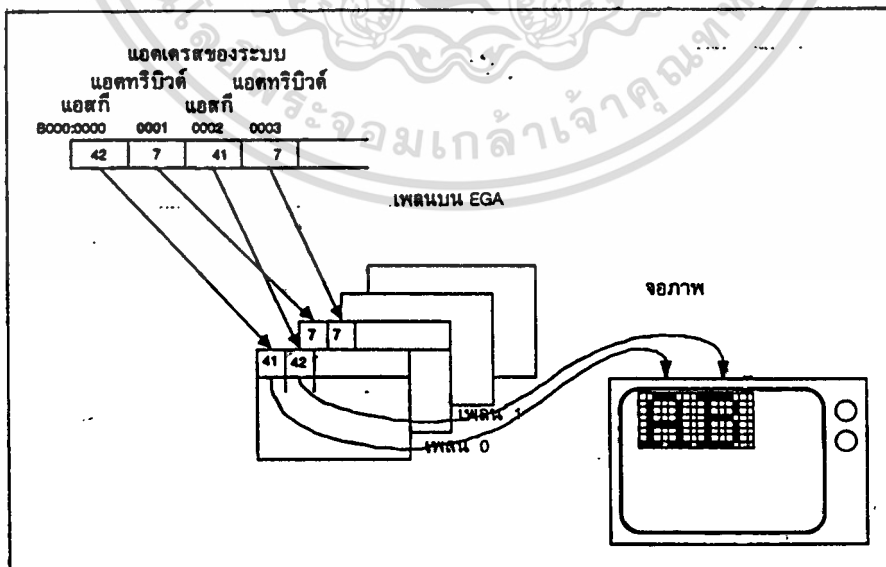
หน่วยความจำแสดงผลในโหมดข้อความ

การแสดงผลในโหมดข้อความสามารถทำได้ง่ายกว่าโหมดกราฟฟิก และนิยมใช้กับงานที่ไม่ต้องการแสดงกราฟฟิก การแสดงผลในโหมดข้อความจะลดภาระการทำงานของตัวประมวลผลซึ่งจะจัดการกับรหัสอักษรแอสกีเท่านั้น แทนที่จะต้องกำหนดรายละเอียดจุดภาพทุกจุด

ในโหมดข้อความมาตรฐาน การแสดงผลจะถูกแบ่งออกเป็น 2 แบบ คือ แบบ 25 บรรทัด 40 อักษร และแบบ 25 บรรทัด 80 อักษร สำหรับโหมด 40 อักษรต่อบรรทัด สามารถแสดงอักษรได้ 1000 อักษรในหนึ่งหน้าจอ และแสดงได้ 2000 อักษรในโหมด 80 อักษรต่อบรรทัด (ดังรูปที่ 4.9) ในกรณีนิยมตัวอักษรแต่ละตัวจะใช้หน่วยความจำ 2 ไบต์ โดยไบต์แรกอยู่ที่ตำแหน่งแอดเดรสคู่ใช้กำหนดรหัสอักษรแอสกี ส่วนไบต์ที่สองอยู่ที่ตำแหน่งแอดเดรสคี่ใช้กำหนดข้อมูลสี ซึ่งเรียกว่า *แอดเดรสสีอักษร*



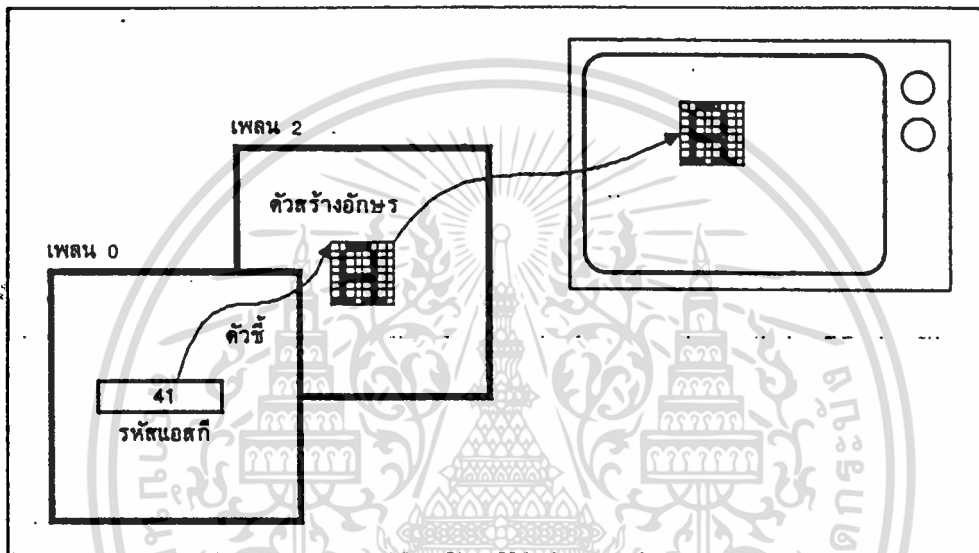
รูปที่ 4.9 หน่วยความจำแสดงผลในโหมดข้อความ



รูปที่ 4.10 แอดเดรสอ้างโดยตัวประมวลผลและเฟรมบน EGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการแปลงรหัสอักษรแอสกีไปเป็นอาร์เรย์ (Array) ของจุดภาพ จะใช้ตารางการแปลงรหัสหรือตัวสร้างอักษร (Character Generator) ในระบบการแสดงผลแบบเก่า เช่น MDA และ CGA ตัวสร้างอักษรจะอยู่ในรอม ส่วน EGA และ VGA จะไม่ใช่ตัวสร้างอักษรบนรอมแต่จะโหลดข้อมูลของตัวสร้างอักษรไปไว้ในเพลนที่ 2 ของหน่วยความจำแสดงผล การทำงานในลักษณะนี้ผู้ใช้สามารถกำหนดรูปแบบตัวอักษรเองได้ และสามารถใช้งานรูปแบบตัวอักษรหลาย ๆ แบบพร้อมกัน (มากที่สุด 4 แบบ) ซึ่งทำการโหลดโดยการเรียกใช้ฟังก์ชันไบออส



รูปที่ 4.11 การใช้รหัสอักษรตัวชี้ตัวสร้างอักษร

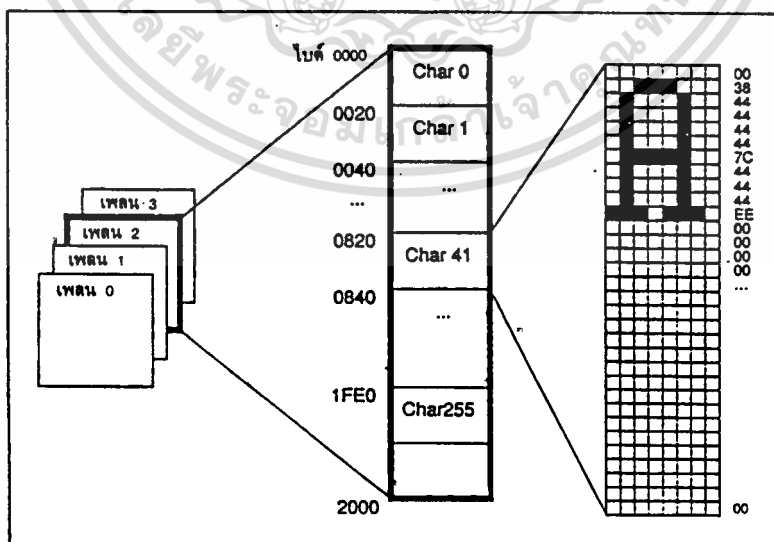
ตัวสร้างอักษร

การ์ด EGA และ VGA ให้ความสะดวกแก่ผู้ใช้ในการเลือกชุดอักษรที่จะใช้งาน โดยไม่ยึดติดกับชุดอักษรในรอม โดยจะโหลดชุดอักษรไปเก็บไว้ในเพลนที่ 2 ของหน่วยความจำแสดงผล การใช้งานการ์ด EGA สามารถโหลดชุดอักษรได้พร้อมกันถึง 4 ชุด สำหรับการแสดงผลแบบ VGA จะใช้ได้พร้อมกันถึง 8 ชุด ชุดอักษรแต่ละชุดประกอบด้วยตัวอักษร 256 ตัว และในการแสดงผลแต่ละครั้งสามารถกำหนดค่าให้ทำงานได้พร้อมกันถึง 2 ชุด ดังนั้นจึงสามารถแสดงตัวอักษรที่ต่างกันถึง 512 อักษรพร้อมกัน การเลือกชุดตัวอักษรทั้งสองชุดนี้จะเลือกโดยการกำหนดค่ารีจิสเตอร์ในส่วนซีแควนเซอร์ ซึ่งเลือกใช้ได้ 2 ชุดจาก 4 ชุด แล้วใช้หนึ่งบิตในแต่ละแอสกีบิตตัวอักษรเป็นตัวเลือกชุดอักษรที่จะนำออกแสดง

รูปแบบมาตรฐานของอักษรในการแสดงผลแบบ CGA กว้าง 8 จุด สูง 8 จุด, สำหรับ EGA กว้าง 8 จุด สูง 14 จุด และสำหรับ VGA กว้าง 9 จุด สูง 16 จุด ตำแหน่งของจุดอักษรแสดงดังตารางที่ 4.7 ความปกติตัวอักษรแต่ละตัวจะใช้หน่วยความจำขนาดตั้งแต่ 32 ไบต์ เช่น อักษรแบบ 8 x 14 จุด ต้องการ 14 ไบต์ต่อตัวอักษร ดังนั้น 18 ไบต์ที่เหลือจะไม่ถูกใช้งานดังรูปที่ 4.12

ตารางที่ 4.6 ตำแหน่งของตัวสร้างอักษรที่ฝั่งตัวบนแรม

จุดตัวอักษร A	จุดตัวอักษร B
0000h ถึง 001Fh-Char.0	2000h ถึง 201Fh-Char.0
0020h ถึง 003Fh-Char.1	2020h ถึง 203Fh-Char.1
0040h ถึง 005Fh-Char.2	2040h ถึง 205Fh-Char.2
1FE0h ถึง 1FFFh-Char.255	3FE0h ถึง 3FFFh-Char.255
จุดตัวอักษร C	จุดตัวอักษร D
4000h ถึง 401Fh-Char.0	6000h ถึง 601Fh-Char.0
5FE0h ถึง 5FFFh-Char.255	7FE0h ถึง 7FFFh-Char.255
สำหรับ VGA :	
จุดตัวอักษร E	จุดตัวอักษร F
8000h ถึง 801Fh-Char.0	A000h ถึง A01Fh-Char.0
9FE0h ถึง 9FFFh-Char.255	BFE0h ถึง BFFFh-Char.255
จุดตัวอักษร G	จุดตัวอักษร H
C000h ถึง C01Fh-Char.0	E000h ถึง E01Fh-Char.0
DFE0h ถึง DFFFh-Char.255	FFE0h ถึง FFFFh-Char.255



รูปที่ 4.12 รูปแบบการเก็บตัวสร้างอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอดทริบิวต์ข้อความ

ตัวอักษรที่จะแสดงบนจอจะมีไบต์แอดทริบิวต์สำหรับใช้กำหนดสี ลักษณะการแสดงผลข้อความของแอดทริบิวต์ข้อความขึ้นอยู่กับโหมดการทำงาน

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
การกะพริบ หรือความ สว่างสีพื้น		สีพื้น		ความสว่าง วัตถุ หรือเลือก ชุดอักษร		สีวัตถุ	

รูปที่ 4.13 หน้าที่ของแต่ละบิตในไบต์แอดทริบิวต์

แอดทริบิวต์ข้อความสีมาตรฐาน

รูปที่ 4.16 แสดงถึงความหมายของแต่ละบิตของแอดทริบิวต์ข้อความ เมื่อทำงานในโหมดข้อความสีมาตรฐาน บิต D0-D2 ทำหน้าที่กำหนดสีตัวอักษร, บิต D4-D6 ทำหน้าที่กำหนดสีพื้น และบิต D3 ทำหน้าที่กำหนดความสว่างของสีตัวอักษร ในการเพิ่มความสว่างของแสงขึ้นโดยการเปลี่ยนบิต D3 จาก 0 เป็น 1 จะเปลี่ยนสีจากปกติสีดำเป็นสีเทา หรือจากสีม่วง (Magenta) เป็นสีม่วงอ่อน (Light magenta) ซึ่งทำให้แสดงสีได้เพิ่มขึ้นอีก 8 สี ดังตารางที่ 4.7

ตารางที่ 4.7 แอดทริบิวต์สีมาตรฐาน

แอดทริบิวต์	สีมาตรฐาน	สีสว่าง
000	Black	Gray
001	Blue	Light Blue
010	Green	Light Green
011	Cyan	Light Cyan
100	Red	Light Red
101	Magenta	Light Magenta
110	Brown	Yellow
111	Gray	White

ในการแสดงผลอักษร 2 ชุดพร้อมกัน (เลือกชุดอักษรโดยใช้รีจิสเตอร์ในซีแควนเซอร์) บิต D3 จะไม่ทำหน้าที่เป็นตัวกำหนดความสว่างของตัวอักษร แต่จะใช้สำหรับเลือกชุดอักษรที่จะแสดง

บิต D7 ในไบต์แอดทริบิวต์สามารถใช้ในการกำหนดให้อักษรกะพริบหรือกำหนดความสว่างของสีพื้น โดยใช้รีจิสเตอร์ควบคุมแอดทริบิวต์ (Attribute Control Register) เป็นตัวกำหนดหน้าที่ของบิต D7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอตทริบิวต์ข้อความโมโนโครม

ตารางที่ 4.9 แสดงไบต์แอตทริบิวต์ข้อความในการแสดงผลของจอโมโนโครม ซึ่งจะคล้ายกับการกำหนดแอตทริบิวต์ข้อความสี โดยบิต D0-D2 ใช้ในการควบคุมลักษณะการแสดงผลตัวอักษร เช่น แสดงตัวอักษรตามปกติ หรือขีดเส้นใต้ หรือกะพริบ บิต D3 ใช้กำหนดความสว่างของตัวอักษร บิต D4-D6 ใช้ในการสลับสีพื้นกับสีตัวอักษร (Reverse Video) การกำหนดบิต D7 และ D3 จะเหมือนกับการกำหนดแอตทริบิวต์ข้อความสีมาตรฐาน

ตารางที่ 4.8 แอตทริบิวต์ข้อความโมโนโครม

แอตทริบิวต์การแสดงผลแบบโมโนโครม	
00000000	ไม่แสดงอักษร
00000111	อักษรปกติ
10000111	อักษรกะพริบ
00001111	อักษรสว่าง
10001111	อักษรสว่างและกะพริบ
00000001	อักษรพร้อมเส้นใต้
10000001	อักษรกะพริบพร้อมเส้นใต้
00001001	อักษรสว่างพร้อมเส้นใต้
10001001	อักษรสว่างและกะพริบพร้อมเส้นใต้
01110000	กลับการแสดงผลภาพ (สลับการแสดงผลสีดำและขาว)
11110000	กลับการแสดงผลภาพกะพริบ (สลับการแสดงผลสีดำและขาว)

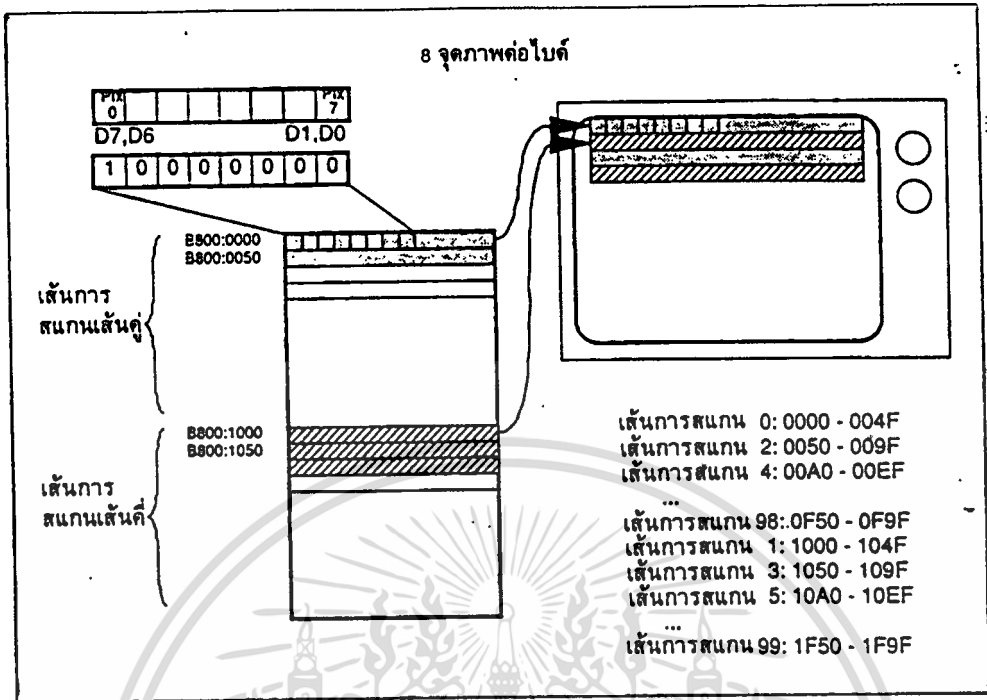
แอตทริบิวต์ข้อความของผู้ใช้

เราสามารถทำการโปรแกรมการทำงานตัวควบคุมแอตทริบิวต์ใหม่ เพื่อให้แสดงลักษณะข้อความได้ตามต้องการ

หน่วยความจำแสดงผลโมโนโครมกราฟฟิก

โมด 6 (กราฟฟิก 2 สี แบบ CGA)

โมด 6 เป็นโมดที่ให้ความละเอียดสูงสุดตามแบบของการ์ด CGA 640 x 200 จุด โดยใช้ 1 บิตต่อ 1 จุดภาพ หรือ 8 จุดภาพต่อ 1 ไบต์ ถ้าบิตมีค่าเป็น 0 จะให้จุดสีดำ ถ้าบิตมีค่าเป็น 1 จะให้จุดสีขาว โดยข้อมูลของจุดภาพจะถูกเก็บไว้ในเทเลน 0 ข้อมูลในการแสดงผลจะถูกส่งออกมาแบบอนุกรม โดยส่งบิต D7 ออกไปก่อน ให้ไปแสดงที่มุมบนซ้ายของจอ (ดูรูปที่ 4.14)



รูปที่ 4.14 การแมปหน่วยความจำในการแสดงกราฟิกแบบ CGA โหมด 6

ข้อจำกัดของไอซีควบคุม CRT 6845 ซึ่งใช้ในการ์ด CGA จะทำให้การแมปแอดเดรสของหน่วยความจำแสดงผลไม่เป็นเชิงเส้น ซึ่งทำให้อัลกอริทึมในการวาดรูปซับซ้อนขึ้น เนื่องจากการคำนวณจะต้องทราบถึงตำแหน่งของจุดบนจอภาพและตำแหน่งบิตในหน่วยความจำแสดงผล โดยครึ่งหนึ่งของหน่วยความจำแสดงผลจะเก็บข้อมูลเส้นการสแกนเส้นคี่ และอีกครึ่งหนึ่งจะเก็บข้อมูลเส้นการสแกนเส้นคู่ เราสามารถแปลงตำแหน่งของจุด (X,Y) บนจอภาพให้เป็นตำแหน่งบิตในหน่วยความจำแสดงผลได้ (X คือ ตำแหน่งในแนวอนมีค่าตั้งแต่ 0-639 และ Y คือ ตำแหน่งในแนวตั้งมีค่า 0-199) โดยใช้สูตร

