สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การประมวลผลภาพวิดีโอสามมิติ

3D Video Image Processing

โดย
นายพงษ์สิน จันทรพิพัฒน์
นายยุทธนา กฤตสัมพันธ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เลขหมู่.....
เลขทะเบียน 36926
วัน, เดือน, ปี 29 ส.ค. 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์เรื่อง	การประมวลผลภาพวิดีโอสามมิติ
	3D Video Image Processing
จัดทำโดย	นายพงษ์สิน จันทรพิพัฒน์
	นายยุทธนา กฤตสัมพันธ์
อาจารย์ที่ปรึกษา	รศ.ดร.รัตติกร วรากูลศิริพันธุ์

ปริญญานิพนธ์ฉบับนี้ได้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ..... อาจารย์ที่ปรึกษา

(รศ.ดร.รัตติกร วรากูลศิริพันธุ์)

วันที่ 17 / พ.ย. / 2563

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่องการสร้างและประมวลผลภาพวิดีโอสามมิตินี้ ได้จัดทำขึ้นเพื่อประกอบการศึกษาโครงงานการสร้างและประมวลผลภาพวิดีโอสามมิติ ซึ่งนับว่าเป็นโครงงานที่มีความทันสมัยต่อเทคโนโลยีอย่างมาก โดยการใช้เทคนิคแบบใหม่ในการประมวลผล ซึ่งผู้จัดทำขอขอบคุณ รศ.ดร.รัตติกร วรากูลศิริพันธุ์ อาจารย์ที่ปรึกษา และอาจารย์พารา ลิมมะณีประเสริฐ ที่ได้ให้คำปรึกษา คำแนะนำ ตลอดจนความช่วยเหลือในด้านทัศนอุปกรณ์เป็นอย่างดี จึงใคร่ขอขอบพระคุณมา ณ ที่นี้

พงษ์สิน จันทรพิ

(นายพงษ์สิน จันทรพิพัฒน์)

ยุทธนา กฤ

(นายยุทธนา กฤตสัมพันธ์)

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประมวลผลภาพวิดีโอสามมิติ

นายพงษ์สิน จันทรพิพัฒน์
นายยุทธนา กฤตสัมพันธ์
รศ.ดร.รัตติกร วรากูลศิริพันธุ์ (อาจารย์ที่ปรึกษา)
ภาคการศึกษาที่ 2 ปีการศึกษา 2542

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอโครงงานการสร้างและประมวลผลภาพวิดีโอสามมิติ ซึ่งองค์ประกอบของโครงงานมีทั้งฮาร์ดแวร์และซอฟต์แวร์ โดยฮาร์ดแวร์ประกอบด้วยวงจรแยกสัญญาณเบี่ยงเบนแนวตั้งจากการ์ดแสดงผลภาพบนจอมอนิเตอร์เพื่อนำมาควบคุมการชัตเตอร์ของแว่นตาสามมิติ ส่วนซอฟต์แวร์นั้นประกอบด้วยโปรแกรมแยกข้อมูลภาพวิดีโอสามมิติจากไฟล์ภาพที่ได้รับการบันทึกเป็นไฟล์ .AVI และโปรแกรมแสดงผลภาพวิดีโอสามมิติ โดยทั้งสองโปรแกรมได้พัฒนาให้ใช้งานบนระบบปฏิบัติการ Microsoft Windows 95/98 เพื่อความสะดวกในการใช้งาน ซึ่งผู้ใช้จะต้องเชื่อมต่อฮาร์ดแวร์เข้ากับคอมพิวเตอร์ และเรียกโปรแกรมใช้งาน จึงจะสามารถมองเห็นภาพสามมิติได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3D Video Image Processing

Mr.Phongsin Chantharaphiphat

Mr.Yuttana Krittasampan

Assoc.Prof.Dr.Ruttikorn Varakulsiripunth

(Advisor)