- ถ้า Y เป็นเลขคู่ ไบต์แอดเดรส = $80 * (Y/2) + (X/8)$
- ถ้า Y เป็นเลขคี่ ไบต์แอดเดรส = $4096 + 80 * ((Y-1)/2) + (X/8)$
- ตำแหน่งบิต $(0-7) = 7 - (X \text{ modulo } 8)$

โดยที่ Modulo คือ ค่าเศษที่เกิดจาก $X/8$

หมายเหตุ ไบต์แอดเดรสที่คำนวณได้จะมีค่าเป็นฐานสิบ

โหมด 4 และ โหมด 5 (กราฟิก 4 สี แบบ CGA)

โหมดนี้นิยมใช้ในการแสดงผลแบบ CGA โดยแสดงสีได้มากที่สุด แต่มีความละเอียดต่ำ 320 x 200 จุด การเก็บข้อมูลจุดภาพในหน่วยความจำแสดงผลจะใช้แบบอัดข้อมูลจุดภาพ โดยใช้ 2 บิตต่อจุด

ภาพ หรือ 4 จุดภาพต่อไบต์ ข้อมูลจุดภาพจะถูกเก็บไว้ที่เทเลน 0 และจะส่งข้อมูลออกแบบอนุกรม โดยส่งบิต D7 และ D6 ของไบต์ 0 ไปแสดงที่มุมบนซ้ายของจอ

ในโหมดนี้การแมปหน่วยความจำแสดงผลจะเหมือนกับโหมดกราฟฟิกแบบ CGA ทั่วไป กล่าวคือ มีความไม่เป็นเชิงเส้นในการแมปแอดเดรสไปยังหน่วยความจำแสดงผล การคำนวณแอดเดรสที่แท้จริงจะต้องทราบถึงตำแหน่งของจุดบนจอภาพและตำแหน่งบิตในหน่วยความจำแสดงผล โดยครึ่งหนึ่งของหน่วยความจำแสดงผลจะใช้ในการเก็บข้อมูลของเส้นการสแกนเส้นคู่ และอีกครึ่งหนึ่งของหน่วยความจำแสดงผลจะใช้ในการเก็บข้อมูลเส้นการสแกนเส้นคู่

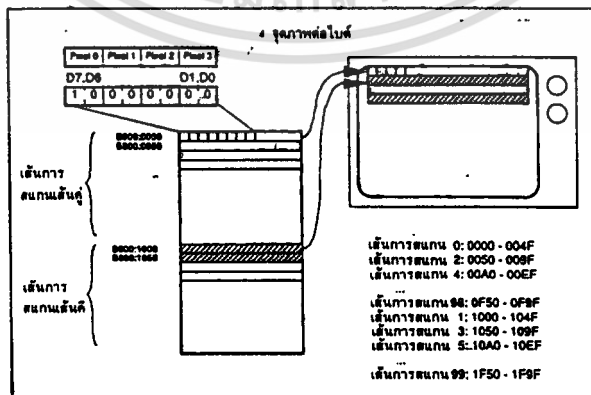
ในการแปลงตำแหน่งจุด (X,Y) บนจอไปยังตำแหน่งบิตบนหน่วยความจำแสดงผล (X = 0-320, Y=0-199) หาได้จากสูตร

- ถ้า Y เป็นเลขคู่ ไบต์แอดเดรส = $80 * (Y/2) + (X/4)$
- ถ้า Y เป็นเลขคี่ ไบต์แอดเดรส = $4096 + 80 * ((Y-1)/2) + (X/4)$
- ตำแหน่งบิต (0,2,4,6) = $(X \text{ modulo } 4) * 2$

สีมาตรฐานที่ใช้ในโหมด 4 และ 5 มี 2 ชุด โดยจะใช้ฟังก์ชันไบออส 11 ในการเลือกสี ซึ่งแสดงไว้ในตารางที่ 4.9

ตารางที่ 4.9 สีมาตรฐาน-โหมด 4 และ 5

ค่าจุดภาพ	สีมาตรฐาน	สีพิเศษ
00	Black	Black
01	Light Cyan	Green
10	Light Magenta	Red
11	Intensified White	Brown



รูปที่ 4.15 การแมปหน่วยความจำในการแสดงกราฟฟิกแบบ CGA โหมด 4 และ 5

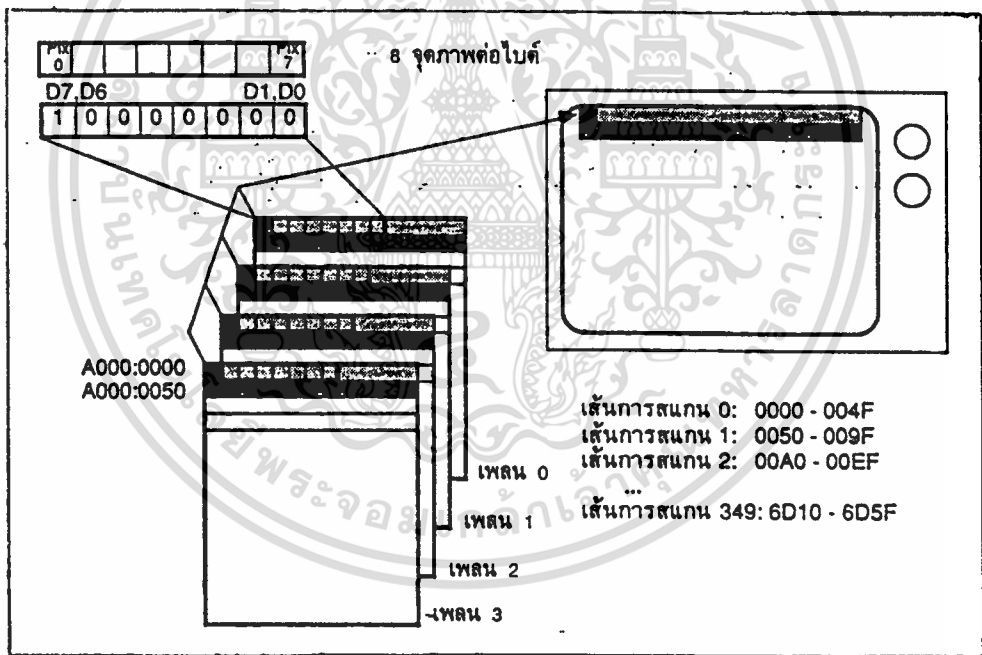
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด F (กราฟฟิกโมโนโครม)

โหมดนี้ทำงานได้เฉพาะกับการ์ด EGA และ VGA และไม่มีปัญหาความไม่เป็นเชิงเส้นของการแมปแอดเดรสไปยังหน่วยความจำแสดงผล ซึ่งพบในการทำงานแบบ CGA โหมด F ให้ความละเอียดภาพ 640 x 350 จุด และใช้สองเพลนสี (เพลน 0 และ เพลน 1) โดยข้อมูลแต่ละจุดภาพจะใช้เพียงหนึ่งบิตของแต่ละเพลนรวมกัน สามารถแสดงสีได้ 4 สี คือ ดำ, ขาว, ขาวสว่าง และกะทริบ ในการเขียนข้อมูลลงเพลนทั้งสองจะเป็นอิสระต่อกัน และควบคุมการเขียนได้โดยใช้รีจิสเตอร์ควบคุมการเขียน (write enable register) ในส่วนซีแควนเซอร์

รูปที่ 4.16 แสดงการแมปหน่วยความจำในโหมด F ในการแปลงตำแหน่งของจุดภาพ (X,Y) บนจอไปเป็นตำแหน่งบิตบนหน่วยความจำส่วนแสดงผล (X มีค่าได้ตั้งแต่ 0-639 และ Y มีค่าได้ตั้งแต่ 0-349) สามารถหาได้จาก

- ไบต์แอดเดรส = $Y * 80 + X/8$
- ตำแหน่งบิต (0-7) = $7 - (X \text{ Modulo } 8)$



รูปที่ 4.16 การแมปหน่วยความจำในการแสดงกราฟฟิกโหมด F

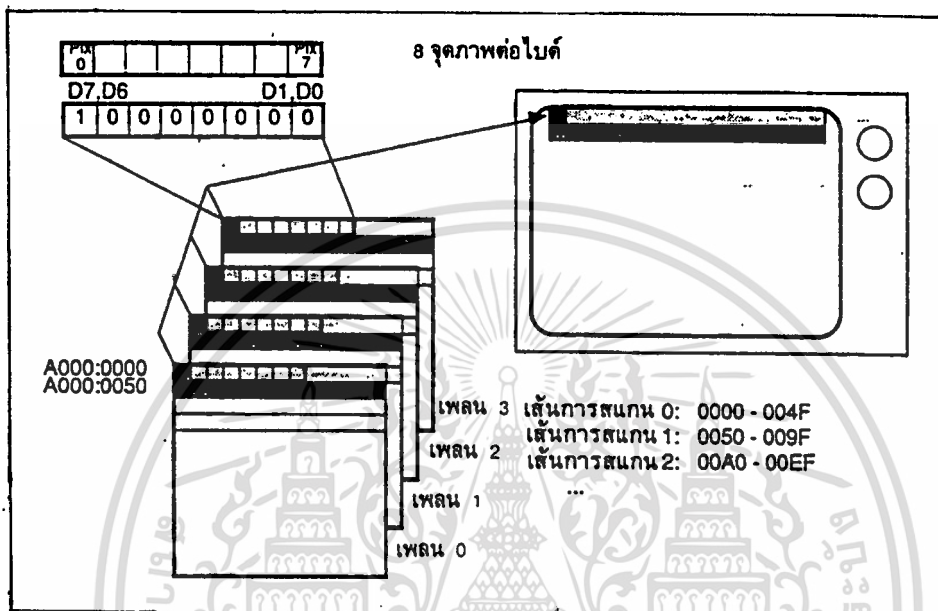
โหมด 10 (กราฟฟิกสีแบบ Enhanced)

โหมดนี้ใช้กับการ์ด VGA และ EGA เท่านั้น และเป็นโหมดที่นิยมใช้ในการแสดงกราฟฟิกในปัจจุบัน ให้ความละเอียด 640 x 350 จุด และใช้เพลนสี 4 เพลน ซึ่งแต่ละเพลนทำงานได้อย่างเป็นอิสระ และสามารถควบคุมโดยใช้รีจิสเตอร์ควบคุมการเขียนในส่วนซีแควนเซอร์ ข้อมูลจุดภาพแต่ละจุดจะใช้เพียงหนึ่งบิตของแต่ละเพลนรวมกัน เป็นผลให้แสดงสีต่างกันได้ถึง 2^4 สี หรือ 16 สี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.17 แสดงการแมปหน่วยความจำในโหมด 10 HEX การแปลงตำแหน่งจุดภาพบนจอไปเป็นตำแหน่งบิตบนหน่วยความจำแสดงหาได้จาก

- ไบต์แอดเดรส = $Y * 80 + X/8$
- ตำแหน่งบิต (0-7) = $7 - (X \text{ modulo } 8)$



รูปที่ 4.17 การแมปหน่วยความจำในการแสดงกราฟฟิกโหมด F

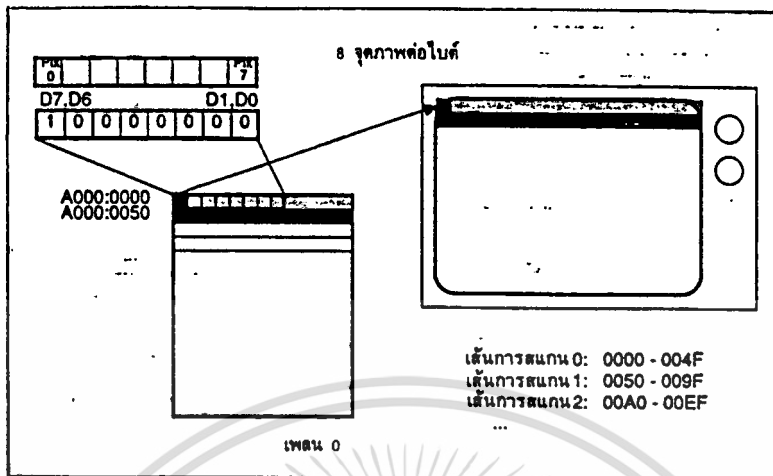
โหมด D และ E (กราฟฟิก 16 สี)

การทำงานในโหมดนี้จะคล้ายกับการทำงานในโหมด 10 HEX ต่างกันเพียงความละเอียดในการแสดงผล โดยโหมด D จะให้ความละเอียด 320 x 200 จุด ส่วนในโหมด E จะให้ความละเอียด 640 x 200 จุด ทำให้ทั้งสองโหมดนี้ไม่เป็นที่นิยมใช้ เนื่องจากข้อจำกัดด้านความละเอียดในการแสดงผล แต่อย่างไรก็ตาม โหมดนี้เป็นโหมดกราฟฟิกแบบ CGA ที่แสดงสีได้ถึง 16 สีบนจอสีมาตรฐาน

โหมด 11 HEX (กราฟฟิก 2 สี)

โหมดนี้ใช้ได้กับการ์ด VGA เท่านั้น มีความละเอียด 640 x 480 จุด และแสดงสีได้ 2 สี โดยเก็บข้อมูลการแสดงผลไว้ที่แผ่น 0 ส่วนแผ่นอื่นๆ ไม่ได้นำไปใช้ แต่แต่ละจุดภาพจะใช้เพียง 1 บิตของแผ่น และการแมปข้อมูลไปยังหน่วยความจำแสดงผลมีความเป็นเชิงเส้น (ดังรูปที่ 4.21) การแปลงตำแหน่งจุดภาพบนจอไปเป็นตำแหน่งบิตบนหน่วยความจำแสดงผลสามารถหาได้จาก

- ไบต์แอดเดรส = $(Y * 80) + (X/8)$
- ตำแหน่งบิต (0-7) = $7 - (X \text{ modulo } 8)$



รูปที่ 4.18 การแมปหน่วยความจำในการแสดงกราฟฟิกโหมด 11

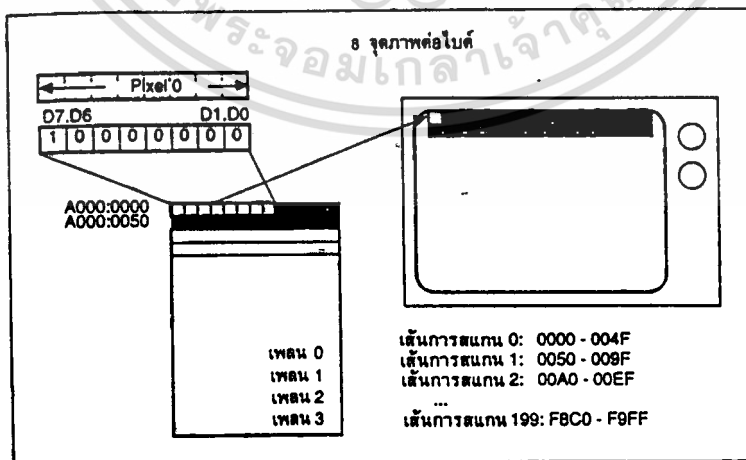
โหมด 12 HEX (กราฟฟิก 16 สี)

โหมด 12 เป็นโหมดที่ใช้เฉพาะกับ VGA การทำงานคล้ายกับโหมด 10 แต่มีความละเอียด 640 x 480 จุด และใช้เพนสี 4 เพน แสดงสีได้ 16 สี การจัดการหน่วยความจำแสดงดังรูปที่ 4.17

โหมด 13 HEX (กราฟฟิก 256 สี)

โหมดนี้ใช้ได้เฉพาะกับการ์ด VGA สามารถแสดงสีได้ 256 สี มีความละเอียด 320 x 200 จุด รูปแบบการจัดการหน่วยความจำแสดงดังรูปที่ 4.19 การแปลงตำแหน่งจุดภาพ (X,Y) บนจอไปเป็นตำแหน่งบิตของหน่วยความจำแสดงผลหาได้จากสมการ

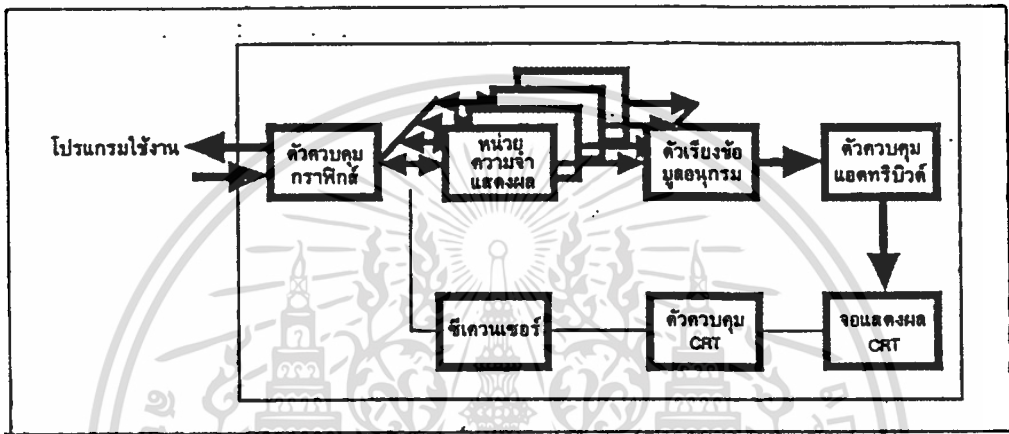
- ไบต์แอดเดรส = $(Y * 320) + X$



รูปที่ 4.19 การแมปหน่วยความจำในการแสดงกราฟฟิกโหมด 12

4.4.6 ตัวควบคุมกราฟฟิก

ตัวควบคุมกราฟฟิก (Graphics Controller) เป็นชิป VLSI ซึ่งอยู่บนทางเดินข้อมูลระหว่างตัวประมวลผลกับหน่วยความจำแสดงผล (ดังรูปที่ 4.20) ปกติตัวควบคุมกราฟฟิกจะไม่ทำอะไรกับข้อมูลที่อ่านหรือเขียนกับหน่วยความจำแสดงผล ตัวประมวลผลหลักสามารถโปรแกรมตัวควบคุมกราฟฟิกให้ช่วยในการวาดรูปตามต้องการได้



รูปที่ 4.20 ตัวควบคุมกราฟฟิก

แลตซ์การอ่าน

ทุกครั้งที่หน่วยประมวลผลหลักอ่านข้อมูลจากหน่วยความจำแสดงผลจะทำการค้างข้อมูลไว้ที่แลตซ์การอ่าน (Processor read latches) บนการ์ด EGA ด้วย ในระหว่างไจเคิลการเขียนข้อมูลในแลตซ์การอ่านอาจจะนำมากระทำทางตรรกะกับข้อมูลจากหน่วยประมวลผลหลักได้ การทำงานเช่นนี้สามารถช่วยหน่วยประมวลผลกำหนดรูปแบบการวาดรูป ในขณะที่หน่วยประมวลผลอ่านข้อมูลครั้งละเฟรมจะมีการค้างข้อมูลไว้ที่แลตซ์การอ่านทั้ง 4 เฟรมพร้อมกัน ซึ่งจะทำให้สามารถย้ายข้อมูลจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่งบนหน่วยความจำแสดงผลได้อย่างรวดเร็ว

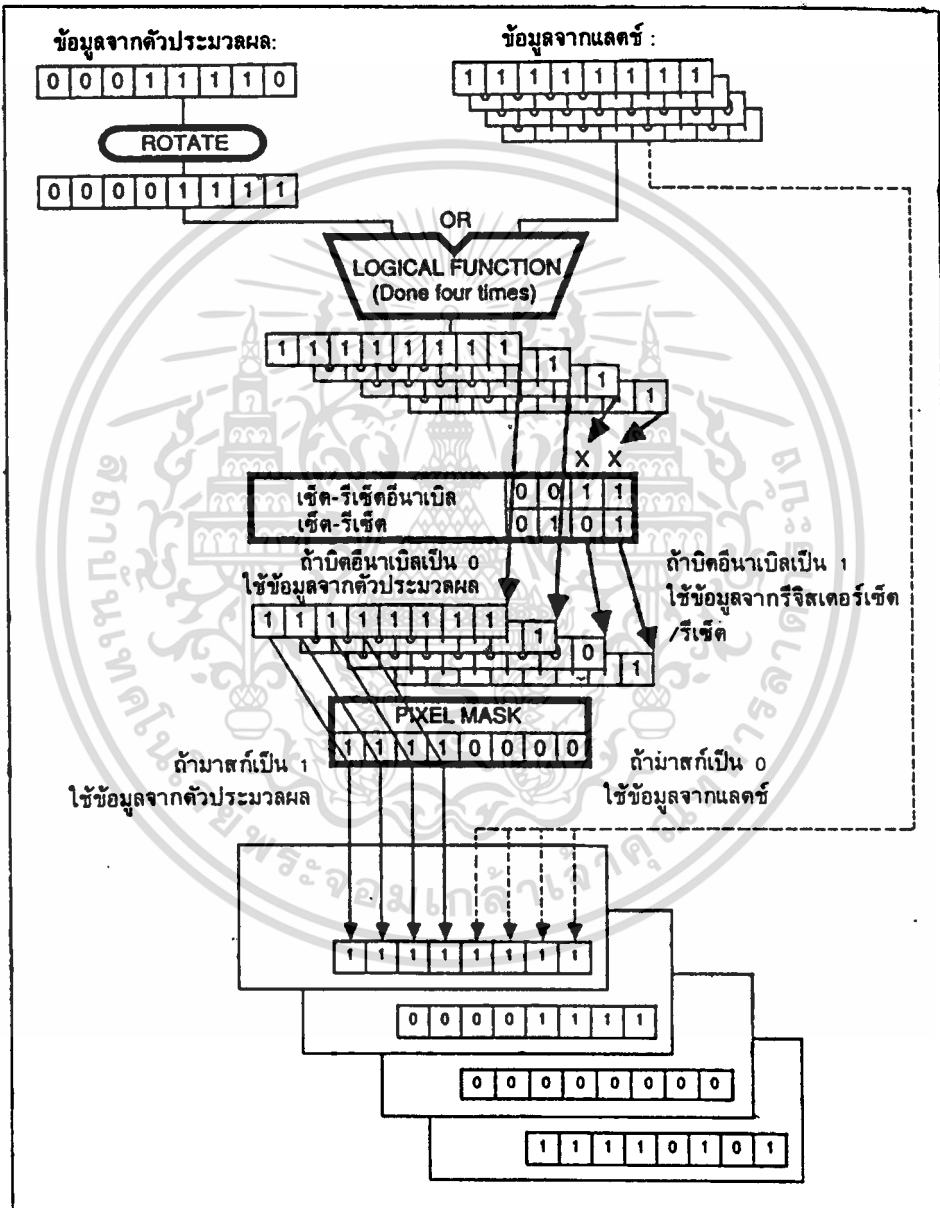
ตัวกระทำทางตรรกะ

ในระหว่างไจเคิลการเขียน ตัวควบคุมกราฟฟิกจะใช้ฟังก์ชันตรรกะ (Logical Unit) ในการกำหนดรูปแบบของการเขียนข้อมูล ดังนี้

- ไม่เปลี่ยนแปลงข้อมูล
- ข้อมูลในแลตซ์กับข้อมูลจากตัวประมวลผล
- ข้อมูลในแลตซ์กับข้อมูลจากตัวประมวลผล
- ข้อมูลในแลตซ์กับข้อมูลจากตัวประมวลผล
- หมุนวน (Rotate) ข้อมูลจากตัวประมวลผล

ฟังก์ชันทางตรรกะ AND, OR และ XOR จะเป็นประโยชน์ในการรวมและเคลื่อนย้ายวัตถุที่จะแสดงบนจอภาพ

ฟังก์ชันการหมุนวนข้อมูลจะใช้ในการย้ายข้อมูลที่ไม่ครบไบต์เป็นบล็อก การทำงานของตัวควบคุมกราฟฟิกในระหว่างการเขียน แสดงดังรูปที่ 4.21



รูปที่ 4.21 การทำงานของตัวควบคุมกราฟฟิก

การเปรียบเทียบสี

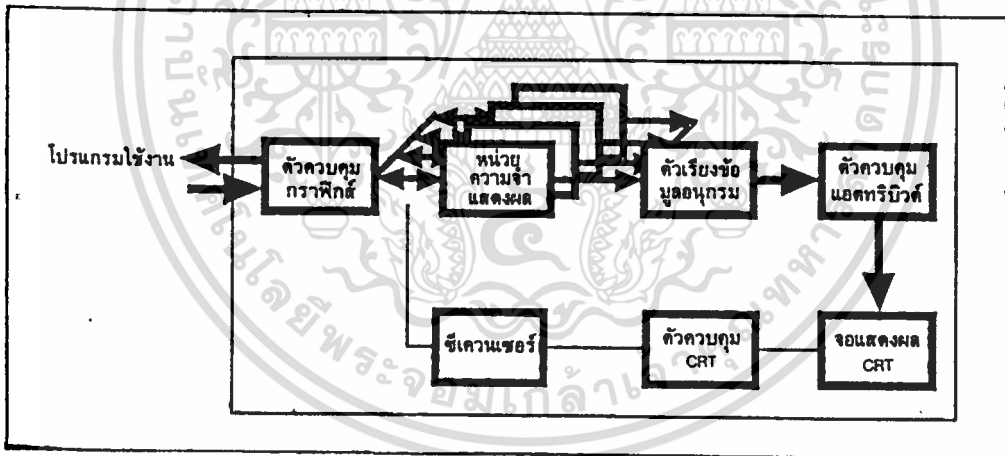
ในช่วงไซเคิลการอ่าน ตัวควบคุมกราฟฟิกสามารถใช้ฟังก์ชันการเปรียบเทียบสี (Color compare) ซึ่งจะเป็นประโยชน์มากสำหรับอัลกอริทึมการวาดรูป เช่น FLOOD FILL ซึ่งจะคอยตรวจสอบสีที่จะแสดงบนจอภาพหรือการเปลี่ยนแปลงสีในช่วงไซเคิล การอ่านหน่วยความจำแสดงผลตัวประมวลผลสามารถตรวจสอบได้ครั้งละ 1 เฟรมสี โดยตัวประมวลผลจะทำการเปรียบเทียบข้อมูลทั้ง 4 เฟรมกับสีอ้างอิง และจะแสดงผลการเปรียบเทียบออกมา

เราสามารถใส่ฟังก์ชันเปรียบเทียบสีคันทาวิตูในหน่วยความจำแสดงผลตามสปีทรีบุ โดยใช้คำสั่ง REP SCASB ใน 8086

ตัวเรียงข้อมูลอนุกรม

ตัวเรียงข้อมูลอนุกรม เป็นส่วนของตัวควบคุมกราฟฟิกที่ใช้เรียงข้อมูลที่อ่านจากหน่วยความจำแสดงผลให้อยู่ในรูปอนุกรมเพื่อส่งผ่านไปยังจอ CRT โดยจะส่งบิตที่มีนัยสำคัญสูงสุด (MSB บิต) เป็นบิตแรก

ตัวควบคุมแอสทรีบิวต์

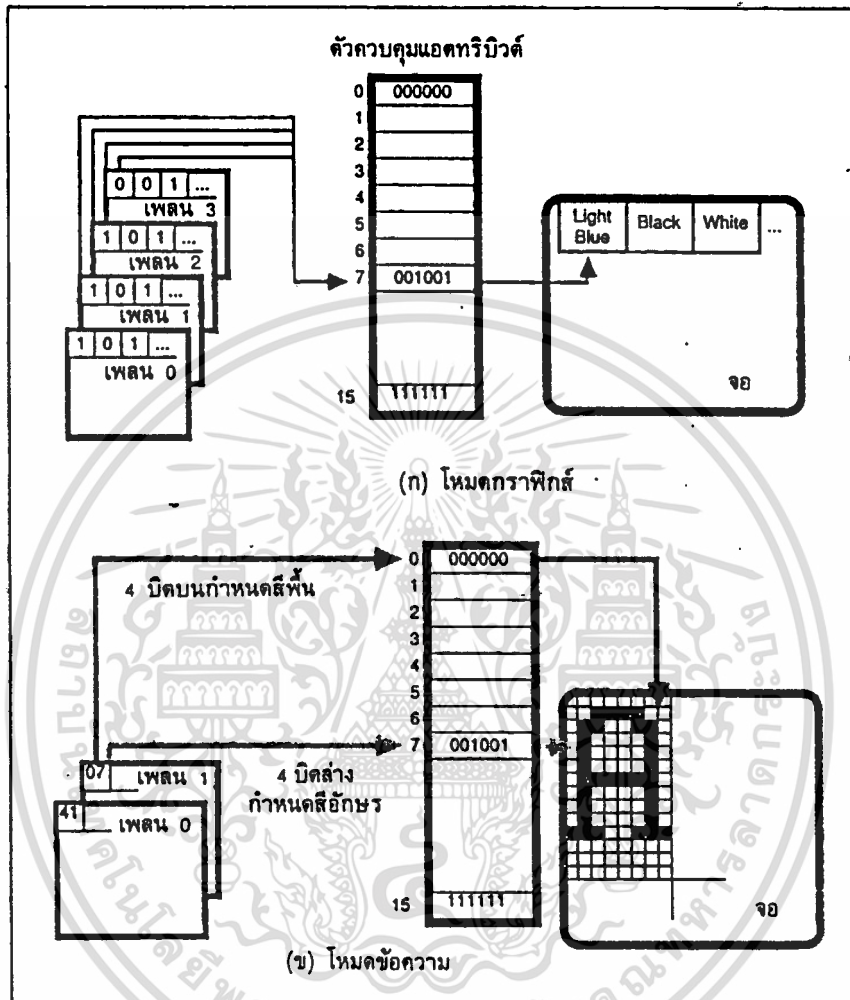


รูปที่ 4.22 ตัวควบคุมแอสทรีบิวต์

ตัวควบคุมแอสทรีบิวต์ (Attribute Controller) ทำหน้าที่กำหนดสีในการแสดงผลโหมดข้อความและโหมดกราฟฟิก โดยจะใช้ตารางค้นหาสีทำการแปลงข้อมูลขนาด 4 บิตบนหน่วยความจำแสดงผลเป็น 6 บิต ในการแสดงผลแบบ EGA หรือ 12 บิต ในการแสดงผลแบบ VGA

ตารางค้นหาสีจะกำหนดสีเริ่มต้นให้เหมาะสมกับโหมดการทำงานที่ใช้ ในโหมดโมโนโครมตารางค้นหาสีจะกำหนดสีเริ่มต้น 2 สี ส่วนในโหมด CGA ตารางค้นหาสีจะกำหนดจำนวนสีเริ่มต้นให้เหมาะสมกับการทำงานในโหมดนั้น สำหรับโหมด EGA และ VGA ตารางค้นหาสีจะสนับสนุนการแสดงผลได้

มากขึ้น โดยสามารถโปรแกรมเลือกชุดสีที่จะใช้แสดงผลจากสีทั้งหมดที่สามารถแสดงได้ และสามารถโปรแกรมตารางค้นหาสีใหม่ได้ตลอดเวลา



รูปที่ 4.23 การทำงานของตารางค้นหาสีในตัวควบคุมแอดทรีบิวต์

รูปที่ 4.23 แสดงการทำงานของตารางค้นหาสีในตัวควบคุมแอดทรีบิวต์ระหว่างไซเคิล การรีเฟรชภาพ พิจารณาตารางรูปที่ 4.23 ชุดสีที่มีค่า 0111B(7) บนเพลนสี ซึ่งค่าของสีนี้จะใช้เป็นแอดเดรสสำหรับเลือกกรีจิสเตอร์ของตารางค้นหาสี โดยในที่นี้จะเลือกกรีจิสเตอร์ 7 ในตารางค้นหาสี ซึ่งกรีจิสเตอร์นี้จะเก็บค่า 001001B เพื่อใช้ในการแสดงสี (ซึ่งหมายถึง สีน้ำเงินสว่าง ในโหมด 10)

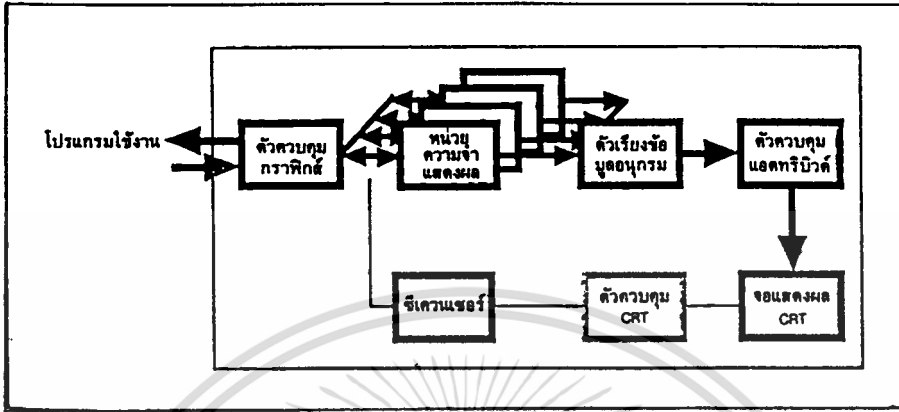
หึ่งสังเกตว่า การกำหนดแอดทรีบิวต์สีในโหมดข้อความจะต่างกับโหมดกราฟฟิก

ตัวควบคุม CRT

รีจิสเตอร์ส่วนใหญ่ของตัวควบคุม CRT (CRT Controller) ใช้ในการสร้างสัญญาณควบคุม การกวาดภาพ โดยรีจิสเตอร์เหล่านี้จะถูกกำหนดค่าเริ่มต้นระหว่างการเลือกโหมดการแสดงผลและชนิดจอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยไม่สนใจการกำหนดในตัวโปรแกรม (ดังรูปที่ 4.24) นอกจากนี้การแก้ไขค่าของรีจิสเตอร์เหล่านี้ อาจก่อให้เกิดความคิดพลาดของสัญญาณเวลาควบคุม CRT และทำให้เกิดความเสียหายแก่จอภาพ



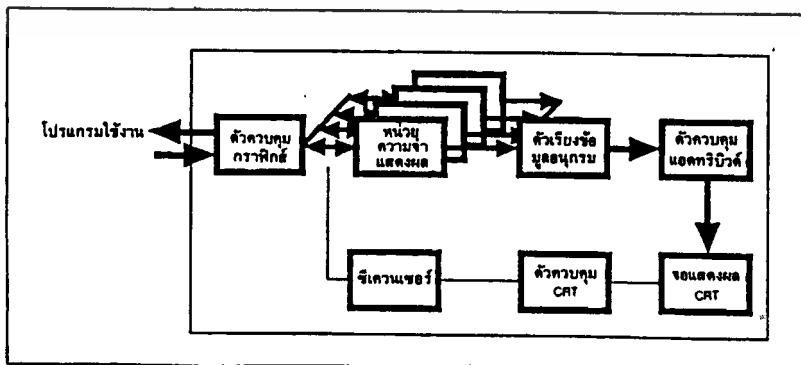
รูปที่ 4.24 ตัวควบคุม CRT

นอกจากนี้ยังมีรีจิสเตอร์อีกกลุ่มในตัวควบคุม CRT ซึ่งใช้ในการกำหนดรูปแบบการแสดงผล ข้อมูล ตัวแปรเหล่านี้ ได้แก่ จำนวนจุดต่อเส้นการสแกน , จำนวนเส้นการสแกน , ความสูงตัวอักษร , ตำแหน่งเริ่มต้นของหน่วยความจำแสดงผลสำหรับการรีเฟรชภาพ เป็นต้น รีจิสเตอร์เหล่านี้จะถูกกำหนดค่าเริ่มต้นให้สอดคล้องกับโหมดการทำงาน เราไม่ควรเข้าไปแก้ไขค่าของรีจิสเตอร์เหล่านี้

ส่วนรีจิสเตอร์ที่ใช้ในการกำหนดรูปร่างและตำแหน่งเคอร์เซอร์ รีจิสเตอร์สนับสนุนการใช้ปากกาแสง (Light pen) และรีจิสเตอร์สำหรับการเลื่อนบรรทัดในแนวตั้ง รีจิสเตอร์เหล่านี้จะอยู่ในไบออสของการ์ดแสดงผล

ซีควเอนเซอร์

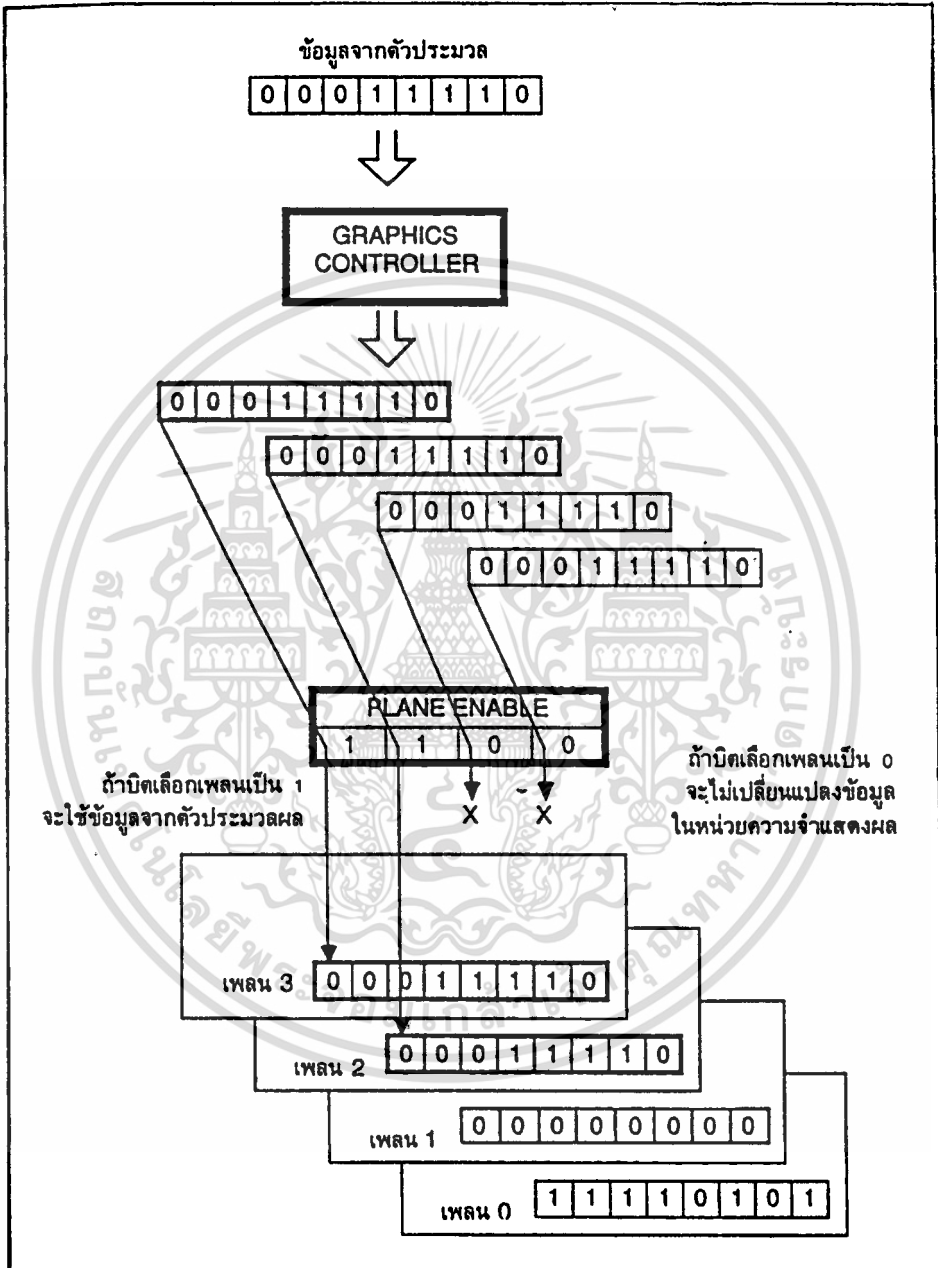
ซีควเอนเซอร์ (Sequencer) มีหน้าที่ควบคุมลำดับการทำงานของการ์ดแสดงผล (ดังรูปที่ 4.29) โดยสร้างสัญญาณเวลาสำหรับแสดงจุดภาพและตัวอักษรเพื่อใช้ในการรีเฟรชภาพ และยังใช้ควบคุมไฟเกิดการอ่านหรือเขียนหน่วยความจำแสดงผล ในบางครั้งยังใช้ในการสร้างสัญญาณสถานะรอ (wait state)