2nd Semestor,Education Year 1999

Abstract

In this paper, we present our project in Generating and Processing 3D Video. This project consist both hardware and software. For the hardware, we create the circuit to get the Vertical Sync signal from video monitor card in the computer to control 3D Glasses. In the software, we create two application, the Video Field Separater program and Video Playback program. Both developed to run in Microsoft Windows 95/98 for convenient in using. User will connect hardware to the computer and run these application, you will see 3D video image.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีความจริงเสมือน	2
2.1 ทฤษฎีความจริงเสมือน	2
2.2 แสงและการมองเห็น	2
2.3 การมองเห็นภาพสามมิติโดยธรรมชาติ	2
2.4 การมองเห็นภาพจากจอคอมพิวเตอร์หรือจากจอโทรทัศน์	2
2.5 หลักพื้นฐานในการสร้างภาพสามมิติ ในจอคอมพิวเตอร์ หรือจอโทรทัศน์	2
บทที่ 3 การสร้างภาพสามมิติบนจอมอนิเตอร์โดยวิธีลำดับภาพ	7
3.1 ทฤษฎีการสร้างภาพบนหน้าจอมอนิเตอร์	7
3.2 โหมด อินเทอร์แลซ (Interlaced Mode) และ นอนอินเทอร์แลซ (Non - Interlaced Mode) กับการแสดงภาพสามมิติ บนจอมอนิเตอร์	8
บทที่ 4 ขั้นตอนการดำเนินงานสร้างภาพวิดีโอสามมิติ	10
บทที่ 5 การออกแบบและพัฒนาซอฟแวร์ประมวลผลภาพวิดีโอสามมิติ	11
5.1 การออกแบบและพัฒนาซอฟต์แวร์	12
5.2 โปรแกรมจัดเรียงข้อมูลภาพวีดีโอสามมิติ	12
5.3 ขั้นตอนการทำงานของโปรแกรม	16
5.4 โปรแกรมแสดงข้อมูลภาพวีดีโอสามมิติ	18
5.5 ขั้นตอนการทำงานของโปรแกรม	18
บทที่ 6 อุปกรณ์ควบคุมการมองเห็นภาพสามมิติบนจอมอนิเตอร์	20
6.1 หลักการทำงานของการ์ดแสดงผล	20
6.2 สัญญาณที่ควบคุมการสแกนของการ์ดแสดงผลในโหมดอินเทอร์แลซ	21
6.3 วิธีการนำ สัญญาณซิงค์ทางแนวตั้ง มาใช้ควบคุมการมองเห็นภาพสามมิติ	22
6.4 วงจรในส่วนอุปกรณ์ควบคุมการชัตเตอร์	23
6.5 การทำงานของแว่นตาสามมิติ (3D Glasses) : LC แบบเซลล์เดียว	24
บทที่ 7 ผลการทดลอง	25
7.1 ภาพที่ได้จากการบันทึกเป็นไฟล์ .AVI	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 ภาพที่ได้จากการแยกข้อมูลภาพสองมุมมองออกจากกัน 25
7.3 ลำดับภาพที่เกิดขึ้นในการประมวลผลโดยสมอง 26
บทที่ 8 สรุปผลการทดลอง 27
ภาคผนวก
บรรณานุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 1.1 ขั้นตอนทั้งหมดของการสร้างภาพวิดีโอสามมิติของโครงงาน	1
รูปที่ 2.1 โครงสร้างของภาพสามมิติแบบแว่นตากรองแสงสีแดงและสีน้ำเงิน	3
รูปที่ 2.2 ตัวอย่างของภาพสามมิติแบบเหลื่อมกัน	4
รูปที่ 2.3 ลำดับขั้นตอนการมองภาพสามมิติแบบเหลื่อมกัน	4
รูปที่ 2.4 การสร้างภาพสามมิติในระบบสเตอริโอสโคปิค	5
รูปที่ 2.5 ลักษณะการเดินทางของคลื่นแสงโดยผ่านอุปกรณ์โพลาไรซ์	6
รูปที่ 3.1 แสดงการเก็บข้อมูลภาพลงสู่หน่วยความจำ	7
รูปที่ 3.2 แสดงการเก็บข้อมูลภาพฟิลด์คู่ลงสู่หน่วยความจำ	8
รูปที่ 3.3 แสดงการเก็บข้อมูลภาพฟิลด์คี่ลงสู่หน่วยความจำ	8
รูปที่ 4.1 ขั้นตอนทั้งหมดของการสร้างภาพสามมิติบนจอมอนิเตอร์	10
รูปที่ 5.1 ภาพสามมิติที่มีลักษณะซ้อนทับกันอยู่	11
รูปที่ 5.2 ภาพที่แยกมุมมองทั้งสองข้างให้อิสระกัน	
รูปที่ 5.2 ก ภาพฟิลด์คู่	11
รูปที่ 5.2 ข ภาพฟิลด์คี่	11
รูปที่ 5.3 การจัดเรียงข้อมูลในไฟล์ .AVI	13
รูปที่ 5.4 การจัดเรียงข้อมูลของรูปภาพ .AVI	13
รูปที่ 5.5 แผนผังขั้นตอนการทำงานของโปรแกรมแยกข้อมูลภาพวิดีโอสามมิติ	17
รูปที่ 5.6 แผนผังขั้นตอนการทำงาน ของโปรแกรมแสดงภาพวิดีโอสามมิติ	19
รูปที่ 6.1 แสดงการสร้างภาพบนจอมอนิเตอร์ด้วยหลอดรังสีคาร์โทด	21
รูปที่ 6.2 โหมดอินเทอร์แลซ และ โหมดนอนอินเทอร์แลซ	
รูปที่ 6.2 ก โหมด อินเทอร์แลซ	22
รูปที่ 6.2 ข โหมด นอนอินเทอร์แลซ	22
รูปที่ 6.3 ภาพสัญญาณซิงค์ทางแนวตั้งของการ์ดแสดงผล	22
รูปที่ 6.4 สัญญาณที่ออกจากอุปกรณ์ควบคุมการชัตเตอร์ ไปยังแว่น LC	23
รูปที่ 6.5 วงจรหารความถี่ลง 2 เท่าโดยใช้ ไอซีเบอร์ CD4017	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7.1 ภาพสามมิติซึ่งยังไม่ได้แยกฟิลด์ 25
รูปที่ 7.2 ภาพมุมมองจากตาขวา ซึ่งอยู่บนฟิลด์คู่ 25
รูปที่ 7.3 ภาพมุมมองจากตาซ้าย ซึ่งอยู่บนฟิลด์คี่ 26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