รูปที่ 4.25 ซีควเอนเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ ซีควนเซอร์ยังใช้สำหรับการเลือกเพลนสี ซึ่งเป็นประโยชน์ต่อผู้เขียนโปรแกรมใช้ด้วย (ดังรูปที่ 4.26)



รูปที่ 4.26 การเลือกเพลนสีของซีควนเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

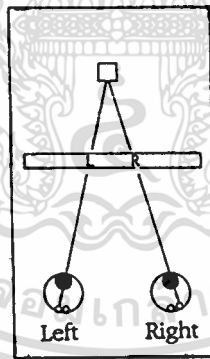
หลักการทํางานของแว่นตา LCD และรูปแบบของภาพเคลื่อนไหวและภาพนิ่ง

5.1 หลักการทํางานของแว่นตา LCD Shutter

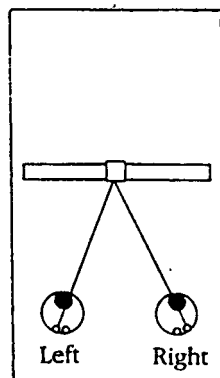
ในการสร้างภาพ 3 มิติจริง ๆ สิ่งที่สำคัญคือ การมองเห็นของภาพที่ตาทั้งสอง ซึ่งตาแต่ละข้างจะได้เห็นรูปภาพในส่วนองภาพของมันเอง ก็คือตาซ้ายจะเห็นเฉพาะภาพของตาซ้ายเท่านั้น และตาขวาจะเห็นเฉพาะภาพของตาขวาเท่านั้น ดังนั้นเมื่อตาซ้ายและตาขวาเห็นภาพดังกล่าวแล้วสมองจะทำให้เกิดความรู้สึกที่มีมิติจริง ๆ ออกมา

ดังนั้นแว่นตา LCD Shutter จะต้องทำหน้าที่ปิดตาข้างใดข้างหนึ่ง โดยเมื่อภาพของตาซ้ายถูกแสดงออกที่จอภาพ ตาซ้ายจะสามารถมองเห็นภาพได้ แต่ตาขวาดังถูกปิดกั้นไว้เพื่อไม่ให้มองเห็น โดยจะใช้ LCD เป็นตัวทำหน้าที่ปิดตาเพื่อไม่ให้ตาขวามองเห็นภาพของตาซ้ายได้ และเช่นกันเมื่อภาพของตาขวาถูกแสดงออกที่จอภาพ ตาขวาจะสามารถมองเห็นภาพได้ แต่ตาซ้ายต้องถูกปิดกั้นไว้เพื่อไม่ให้มองเห็น ดังนั้นแว่นตา LCD จะทำหน้าที่ เปิด / ปิด ให้ตามองเห็นภาพของตาแต่ละข้างอย่างถูกต้อง

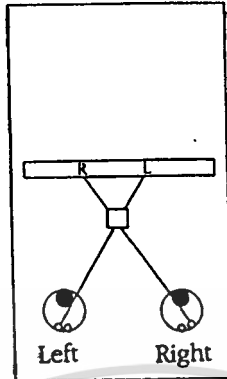
การเกิดภาพ 3 มิติบนหน้าจอจะมีทั้งถักกลงไปในจอซึ่งเรียกว่า Negative Parallax ถ้าอยู่บนจอภาพพอดีเรียกว่า Zero Parallax และภาพที่ลอยออกมาหน้าจอจะเรียกว่า Positive Parallax ซึ่งส่วนสำคัญจะอยู่ที่ Negative Parallax และ Positive Parallax



รูปที่ 5.1 แสดงถึง Negative Parallax



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 5.2 แสดงถึง Zero Parallax นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

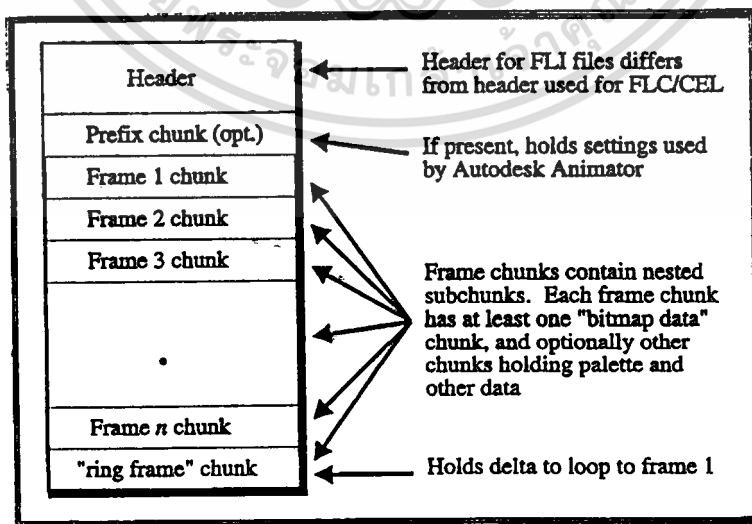


รูปที่ 5.3 แสดงถึง Positive Parallax

การสร้างภาพ 3 มิติอีกวิธีคือ จะอาศัยการ scan ของหน้าจอ คือให้ภาพของตาซ้ายแสดงเฉพาะเส้น scan ที่ และภาพของตาขวาแสดงเฉพาะเส้น scan คู่ แต่วิธีการนี้ทำได้เฉพาะในโหมดการแสดงผลแบบความละเอียดสูงเท่านั้น คือมีการแยก scan เส้นที่และคู่ออกจากกัน โดยครั้งแรกจะ scan เส้นที่ และจะทำการ refresh แล้วทำการ scan เส้นคู่ต่อและทำการ refresh เป็นเช่นนี้ไปเรื่อย ๆ ดังนั้นจึงนำเอาจังหวะ scan เส้นที่มาปิด LCD ของตาขวา และ จังหวะ scan เส้นคู่มาปิด LCD ของตาซ้าย

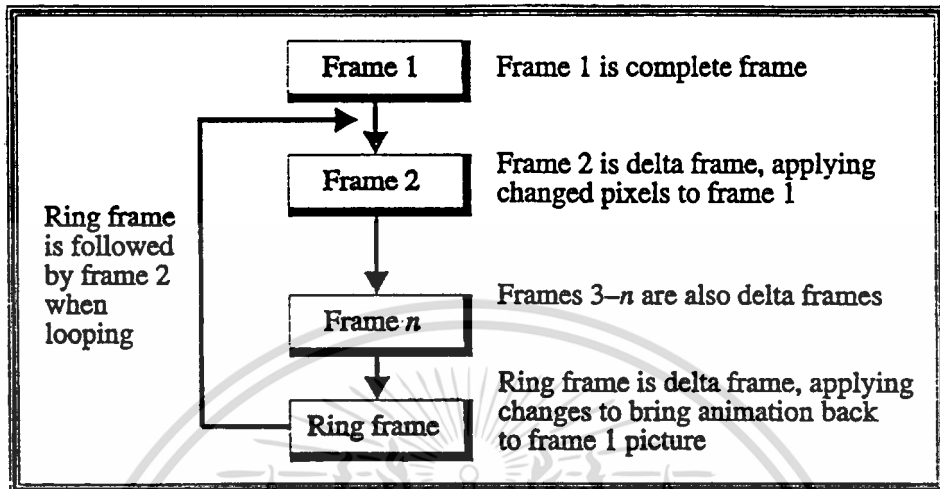
5.2 การทำงานของไฟล์ FLI

ฟอร์แมตไฟล์ FLI นี้เป็นของบริษัท AUTODESK โดยภาพ FLI จะมีขนาดที่แสดงที่หน้าจอได้เฉพาะขนาด 320 x 200 ที่ 256 สี โดยมีรูปแบบการจัดเก็บข้อมูลดังนี้



รูปที่ 5.4 แสดงถึงโครงสร้างของ FLI file

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 แสดงถึงการใช้ Delta ring frame ในการกลับไปเริ่มต้นที่ frame ที่ 1

5.2.1 HEADER ของรูปแบบ FLI

ไบนารีที่	ขนาด	ชื่อ	คำอธิบาย
0	4	size	ขนาดของ file นี้ทั้งหมด
4	2	magic	มีค่าเป็น AF 11 ฐาน 16 สำหรับตรวจสอบฟอร์แมต FLI
6	2	frames	จำนวนเฟรมของ FLI สูงสุดที่ 4000 เฟรม
8	2	width	ความกว้างของหน้าจอแสดงผลมีค่าเท่ากับ 320
10	2	height	ความสูงของหน้าจอแสดงผลมีค่าเท่ากับ 200
12	2	depth	จำนวนบิตต่อหนึ่งจุด (pixel) มีค่าเท่ากับ 8
14	2	flags	มีค่าเป็น 0
16	2	speed	ความเร็วของการแสดงภาพในแต่ละเฟรม มีหน่วยเป็น 1/70 วินาที
18	4	next	ชี้ค่าเท่ากับ 0
22	4	frit	ชี้ค่าเท่ากับ 0
26	102	expand	ที่เหลืออีก 102 ไบนารีที่มีค่าเป็น 0 เพื่อสำหรับเพิ่มเติมในอนาคต

ตารางที่ 5.1 FLI Header ขนาด 128 ไบนารี

ไบนารี	ขนาด	ชื่อ	คำอธิบาย
0	4	size	ขนาดของข้อมูลที่มีการเปลี่ยนแปลงในแต่ละเฟรม ขนาดไม่เกิน 64K
4	2	magic	ค่าตรวจสอบมีค่าเป็น F1 FA ฐาน 16
6	2	chunks	จำนวนของชุดข้อมูลในหนึ่งเฟรม
8	8	expand	เผื่อขนาด 8 ไบนารีที่มีค่าเป็น 0 ทั้งหมด

ตารางที่ 5.2 ค่าที่หลังจาก Header ตอนต้นจะเป็นส่วนของ Frame Header

ไบนารี	ขนาด	ชื่อ	คำอธิบาย
0	4	size	ขนาดของแต่ละกลุ่มข้อมูล
4	2	type	รูปแบบของแต่ละกลุ่มข้อมูลมี 5 รูปแบบด้วยกัน

ตารางที่ 5.3 หลังจาก Frame Header จะเป็นส่วน Header ของแต่ละชุดข้อมูล (Chunk Header)

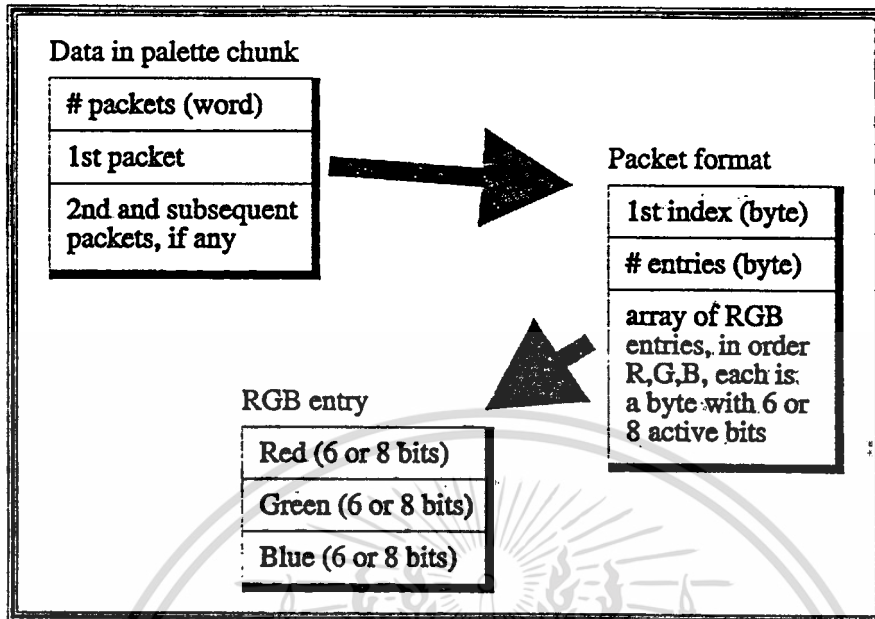
ค่าเป็น	ชื่อ	คำอธิบายของแต่ละรูปแบบ
11	FLI_COLOR	Compressed color map เกี่ยวกับการจัดการของสี
12	FLI_LC	Line compressed เป็นการจัดการของการบีบขนาดข้อมูล ในแต่ละเฟรมยกเว้นเฟรมแรก
13	FLI_BLACK	ถ้าเฟรมแรกเป็นสี 0 หรือสีค่าจะเซตค่าหน้าจอสดำหมด
15	FLI_BRUN	Bytewise run-length compression วิธีบีบขนาดข้อมูลที่ใช้ เฉพาะในเฟรมแรกหรือมีการเปลี่ยนแปลงจากก่อนหน้าไปทั้งเฟรม
16	FLI_COPY	ข้อมูลที่เกิดขึ้นโดยไม่มีการบีบข้อมูลคือเป็นข้อมูลล้วน ๆ ทั้ง 64000 ไบนารี จะใช้ในกรณีที่ต้องการจัดเก็บข้อมูลแบบบีบขนาด แล้วเกิน 64000 ไบนารี และบางเฟรมที่ต้องการความเร็ว

ตารางที่ 5.4 ตารางแสดงค่าของรูปแบบของแต่ละกลุ่มข้อมูล

5.2.2 กลุ่มข้อมูล FLI_COLOR

2 ไบนารีแรกจะเป็นจำนวนชุดของสีที่มีการเปลี่ยนแปลง เช่นสีที่เปลี่ยนเป็นสี 1 - 15 และ 60 - 100 ก็แสดงว่ามี 2 ชุดด้วยกัน ไบนารีต่อไปจะเป็นสีเริ่มต้นที่มีการเปลี่ยนแปลงในแต่ละเฟรม ไบนารีต่อไปจะเป็นจำนวนสีที่เปลี่ยน ถ้าไบนารีนี้มีค่าเป็น 0 แสดงว่ามีการเปลี่ยนแปลงทั้ง 256 สี ไบนารีต่อ ๆ ไปจะเป็นข้อมูลของสี สีละ 3 ค่า คือ แดง เขียว น้ำเงิน จนครบสีทั้งหมดที่มีการเปลี่ยนแปลงใน 1 เฟรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะในรูปแบบใด ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



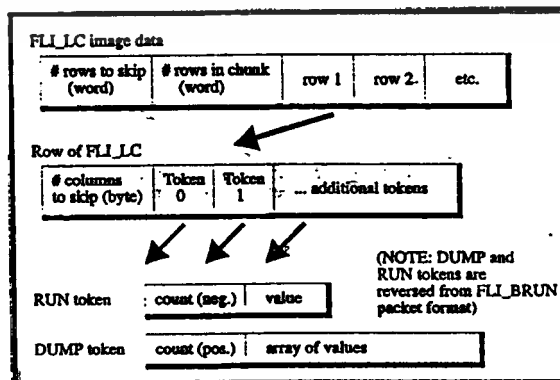
รูปที่ 5.6 Format ของ Palette chunk (FLI_COLOR)

5.2.3 กลุ่มข้อมูล FLI_LC

ถ้าเป็นกลุ่มข้อมูลกลุ่มนี้ 2 ไบท์แรกจะเป็นข้อมูลบอกให้ทราบแถวบนสุดที่มีการเปลี่ยนแปลงจากเฟรมก่อน 2 ไบท์ ถัดไปจะเป็นจำนวนแถวที่มีการเปลี่ยนแปลง โดยในการบีบขนาดข้อมูลจะเป็นในลักษณะทีละ 1 แถว โดยไบท์ต่อ ๆ ไปจะเป็นข้อมูลที่มีรูปแบบดังนี้

ไบท์แรกจะเป็นจำนวนกลุ่มที่มีการเปลี่ยนแปลงใน 1 แถว ถ้าเป็น 0 แสดงว่าทั้งแถวไม่ได้เปลี่ยนจากเฟรมก่อนหน้า จะข้ามไปแถวต่อไป

รูปแบบข้อมูลของแต่ละกลุ่มใน 1 แถวคือ ไบท์แรกจะซึ่งเป็นค่าที่ต้องข้ามจากจุดเริ่มต้นของแต่ละแถวเป็นจำนวนเท่าไร ไบท์ที่ 2 ถ้ามีค่ามากกว่า 0 จะเป็นจำนวนไบท์ที่ต้อง COPY จากไฟล์ FLI ลง MEMORY ไบท์ต่อ ๆ ไปจะเป็นข้อมูล จำนวนไบท์เท่ากับไบท์ที่ 2 ถ้าไบท์ที่ 2 มีค่าน้อยกว่า 0 หรือค่าติดลบ ให้แปลงเป็นตรงข้ามหรือไม่คิดเครื่องหมาย จะเป็นจำนวนไบท์ที่มีค่าซ้ำกัน ไบท์ต่อไปจะเป็นค่าข้อมูลของสีที่ซ้ำกัน



รูปที่ 5.7 Format ของ FLI_LC image chunk

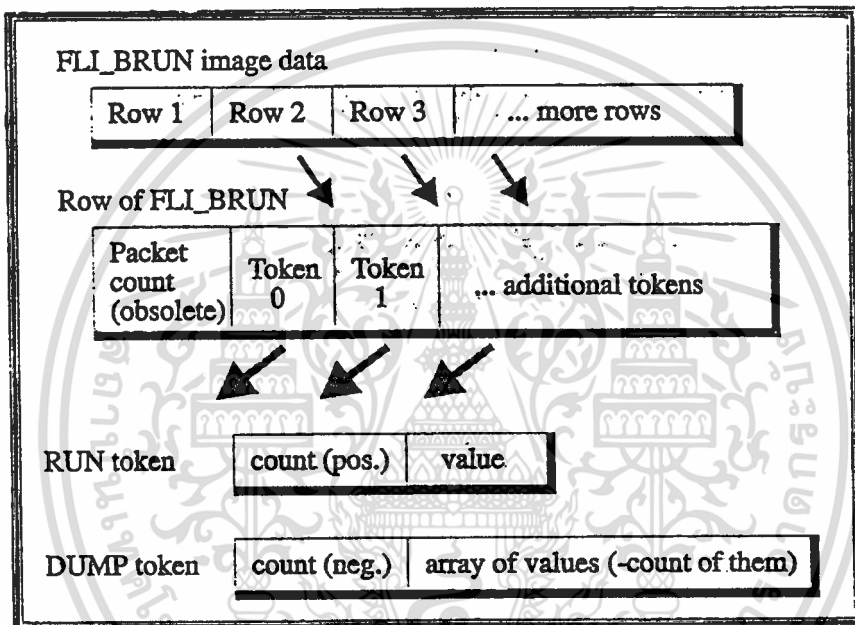
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.4 กลุ่มข้อมูล FLI_BLACK

จะไม่มีข้อมูลอยู่เลย เมื่อรู้ว่าเป็นกลุ่มข้อมูลชนิดนี้จะใช้โปรแกรมจัดการ เคลียร์ MEMORY หรือหน้าจอเป็นสี 0 จะใช้สำหรับในกรณีที่ทั้งหน้าจอเป็นสี 0

5.2.5 กลุ่มข้อมูล FLI_BRUN

คล้ายกับกลุ่มข้อมูล FLI_LC โดยไม่มีการข้ามบางแถวที่ไม่มีการเปลี่ยนแปลง โดยเริ่มด้วยข้อมูลที่แถวแรกถึงแถวสุดท้าย คือ 0 - 199 โดยการจัดเก็บข้อมูลแบบบีบในแต่ละแถวเหมือนกับ FLI_LC แบบไม่มีการข้ามแถว



รูปที่ 5.8 Format ของ FLI_BRUN image chunk

5.2.6 กลุ่มข้อมูล FLI_COPY

จะมีขนาดของข้อมูล 64000 ไบท์ที่จะอ่านได้โดยตรงจากไฟล์ FLI ลง MEMORY จะใช้ในกรณีที่ต้องการจัดเก็บข้อมูลแบบบีบขนาดแล้วเกิน 64000 ไบท์ และบางเฟรมที่ต้องการความเร็ว

5.3 การทำงานของไฟล์ PCX

PCX file เป็น file รูปภาพชนิดหนึ่งซึ่งเป็นที่นิยมใช้กันอย่างทั่วไปในการเก็บภาพนิ่งทั้งบน DOS และบน Windows ซึ่งจะประกอบด้วย header ของ .PCX file format มีขนาด 128 bytes ซึ่งแสดงได้ดังในตารางที่ 5.5

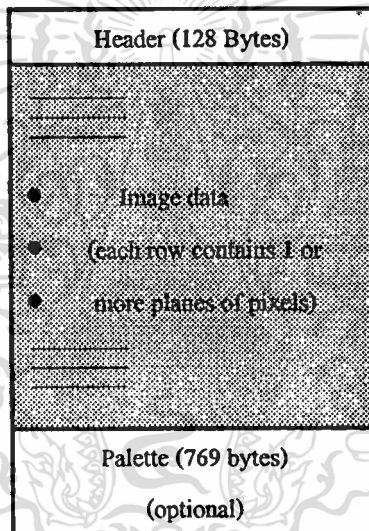
Byte	Size (Bytes)	Name	Description
0	1	Password	ใน byte แรกของ file ในรูปแบบ PCX จะมีค่า = 0AH เสมอ
1	1	Version	Version ของ PC Paintbrush คือ : 0 = version 2.5 2 = version 2.8 with palette information. 3 = version 2.8 without palette information. 5 = version 3.0
2	1	Encoding	Encoding scheme used. 1 = .PCX run length encoding.
3	1	Bits per pixel	จำนวน bit ที่ใช้ในการเก็บ data สำหรับ 1 pixel จาก 1 plan 1 = EGA,VGA, or Hercules 2 = CGA
4	8	Window Dimension	เป็น integer 4 ตัว (ตัวละ 2 bytes) สำหรับเก็บ มุม top left และ bottom right ของภาพที่ display x1,y1,x2,y2 ตามลำดับ
12	2	Horizontal Resolution	Horizontal resolution ของอุปกรณ์ที่ display (columns) 640 = EGA,VGA 320 = CGA 720 = Hercules
14	2	Vertical Resolution	Vertical resolution ของอุปกรณ์ที่ display (rows) 480 = VGA 350 = EGA 200 = CGA 348 = Hercules
16	48	Color map	ข้อมูลของ color palette (ชุดสี) ของรูปภาพ
64	1	Reserved	flag
65	1	Number of plan	จำนวนของ color planes ใน original image 1 = CGA,Hercules 4 = EGA,VGA
66	2	Bytes per line	จำนวน byte ต่อการ scan 1 line ของรูปภาพ
68	2	Palette description information	ชนิดของ palette
70	56	Not used	Fill to the end of the header block

ตารางที่ 5.5 แสดงส่วนของ header ของไฟล์ในรูปแบบ PCX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.1 256-color PCX files

เมื่อเราทำการเปิด file ในรูปแบบของ PCX file ค่า byte แรกที่อ่านได้จะมีค่าเป็น 0Ah เสมอ PCX file ที่เป็น 256 color ค่าของ version number data ใน header (byte 1) จะมีค่าเท่ากับ 5 (version 3.0) file รูปถ่ายใน format ของ PCX image data จะถูกเข้ารหัส code อยู่ในแบบของ Run length encoding (RLE) ฉะนั้นในการที่จะ read หรือ write file PCX จะต้องมีการทำ unpack data หรือ compress data ตามลำดับ ใน PCX file - 256 color จะมีการเก็บค่า palette ที่ end of file หลังจากที่เก็บ image data palette register จะมีขนาด 769 bytes โดยจะเก็บ 3 byte ต่อ RGB colors ใน mode ของ 256 colors palette จะเก็บค่า 0Ch นำมาก่อนเสมอ การอ่านค่าจะทำได้โดยการอ่านจากท้าย file ช้อนกลับ 769 bytes แล้วทำการอ่านค่าที่ตำแหน่งนั้นออกมา ถ้าค่าที่อ่านได้ไม่เท่ากับ 0Ch แสดงว่าไม่ใช่ mode ของ 256 colors

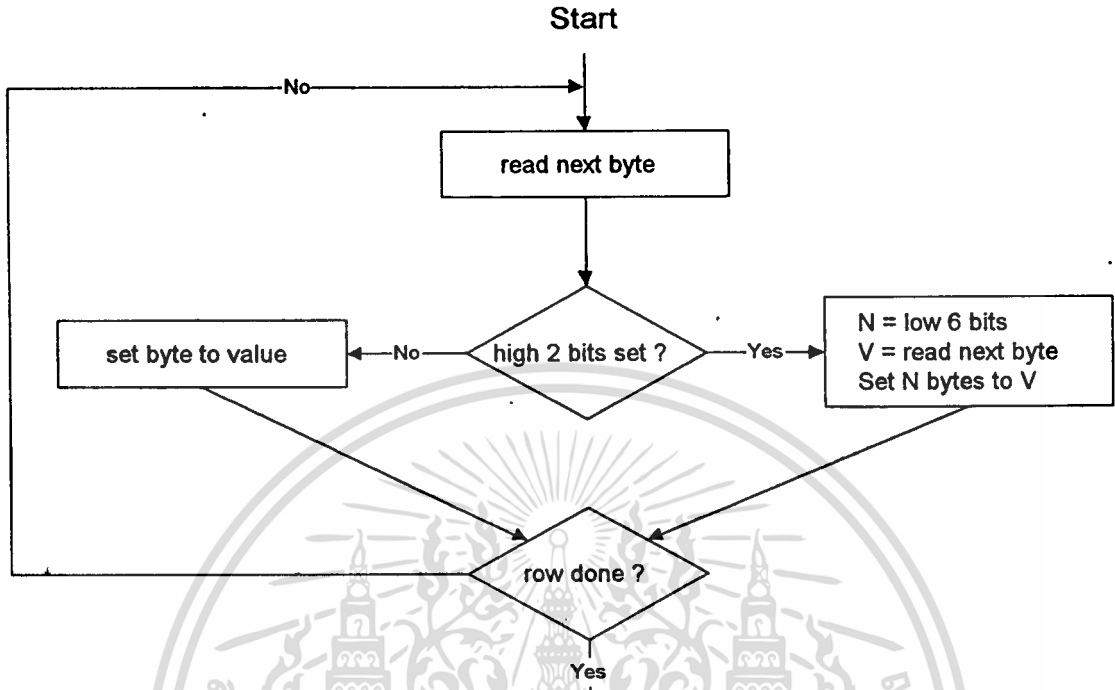


รูปที่ 5.9 Structure ของ PCX graphics file

5.3.2 Run length encoding

เมื่อเรา read ค่า header file 128 bytes เรียบร้อยแล้ว byte ถัดไปเป็น byte แรก จะเป็นค่าของ image data ที่ถูก compress เมื่อเราจะทำการ display ภาพจะต้องทำการ unpack data ออกมาก่อน PCX format จะใช้วิธี run-length encoding อย่างง่าย แต่ละ line ใน PCX file จะ unpack จำนวน byte ใน field ของ byte per line ของ header ของมัน

ใน PCX run-length compression ถ้าใน byte ของมัน 2 high order bits set (11XXXXXX) มันเป็น key byte สำหรับการ run ของ bytes field เราจะใช้ 6 low order bits แทนความยาวของ field ความยาวของ field นี้จะมีค่าตั้ง 0 - 63 ส่วน byte ถัดไปใน file จะถูกทำซ้ำเท่ากับจำนวนความยาวของ field ถ้า 2 high order bits ไม่ถูก set byte ที่ read ออกมาจาก source PCX file มันไม่ได้ถูก compress



รูปที่ 5.10 flowchart แสดงการ decode row เพื่อ compressed pixel จาก file .PCX

```

readpcxline(p,fp,bytes)
char *p;
FILE *fp;
int bytes;

{
    int n = 0,c,i;
    do {
        c=fgetc(fp) & 0xff;

```

รูปที่ 5.11 แสดง function unpack line in pcx file

```

if ((c & 0xC0) == 0xC0) /* check key byte */
{
    i = c & 0x3f;
    c = fgetc(fp);
    while(i--) p[n++] = c;
}
else p[n++] = c;
} while (n < bytes);
return(n);
}

```

รูปที่ 5.11 (ต่อ) แสดง function unpack line in pex file

บทที่ 6

การออกแบบและการสร้าง

6.1 ขั้นตอนในการดำเนินงาน

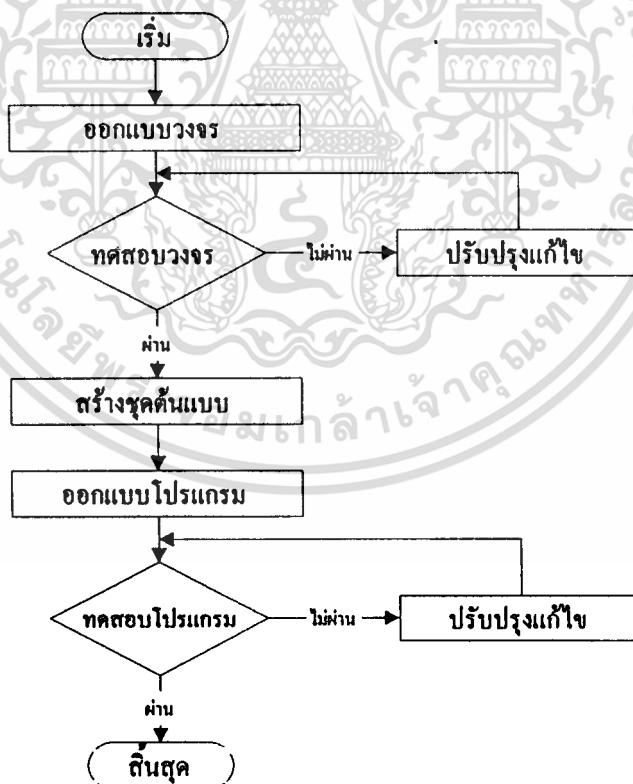
การดำเนินงานออกแบบสร้างโครงการแบ่งออกเป็น 2 ส่วนคือ

1. ทางด้านฮาร์ดแวร์ซึ่งประกอบไปด้วย

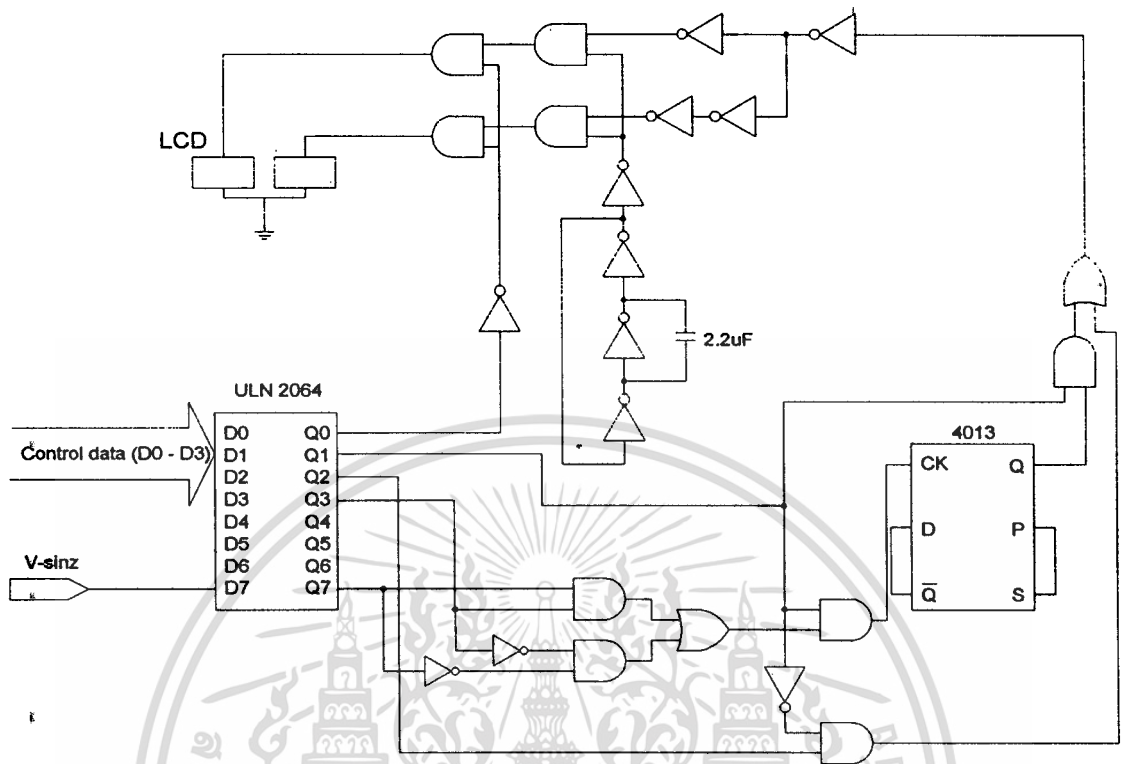
- ภาควงควบคุมการเปิดปิดแวนตา LCD
- ภาควงควบคุมการทำงาน
- ภาควงรับส่งข้อมูลกับคอมพิวเตอร์

2. ทางด้านซอฟต์แวร์ซึ่งประกอบไปด้วย

- โปรแกรมสร้าง script file ที่ใช้สำหรับการ run โปรแกรม
- โปรแกรมที่อ่านค่า script file มาทำการแสดงผลออกหน้าจอ
- โปรแกรมที่ทำหน้าที่แสดงภาพในแบบ 3 มิติ



รูปที่ 6.1 แสดงแผนผังการดำเนินงาน



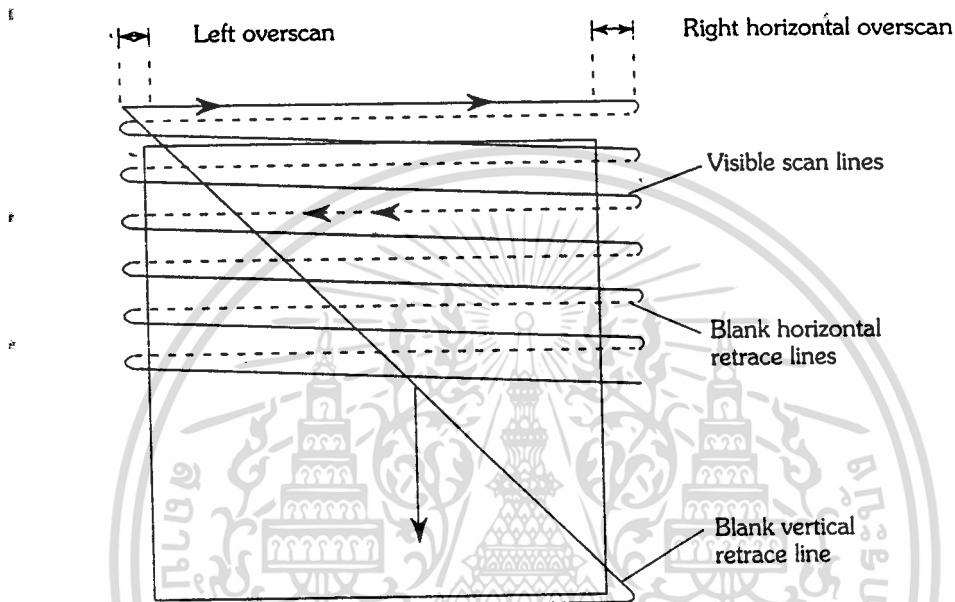
รูปที่ 6.4 วงจรควบคุมการทำงานของแวนตา LCD

วงจร IC ULN2064 จะทำหน้าที่เป็น Buffer และทำการแปลงแรงดันระดับ TTL ให้เป็นแรงดันระดับของ CMOS เพื่อให้สามารถขับแวนตา LCD ได้ โดยตัวแวนตา LCD นี้จะต้องทำการเปิดปิดอย่างรวดเร็ว ดังนั้นแรงดัน 5 โวลต์ไม่สามารถที่จะทำให้ผลึก crystal ปิดตัวได้อย่างรวดเร็ว การควบคุมแวนตาต้องอาศัยสัญญาณ control data ซึ่งจะมีการกำหนดรูปแบบการทำงานดังนี้

- | | | |
|-----|---|---|
| BIT | 0 | ทำการ ON หรือ OFF แวนตา LCD |
| BIT | 1 | เป็นตัวเลือกโหมดการทำงาน โดยถ้าบิตนี้มีค่าเป็น 0 จะเป็นการควบคุมแวนตา LCD ด้วยตัวเอง และถ้าบิตนี้เป็น 1 จะเป็นการควบคุมแวนตาแบบอัตโนมัติ |
| BIT | 2 | เป็นการควบคุมการเปิดปิด LCD ด้านซ้ายหรือขวา โดยจะทำงานร่วมกับโหมดการทำงานแบบควบคุม LCD เอง ถ้าบิตนี้มีค่าเป็น 0 จะเป็นการ OFF แวนตาด้านขวาและ ON แวนตาด้านซ้าย และถ้าบิตนี้มีค่าเป็น 1 จะเป็นการ ON แวนตาด้านขวา และ OFF ด้านซ้าย |
| BIT | 3 | มีไว้เพื่อทำการ toggle การ ON หรือ OFF แวนตา LCD ด้านซ้ายและขวา ซึ่งเกิดจากจังหวะการ scan ของจอภาพไม่เหมาะสมกับภาพที่กำลังแสดงอยู่บนหน้าจอ ซึ่งบิตนี้จะทำงานร่วมกับโหมดการทำงานแบบอัตโนมัติ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของวงจรทำหน้าที่ toggle สัญญาณจะทำสัญญาณ vertical sync จาก card VGA ของจอภาพนำมาทำการ invert สัญญาณอีกหนึ่งสัญญาณ แล้วจะใช้ AND GATE และ NOT GATE ทำการเลือกสัญญาณว่าจะนำสัญญาณ vertical sync ปกติ หรือจะนำสัญญาณ vertical sync ที่ผ่านการ invert แล้วไปควบคุมการเปิดปิดของแวนตา LCD โดยสัญญาณที่ออกมาจะต้องผ่าน AND GATE ก่อนที่จะส่งไปให้ D - FLIPFLOP จะถูกทำการหารสอง โดยปกติแล้วบนหน้าจอก็จะมีการกำหนดดังนี้



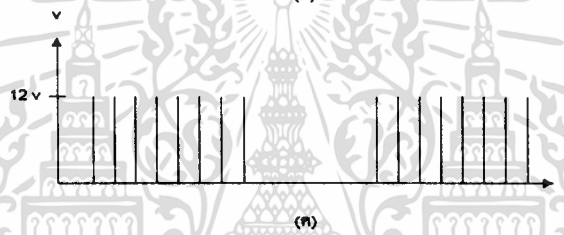
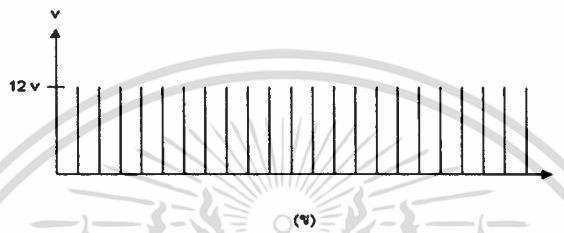
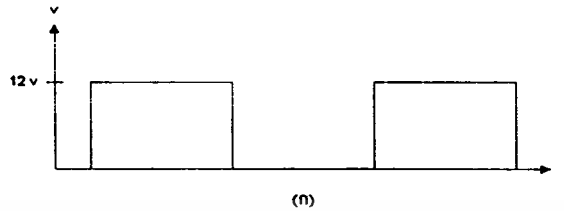
รูปที่ 6.5 รูปแสดงการสแกนของอิเล็กทรอนิกส์บนหน้าจอ

electron beam จะทำการกวาดไปตามแนวนอนจากขอบจอด้านหนึ่งไปยังขอบจออีกด้านหนึ่ง แล้วจะกลับไปเริ่มต้นการกวาดตามแนวนอนอีก แต่จะเป็นการกวาดที่แนวเส้นถัดลงมา ซึ่งเมื่อทำงานถึงมุมจอด้านล่างสุดแล้วก็จะกลับไปยังมุมจอด้านบนอีกครั้ง ช่วงเวลาการเริ่มต้นการกวาดที่มุมจอด้านบนจนมาถึงมุมจอด้านล่างนั้นคือช่วงเวลาของสัญญาณ vertical sync ซึ่งปกติแล้วจะมีค่าความถี่ประมาณ 60 Hz ดังนั้นจึงต้องทำการหารความถี่ลงมาที่ 30 Hz หรือเป็นการแสดงภาพ 1 รูปที่อัตราการแสดงผล 30 เฟรมต่อวินาที โดยการทำงานของจอภาพจำเป็นต้องอยู่ในโหมด interlace ซึ่งจะมีการแบ่งการสแกนหน้าจอออกเป็น 2 ครั้ง โดยครั้งแรกจะสแกนเส้นคี่ (scan odd) จากมุมจอด้านบนลงไปยังมุมจอด้านล่าง และครั้งที่สองจะสแกนเส้นคู่ (scan even) จากมุมจอด้านบนลงไปยังมุมจอด้านล่าง ซึ่งการสแกนเส้นคู่เส้นคี่นั้นจะใช้เวลาครั้งละประมาณ 1/60 วินาที ดังนั้น 1 ภาพจะใช้เวลา 1/30 วินาที จากหลักการทำงานของจอภาพในโหมดนี้เราจะให้ภาพของตาซ้ายถูกสแกนที่เส้นคี่ และภาพของตาขวาสแกนที่เส้นคู่ ซึ่งปกติแล้วโหมดการแสดงผลแบบ interlace นี้จะทำงานที่โหมดความละเอียดสูง ดังนั้นจึงได้ข้อมูลของภาพที่เพียงพอในการแสดงผล ส่วนภาพที่ความละเอียดต่ำ จะต้องทำการแสดงผลที่โหมดการทำงานแบบควบคุมการทำงานเอง

สัญญาณที่จะถูกส่งเข้าไปยังแวนตา LCD จะต้องมีการทำการมอดูเลต (modulate) ความถี่ประมาณ 1.2 kHz เข้าไป เนื่องจากแรงดัน 12 V ที่ควบคุมการเปิดปิด LCD นั้นจะทำให้อัตราการปิดตัวของผลึก crystal เร็ว แต่ปกติแล้วผลึก crystal จะใช้กับแรงดันไฟที่ต่ำ ซึ่งแรงดัน 12 V นี้จึงเป็นอันตรายต่อผลึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

crystal จึงจำเป็นต้องมอดูเลทความถี่ 1.2 kHz เข้าไปเพื่อช่วยให้ผลึก crystal ไม่ได้รับแรงดันที่อาจก่อให้เกิดความเสียหายมากนัก โดยความถี่ 1.2 kHz นี้จะไม่มีผลต่อดวงตาเนื่องจากดวงตาไม่สามารถมองเห็นความแตกต่างของภาพที่เกิดขึ้นได้



รูปที่ 6.6 (ก) รูปของสัญญาณที่ป้อนเข้า LCD ที่ยังไม่ได้ทำการมอดูเลท
(ข) รูปของสัญญาณความถี่ 1.2 kHz ที่จะทำการมอดูเลท
(ค) รูปของสัญญาณที่ป้อนเข้า LCD ที่ทำการมอดูเลทแล้ว

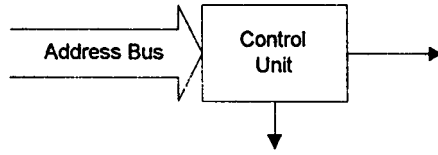
วงจรกำเนิดความถี่มอดูเลทมีการใช้คาปาซิเตอร์เพียงตัวเดียวที่ทำงานร่วมกับ NOT GATE อีก 4 ตัว เพื่อทำการสร้างความถี่ 1.2 kHz โดยคาปาซิเตอร์จะทำให้ NOT GATE ตัวที่มีคาปาซิเตอร์ต่อคร่อมอยู่มีระดับ logic ที่เปลี่ยนแปลงจาก 0 เป็น 1 และจาก 1 เป็น 0 อยู่ตลอดเวลาทำให้สามารถกำเนิดความถี่ได้เนื่องจากการป้อนกลับจาก NOT GATE ตัวที่เป็นเอาต์พุตมาป้อนกลับให้กับ NOT GATE ตัวที่เป็นอินพุต

วงจรที่ทำหน้าที่มอดูเลทจะใช้ AND GATE มาต่อเข้ากับสัญญาณที่จะป้อนให้ LCD และสัญญาณความถี่ 1.2 kHz ที่จะทำการมอดูเลท ดังนั้นจะได้เอาต์พุตที่ออกมาเป็นสัญญาณที่ผ่านการมอดูเลทเรียบร้อยแล้วพร้อมที่ป้อนให้กับ LCD แต่ยังคงมีการป้อนผ่าน AND GATE อีก เพื่อที่จะสามารถควบคุมการเปิดปิดสัญญาณได้

6.2.2 ภาควควบคุม

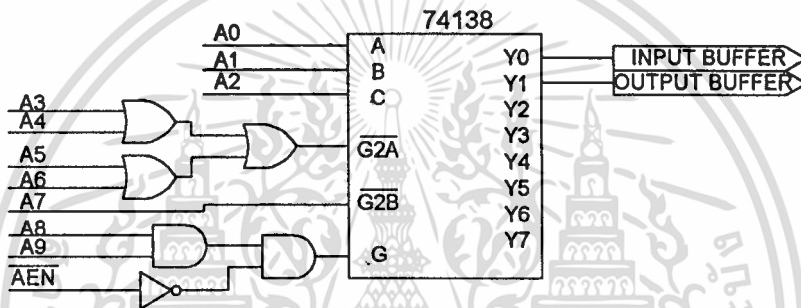
จะทำหน้าที่กำหนดการทำงานต่าง ๆ ของส่วนภาควควบคุมรับอุลตราโซนิก, ส่วนภาคแปลงสัญญาณอนาล็อกเป็นดิจิตอล, ส่วนของภาควควบคุมการทำงานของ LCD และส่วนของภาค Buffer เพื่อให้สามารถทำงานร่วมกันได้ตามวัตถุประสงค์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.7 บล็อกไดอะแกรมแสดงภาคควบคุม

ในส่วนของ control unit นี้จะควบคุมการทำงานของอุปกรณ์ต่าง ๆ โดยใช้สัญญาณ address bus มาควบคุมการทำงานหรือเลือกการทำงานของอุปกรณ์ตัวนั้น ๆ

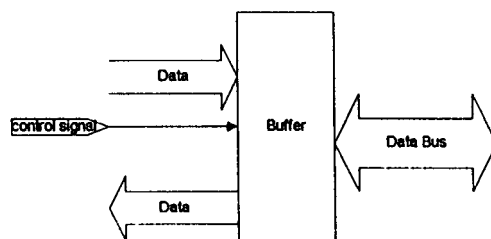


รูปที่ 6.8 แสดงวงจรภาคควบคุม

จากวงจร AND GATE จะทำการ map logic 1 จาก A8 - A9 และ AEN ให้กับขา G ของ 74138 และ OR GATE จะทำการ map logic 0 จาก A3 - A6 ให้ขา G2A และ A7 ให้กับขา G2B และสัญญาณ A0 - A2 จะเป็นตัวเลือกการทำงานว่าจะป็น input data หรือ output data และส่วนที่เป็น data จากภาคแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล

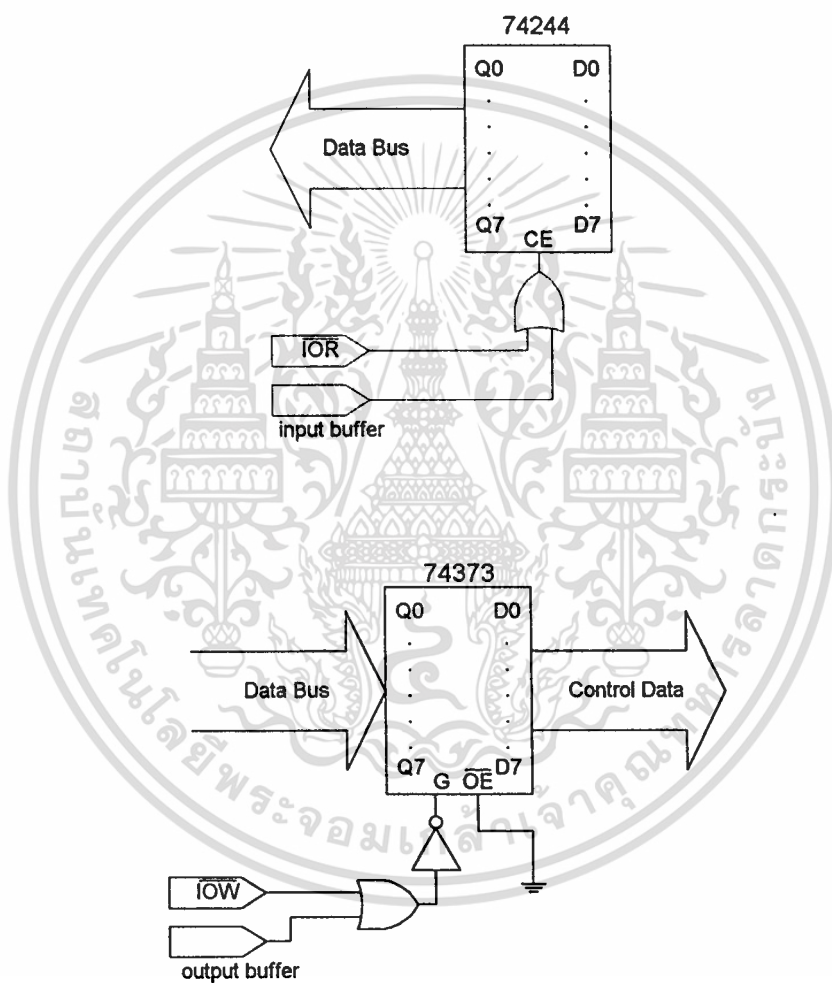
6.2.3 ภาครับส่งข้อมูลกับคอมพิวเตอร์

จะทำหน้าที่โอนย้ายข้อมูลระหว่างคอมพิวเตอร์กับการ์ดอินเทอร์เฟซ และยังมีแลตช์ (Latch) ข้อมูลไว้เพื่อใช้ในการทำงานของวงจรส่วนต่าง ๆ



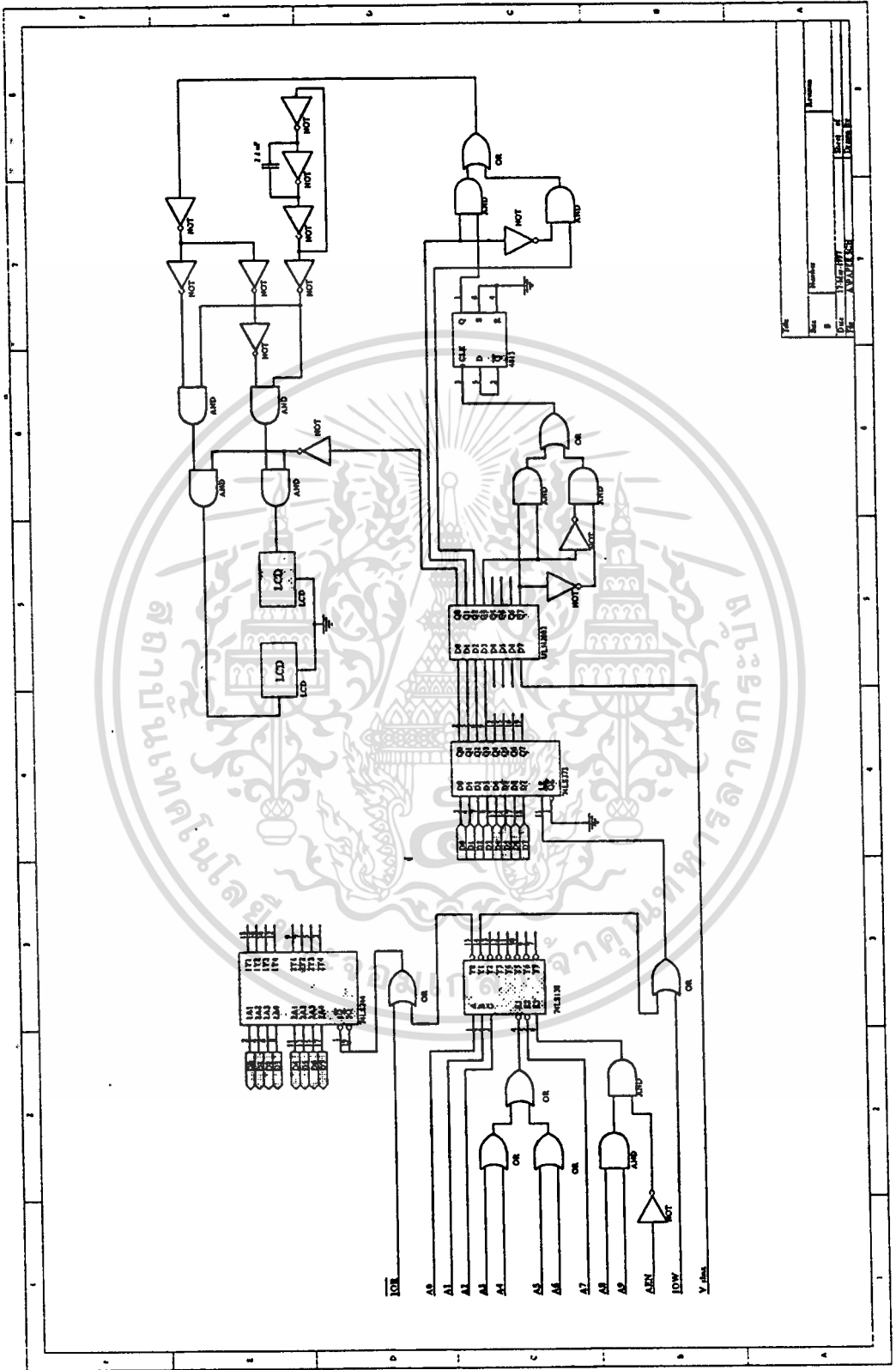
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 6.9 บล็อกไดอะแกรมแสดงส่วนของภาค Buffer นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากบล็อกไดอะแกรมจะเห็นว่าการทำงานของ buffer ต้องอาศัยสัญญาณ control signal เพื่อเลือกที่จะติดต่อกับ Data bus ซึ่งจะทำการแลกเปลี่ยนข้อมูลระหว่างการ์ดอินเทอร์เฟซ โดยสัญญาณนี้จะควบคุมให้ buffer ทำหน้าที่เป็นอินพุต buffer หรือว่าทำหน้าที่เป็นเอาต์พุต buffer จากวงจร IC#74244 ทำหน้าที่เป็นอินพุต buffer โดยจะรับข้อมูลจากการ์ดอินเทอร์เฟซ ส่งไปให้คอมพิวเตอร์เพื่อใช้ในการประมวลผล และ IC#74373 ทำหน้าที่เป็นเอาต์พุต buffer เพื่อที่จะนำข้อมูลจากคอมพิวเตอร์ส่งไปให้การ์ดอินเทอร์เฟซ เพื่อใช้ในการควบคุมการทำงานของการ์ดอินเทอร์เฟซส่วนต่าง ๆ



รูป 6.10 แสดงวงจรของภาค buffer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



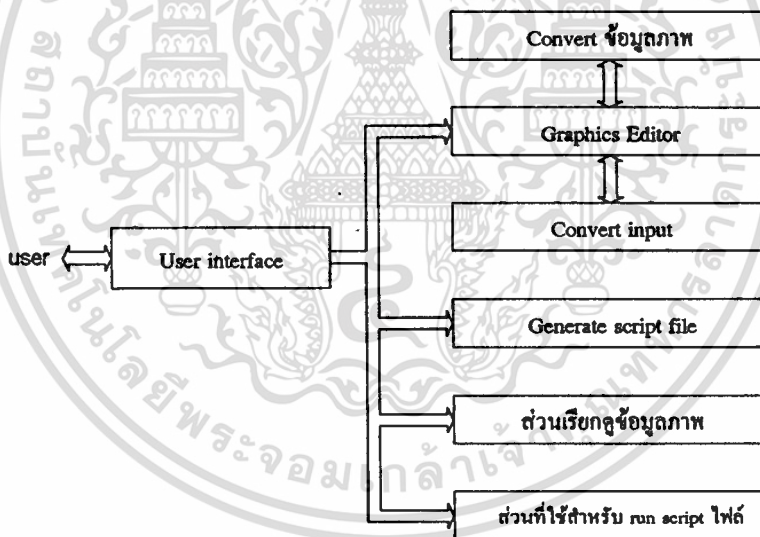
รูปที่ 6.11 แสดงภาพวงจรรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.8 การออกแบบทางด้านซอฟต์แวร์ (Software design)

การออกแบบทางด้านซอฟต์แวร์ได้ทำการออกแบบและสร้างโปรแกรมในส่วนของการสร้าง script ไฟล์ ซึ่งได้ถูกออกแบบให้เป็นลักษณะของ Graphics Editor เพื่อให้ผู้ใช้สามารถเข้าใจลำดับการดำเนินเรื่องราวของแต่ละไฟล์ได้ ซึ่งการนำเสนอในรูปแบบของ Graphics Editor นี้ จะทำให้ผู้ใช้ทำการสร้างหรือแก้ไขลำดับการดำเนินเรื่องได้ง่าย โครงสร้างของตัวโปรแกรมสร้าง script ไฟล์ จะประกอบด้วยส่วนต่าง ๆ ดังนี้

1. ส่วน Graphics Editor
2. ส่วน Convert ข้อมูลภาพ หรือส่วนเตรียมข้อมูล
3. ส่วน Capture input
4. ส่วน Generate script ไฟล์
5. ส่วนที่ใช้เรียกดูข้อมูลภาพ
6. ส่วนที่ใช้สำหรับ Run script ไฟล์
7. ส่วนของ User interface

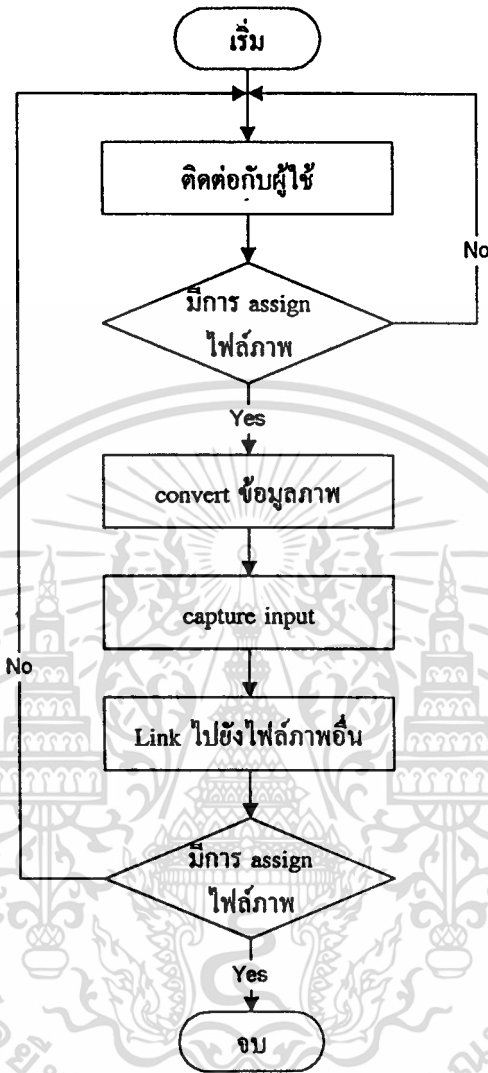


รูปที่ 6.12 บล็อกไดอะแกรมแสดงการทำงานทางด้านซอฟต์แวร์โดยรวม

6.3.1 Graphics editor

ส่วนของ Graphic editor เป็นส่วนที่ทำหน้าที่สร้างและแก้ไข script ไฟล์เพื่อให้ดำเนินเรื่องไปตามลำดับเหตุการณ์ที่ผู้ใช้กำหนด โดยส่วนของ Graphic editor ยังต้องทำงานติดต่อกับส่วนของ Graphic editor จะทำงานดังนี้

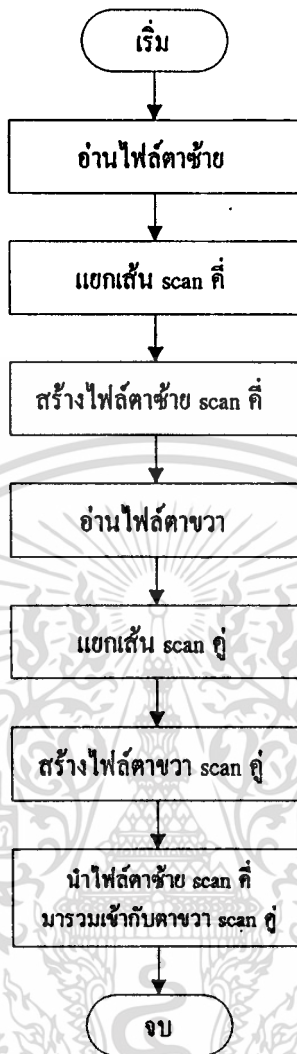
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.13 แสดงการทำงานของส่วน Graphic editor

6.3.2 Convert ข้อมูลภาพ

ส่วนของการ Convert ข้อมูลภาพ จะทำหน้าที่เตรียมไฟล์เพื่อแสดงในโหมด Automatic Shutter ซึ่งไฟล์ภาพของตาซ้ายจะต้องถูกแยกนำข้อมูลออกมาเฉพาะเส้น Scan ที่ และ ไฟล์ภาพของตาขวาจะต้องถูกแยกนำข้อมูลออกมาเฉพาะเส้น Scan คู่ จากนั้นจำเป็นที่จะต้องนำมารวมกัน ก็จะได้ภาพของตาซ้ายและตาขวาในไฟล์เดียวกัน



รูปที่ 6.14 แสดงการ convert ข้อมูลภาพ

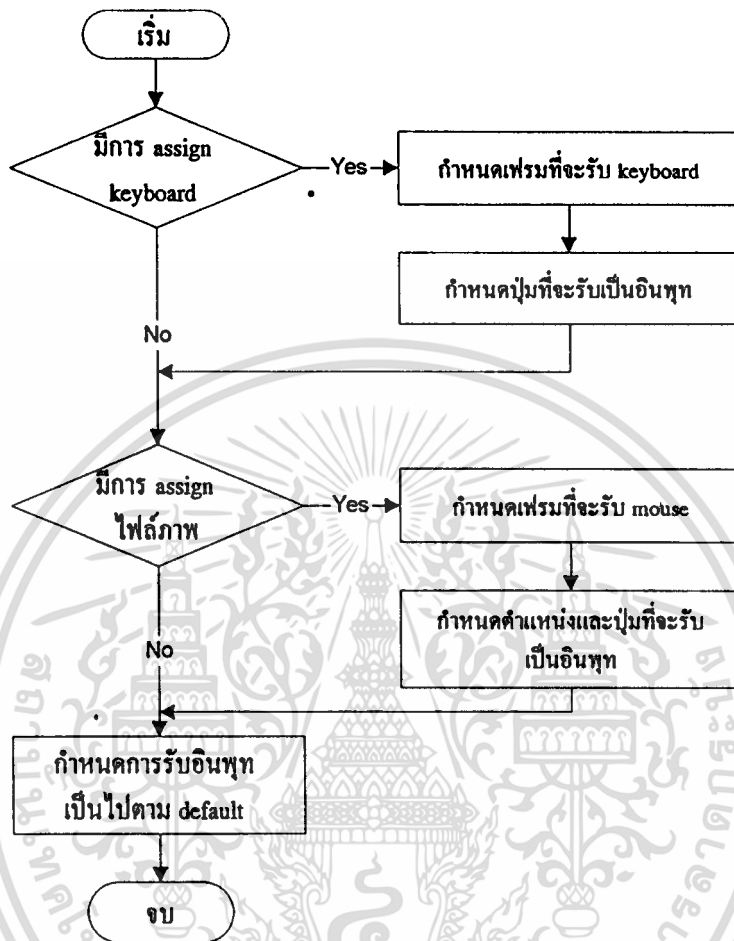
6.3.3 Capture input

ส่วนของการ capture input จะทำหน้าที่ติดต่อกับ user ขณะที่ไฟล์ภาพกำลังถูกดำเนินการเรียงรวบอยู่ ซึ่งอินพุตที่สามารถใช้งานได้มีคือ คีย์บอร์ด (keyboard) และ เมาส์ (mouse) โดยจะต้องมีการกำหนดช่วงของอินพุต ที่จะรับเข้ามาในแต่ละเฟรมภาพ ดังนั้นจึงต้องมีการ capture input เพื่อกำหนดการรับอินพุตล่วงหน้าไว้ให้ script ไฟล์

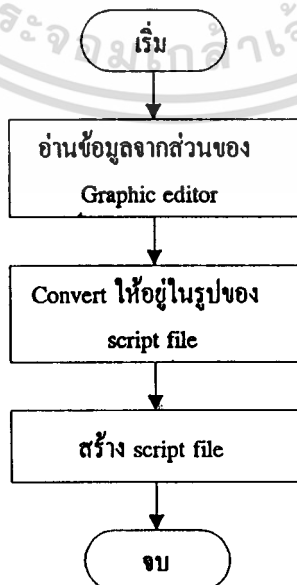
6.3.4 Generate script file

ส่วนของการ Generate script ไฟล์ จะนำข้อมูลที่ได้จากส่วนของ Graphic editor มาสร้างไฟล์ที่ใช้สำหรับทำการ run โปรแกรม เพื่อกำหนดเหตุการณ์ไปตามลำดับที่ได้กำหนดเอาไว้ในส่วนของ Graphics editor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

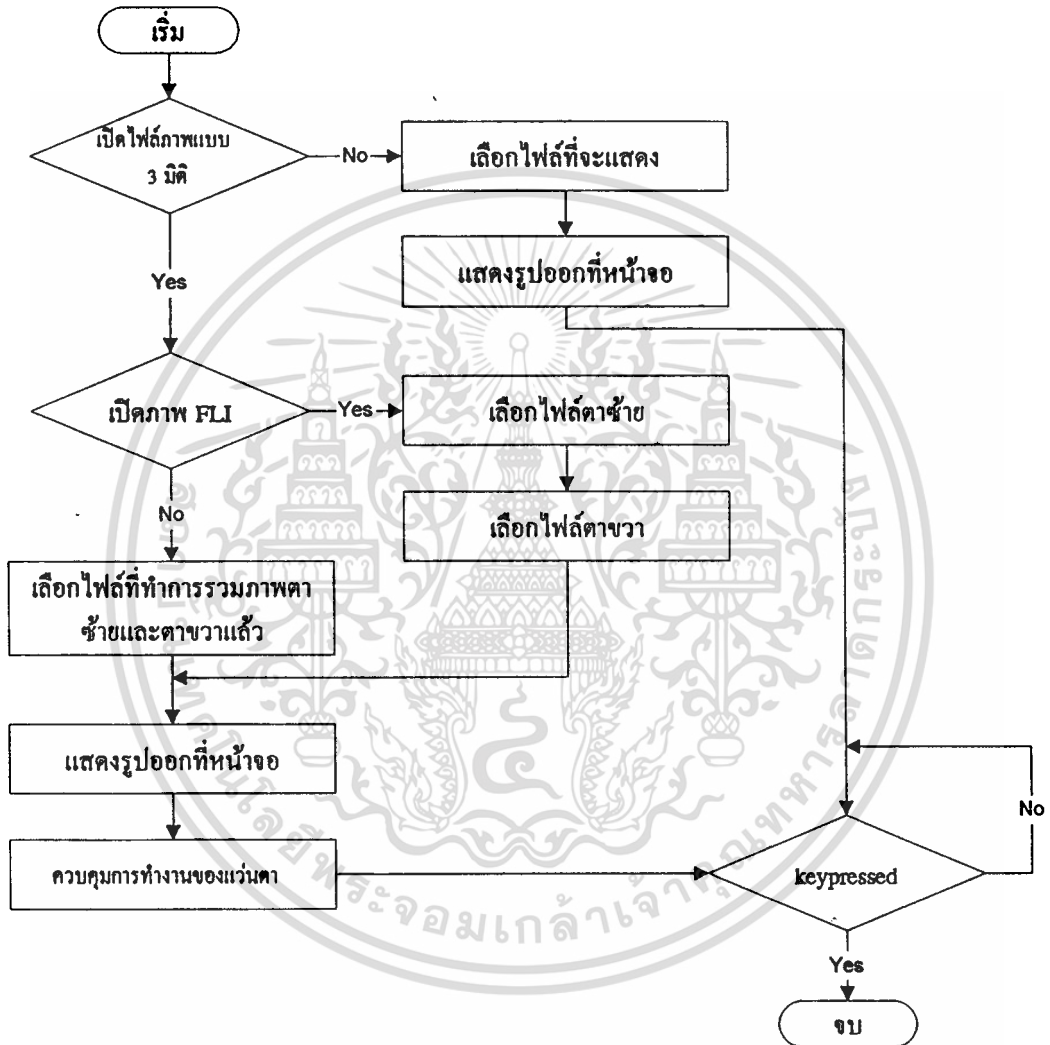


รูปที่ 6.15 แสดงหลักการทำงานในการ capture input



6.3.5 การเรียกดูข้อมูลภาพ

ในส่วนของ การเรียกดูข้อมูลภาพ จะนำภาพที่ต้องการออกมา display บนหน้าจอในรูปแบบ ธรรมดาและรูปแบบที่เป็น 3 มิติ เพื่อให้ผู้ใช้ดูว่าเป็นภาพที่ต้องการหรือไม่



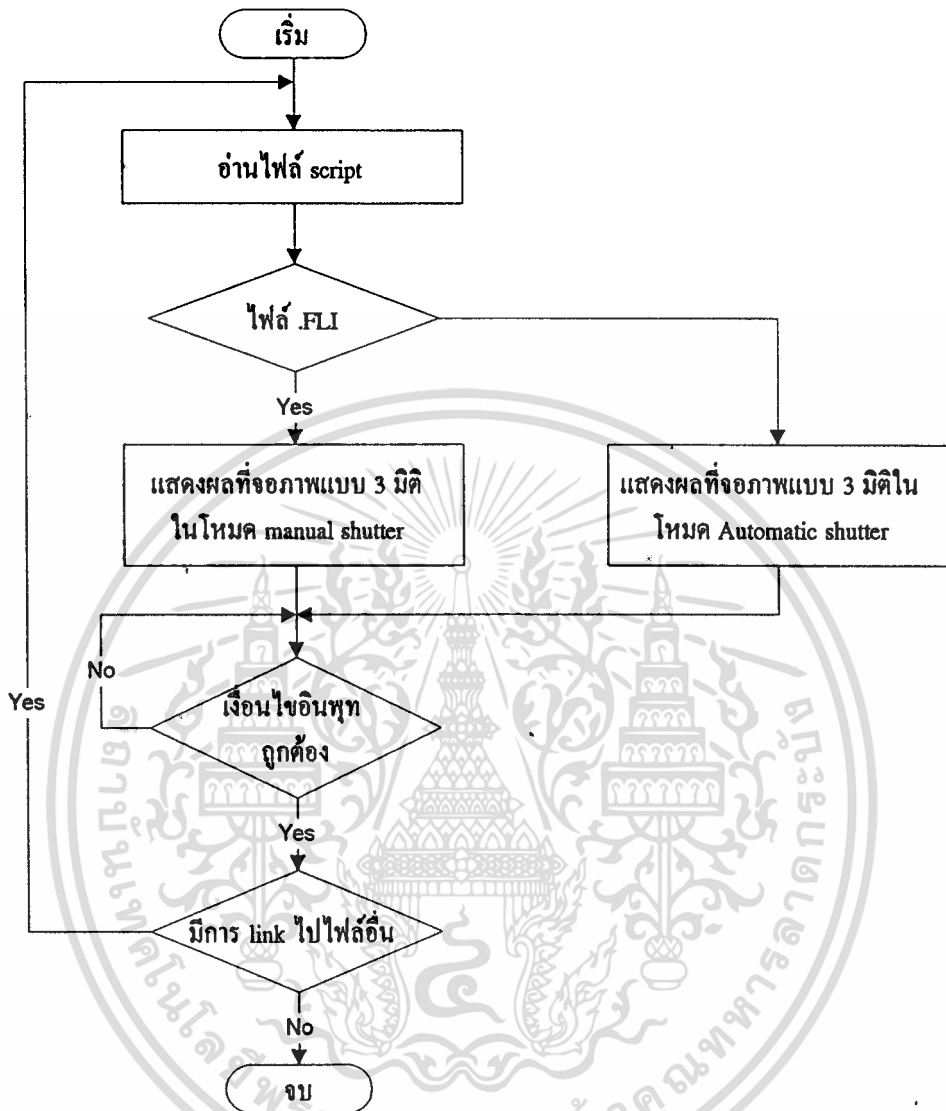
รูปที่ 6.17 แสดงการเรียกดูข้อมูลภาพ

6.3.6 การ Run Script ไฟล์

ในการ run script ไฟล์ จะทำการเปิดไฟล์ในรูปแบบ 3 มิติ เพื่อแสดงผลที่หน้าจอภาพ และมีการรับอินพุตตามลำดับที่ได้กำหนดไว้ และดำเนินเรื่องไปตามที่กำหนดใน Graphics editor

ส่วนของโปรแกรม run script ไฟล์ได้ถูกรวมเอาไว้ในโปรแกรมสร้าง script ไฟล์นี้ด้วยและยังแยกออกมาเพื่อมีไว้สำหรับทำการ run script ไฟล์เพียงอย่างเดียวอีกด้วยเพื่อให้สะดวกในการนำไปใช้งาน

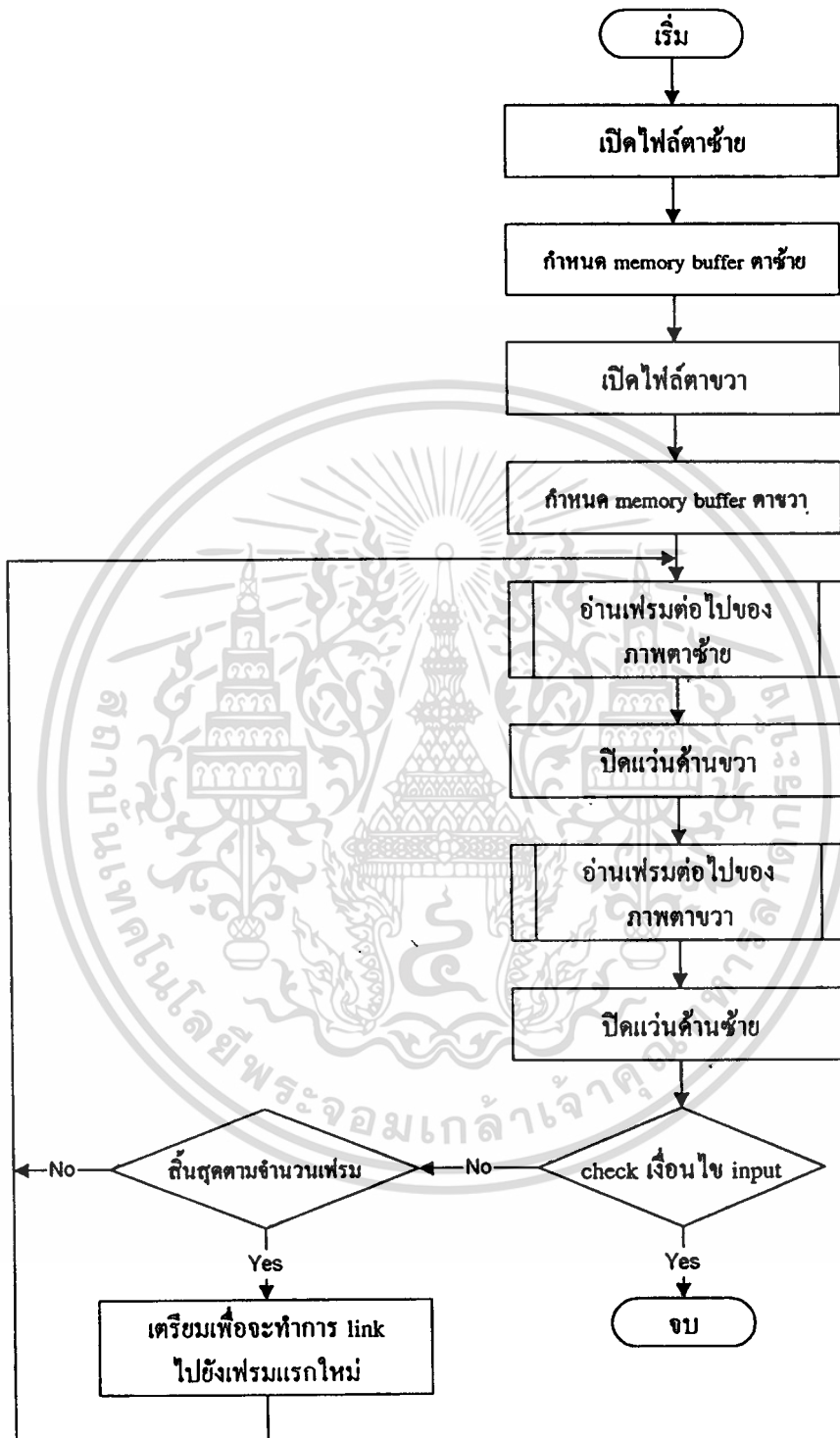
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.18 แสดงหลักการทำงานในการ Run script ไฟล์

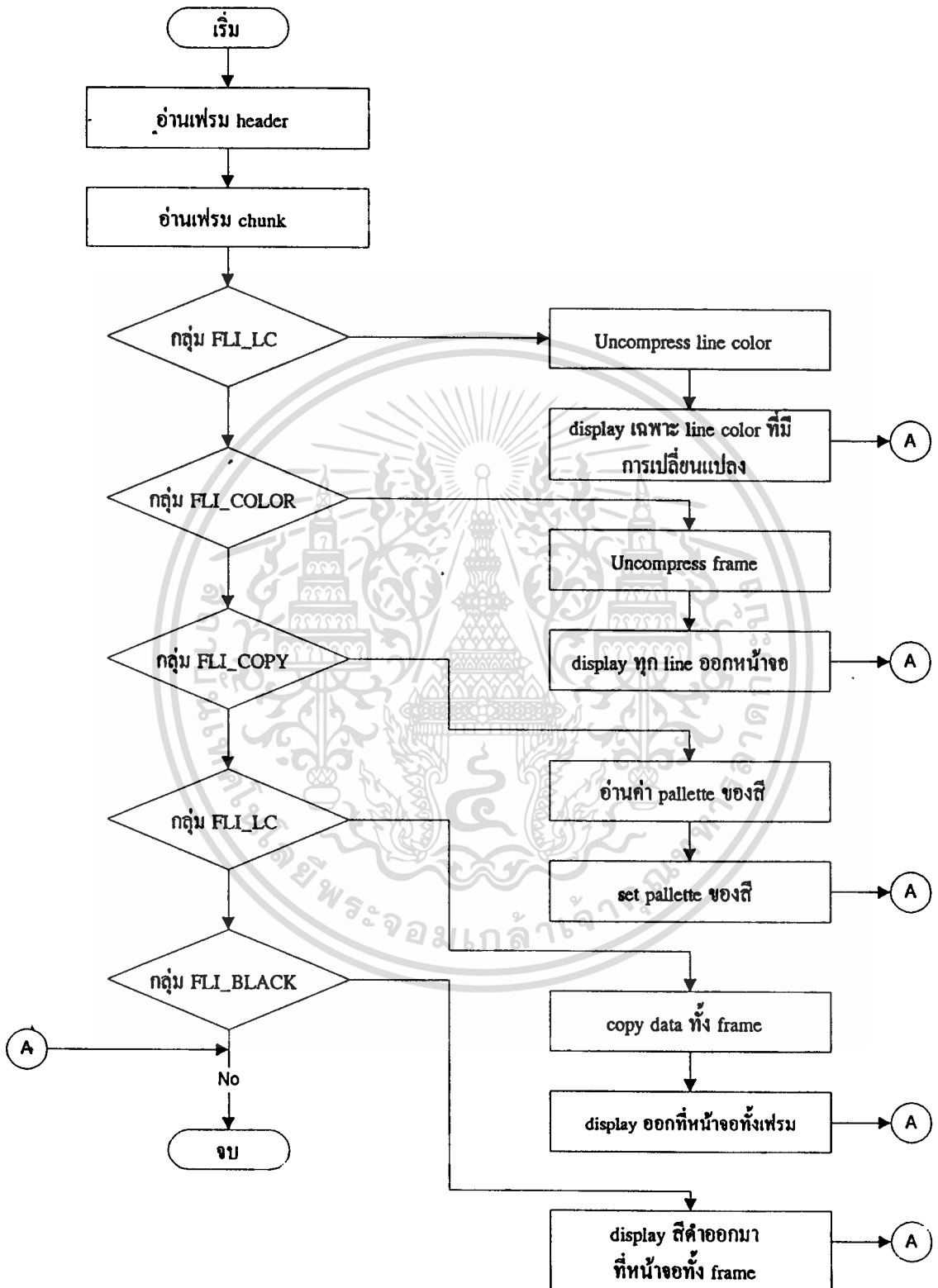
6.3.7 โปรแกรมเปิดไฟล์ภาพ

โปรแกรมในส่วนของการเปิดไฟล์ภาพแบ่งได้เป็น 2 รูปแบบ คือ ไฟล์ภาพเคลื่อนไหว FLI และไฟล์ภาพนิ่ง PCX โดยในส่วน of ไฟล์ภาพเคลื่อนไหวนี้จะต้องทำงานในโหมด manual shutter และไฟล์ภาพนิ่งจะต้องทำงานในโหมด Automatic shutter



รูปที่ 6.19 แสดงหลักการทำงานในการเปิดภาพเคลื่อนไหว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.20 แสดง subroutine ในการอ่านเฟรมต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้จัดทำหน้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

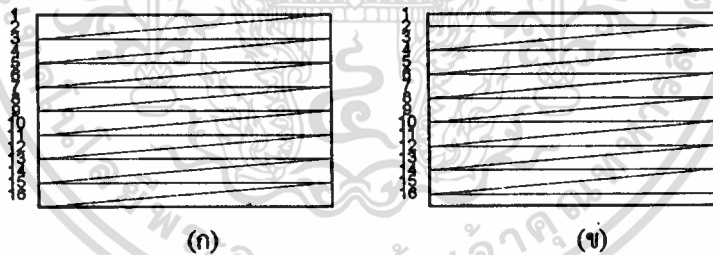
บทที่ 7

บทวิจารณ์และสรุป

7.1 สรุปผลการทดลอง

ปัญหาในการเปิดปิดภาพในการสแกนภาพในโหมดความละเอียดต่ำและความละเอียดสูง ในการแสดงความละเอียดต่ำเราสามารถทำการปิดเปิดภาพทางด้านซ้ายและด้านขวาสลับกัน พร้อมกันนั้นก็เปิดเปิด LCD ให้เหมาะสมกับการแสดงภาพก็จะทำให้เห็นภาพมิมิติได้ แต่ปัญหาก็คือต้องใช้เครื่องที่มีความเร็วในการทำงานสูงพอสมควร เพื่อที่จะทำให้ภาพที่แสดงไม่กะตุกแต่ในการแสดงภาพในความละเอียดสูงไม่สามารถทำวิธีนี้ได้ เพราะการถ่ายโอนข้อมูลจากภาพซ้ายและภาพขวาซึ่งต้องสลับกัน ใช้ข้อมูลจำนวนมากในการสลับกันแสดงภาพ จึงมาใช้โหมด interlace มาแสดงภาพ เพราะว่าไม่ต้องถ่ายโอนข้อมูลภาพซ้ายและภาพขวา เพียงใส่ข้อมูลลงในตำแหน่งสแกนคู่หรือสแกนคี่ตามภาพที่ถูกต้อง ในโหมดนี้จะจัดการนำไปแสดงเอง เช่น ในช่วงหน่วยความจำของการแสดงภาพในการสแกนเส้นคู่ ก็นำรูปของภาพตาซ้ายไปส่งไปและในหน่วยความจำของการแสดงภาพในการสแกนเส้นคี่ก็นำเส้นคี่ก็นำรูปของภาพตาขวาไปส่งไปในโหมดนี้ก็จะจัดการเองโดยเรามีหน้าที่ที่จะต้องเปิด LCD ให้เหมาะสมกับการสแกนภาพ

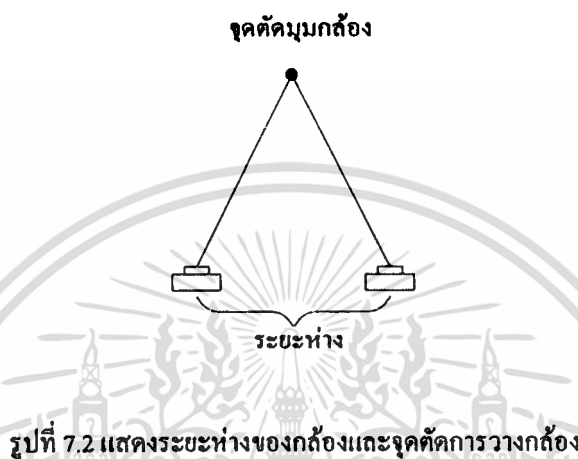
ในการสแกนภาพในโหมด interlace จะเห็นได้ว่าการสแกนภาพ 2 ครั้งใน 1 ภาพ คือจะสแกนเส้นคู่หนึ่งครั้งและสแกนเส้นคี่ 1 ครั้ง



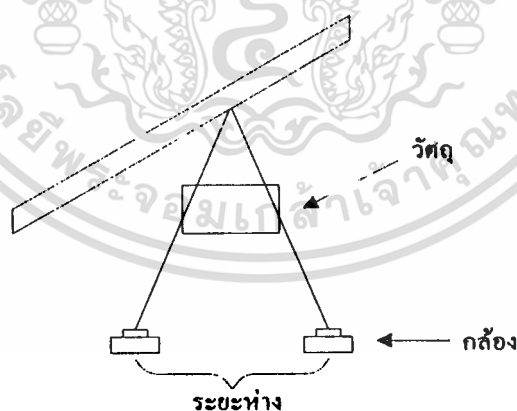
รูปที่ 7.1 (ก) ในช่วงเวลาแรก (ข) ช่วงเวลาต่อมา

จะเห็นได้ว่า เมื่อเราเปิดปิดเครื่องในครั้งแรก เราไม่สามารถบอกได้ว่าเส้นสแกนเส้นไหนจะเริ่มสแกนก่อนในเหตุการณ์เดียวกัน จะเกิดขึ้นในตอนเปลี่ยนโหมดจอด้วย ซึ่งจากทฤษฎี เราจำเป็นต้องเปิดปิด LCD ให้เหมาะสมกับการแสดงภาพ ดังนั้น ถ้าเราให้ภาพที่ไม่เหมาะสมกันโดยที่ถ้าเราให้เส้นสแกนเส้นคู่เป็นของตาซ้ายและเส้นสแกนเส้นคี่เป็นของตาขวา ถ้าเราเปิดได้เหมาะสมจะทำให้เราสามารถเห็นมิติของรูปภาพได้ แต่ถ้าเราให้ไม่เหมาะสมจะทำให้เราไม่สามารถเห็นมิติของภาพได้ ดังนั้นปัญหาที่เราจึงต้องมี key ที่ใช้สำหรับสลับการแสดงผลให้เหมาะสมกับเส้นของการสแกน

ปัญหาในการสร้างภาพ 3 มิติ โดยการใช้ 3D Studio ในการสร้างต้องใช้ความชำนาญในการใช้อย่างสูง และต้องใช้เครื่องคอมพิวเตอร์ที่มีความสามารถสูง ถึงกระนั้นก็ตามในการสร้างภาพเคลื่อนไหวก็ยังใช้เวลาในการสร้างภาพเป็นเวลานานเช่นกัน และยังมีปัญหาในการวางกล้องในการสร้างภาพ 3 มิติอีกด้วย ในการวางกล้องใน 3D studio มีผลทั้งระยะห่างของกล้องและจุดตัดของการวางกล้อง



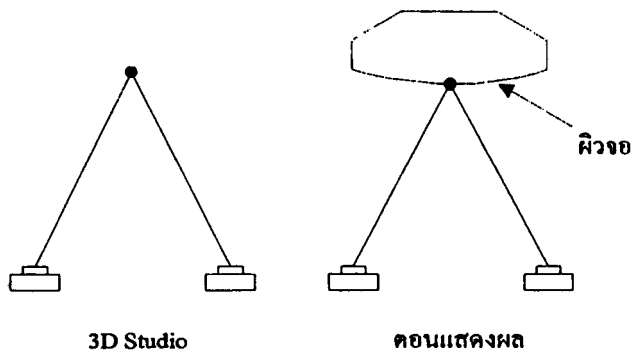
โดยที่ระยะห่างของกล้องนี้จะทำให้องเห็น ในมุมต่างกันมากขนาดไหนซึ่งจะมีผลกับมิติของการมองเห็น เหมือนกับการที่เรามองวัตถุชิ้นใหญ่กับชิ้นเล็ก ถ้าเรามองภาพวัตถุขนาดเล็กก็จะมีมิติของภาพที่มากกว่า เรามองภาพของวัตถุขนาดใหญ่ และมีผลของการที่มองเห็นมีมิตินาน้อยยิ่งขึ้นกับฉากหลังของวัตถุด้วยเช่น



รูปที่ 7.3 การมองวัตถุที่มีฉากหลัง

จะเห็นว่าฉากหลังจะมีผลทำให้ได้มิติภาพที่ดีขึ้นเหมือนกับเรามองวัตถุ ที่มีฉากหลังสีเดียวกับเรามองวัตถุที่มีความลึกในตัวเอง ซึ่งถ้ามีวัตถุที่มีความลึกอยู่ด้านหลัง จะทำให้วัตถุมีมิติที่ดีกว่า จุดตัดของมุมกล้อง จุดที่มุมกล้องตัดกันพอดีจะมองจุดตัดนั้นพอดีกับผิวจอพอดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.4 แสดงตำแหน่งของจุดตัดมุมกล้องใน 3D Studio เมื่อปรากฏบนหน้าจอ

แต่เราจะไม่สามารถแยกความแตกต่างที่ออกมาจากหน้าจอเพียงเล็กน้อยได้ ดังนั้นจุดตัดจึงควรอยู่หน้าหรือหลังวัตถุมาก ๆ แต่อย่างไรก็ตาม ถ้าเราวางห่างมากจนเกินไป เราจะมีสองวัตถุที่เคลื่อนไหวกว้างไปพร้อมกัน



รูปที่ 7.5 แสดงการวางมุมกล้องตั้งฉากกับแนววงกตั้ง

ถ้าแนวตัดมุมกล้องนี้เปลี่ยนไปจะเหมือนว่าเรามองวัตถุโดยที่ไม่ได้หันศีรษะไปด้วยทำให้ภาพที่ได้ คนที่มองต้องปรับความรู้สึกพอสมควร

ปัญหาที่เห็นได้ชัดคือ การวางมุมกล้องใน 3D studio ทั้งในความรู้สึกตอนวางกล้องในตอนทำงานกับ 3D studio ซึ่งจะจัดตำแหน่งได้ยากมากและในการทดสอบก็ต้องทำการสร้างภาพขึ้นมาจริง ๆ แล้วมาลงในโปรแกรมถ่ายภาพ 3 มิติ

ในการสร้างภาพตาซ้ายและภาพตาขวาแสดงออกหน้าจออาจมีจุดที่ผิดพลาดเกิดเป็นจุด ๆ ตามขอบวัตถุเมื่อภาพของวัตถุเคลื่อนที่ ในปัญหานี้ทางกลุ่มยังสรุปปัญหาไม่ได้ แต่คิดว่าอาจเป็นความแตกต่างของภาพตาซ้ายและภาพตาขวาที่แตกต่างกัน ในช่วงที่แตกต่างกันนั้นเมื่อนำมาสลับกันอาจทำให้เกิดปัญหานี้ได้แต่กระนั้นก็ยังไม่สามารถสรุปปัญหาได้แน่ชัด

7.2 แนวทางในการพัฒนาต่อ

เทคโนโลยีทางด้าน virtual reality นั้นมีอยู่หลายประเภท เช่น การมองเห็นวัตถุแบบ 3 มิติ, การสร้างเสียงแบบ 3 มิติ, การอ้างตำแหน่งในระบบ real world, การจำลองแรงกระทบที่มีต่อประสาทสัมผัส และด้านอื่น ๆ อีก ซึ่งในโครงการนี้ได้นำเสนอเพียงด้านการมองเห็นวัตถุแบบ 3 มิติ ดังนั้นแนวทางในการพัฒนาต่อสามารถที่จะทำได้ ดังนี้

1. พัฒนาระบบการมองเห็นภาพ 3 มิติในรูปแบบของ head-mounted visual display (HMD)
2. พัฒนาระบบของเสียงให้มีระบบเสียง 3 มิติ
3. พัฒนาการอ้างตำแหน่งในระบบ real world
4. พัฒนาด้านการสัมผัสวัตถุต่าง ๆ
5. พัฒนาเพื่อประยุกต์นำไปใช้ในสื่อการสอนทางด้านที่สื่อออกมาได้หลากหลายที่จะให้มองเห็นมิติจริง ๆ
6. พัฒนาในงานที่ต้องใช้ความเสี่ยงสูง เช่น พัฒนาการจำลองอุปกรณ์แพทย์, จำลองการฝึกบิน



กิตติกรรมประกาศ

โครงการนี้สำเร็จลงได้ก็โดยความช่วยเหลือและความร่วมมือจากท่านผู้มีพระคุณทั้งหลายดังต่อไปนี้

- ขอบพระคุณ อาจารย์บรรจง ปิยธำรง อาจารย์ที่ปรึกษาโครงการ ที่ให้การสนับสนุนและคำปรึกษาแนะนำและแนวคิดในด้านต่าง ๆ เป็นอย่างดี รวมทั้งช่วยหางบประมาณในการทำโครงการในครั้งนี้
- ขอบพระคุณอาจารย์ทุกท่าน ได้ตั้งสอนให้วิชาความรู้ตลอดเวลา 3 ปีที่ศึกษาในสถาบันการศึกษาแห่งนี้
- ขอบพระคุณพี่พจนารถ มหาพานิช ที่ให้คำปรึกษาแนะนำ แลกเปลี่ยนความรู้ในการใช้งาน 3D studio และช่วยแก้ปัญหาในด้านต่าง ๆ เสมอมา
- ขอบขอบคุณน้องหนึ่งฤทัย บุญโสพิศ ที่ได้ให้ความช่วยเหลือในด้านการจัดทำ thesis และช่วยให้กำลังในการทำ project นี้
- ขอบพระคุณคุณพ่อคุณแม่และเพื่อน ๆ ทุก ๆ คนที่ได้ช่วยเหลือและคอยให้กำลังใจในการทำโครงการนี้ให้สำเร็จลุล่วงไปได้ด้วยดี



หนังสืออ้างอิง

1. พันจันทร์ ธนวัฒน์เสถียร และ คงสวัสดิ์ ถอร์ตนเรืองกิต, “ไมโคร โปรเซสเซอร์ ทฤษฎีและการประยุกต์ใช้งาน”, ซีเอ็ดดูเคชั่น, 610 หน้า, 2537
2. สมนึก เจียมเจริญเดช, “Autodesk 3D Studio release 4”, คอมพิวเตอร์ โปรดักส์ แอนด์ คอนซัลแตนท์, 833 หน้า, 2538
3. วีรวัฒน์ ปิ่นท่วงกูร และ สราวุฒิ เอ็งอุทัยวัฒน์, “การเขียนโปรแกรมใช้งาน EGA/VGA”, ซีเอ็ดดูเคชั่น, 472 หน้า, 2536
4. John Iovine, “Step into virtual reality”, Windcret / Mcgraw-Hill , 310 p ,1994
5. Rex E. Bradford, “Real-Time Animation Toolkit in C++”, John Wiley & Sons, Inc. , 777 page, 1995