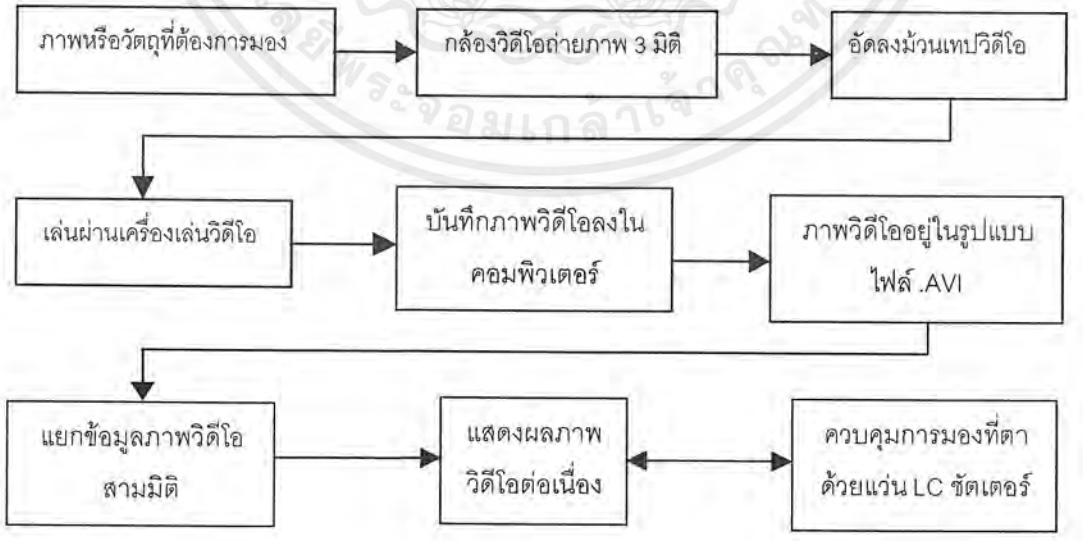
บทที่1
บทนำ

ในปัจจุบันเทคโนโลยีในการสร้างภาพวิดีโอสามมิติ เป็นเทคโนโลยีที่มีการพัฒนาด้วยเทคโนโลยีสมัยใหม่อยู่ตลอด โดยในปัจจุบันสามารถพัฒนาให้ถ่ายภาพวิดีโอสามมิติ โดยใช้กล้องวิดีโอเพียงตัวเดียว ซึ่งได้รับการพัฒนาจากการใช้กล้องวิดีโอ 2 ตัวที่บันทึกภาพคนละมุมมอง แต่การถ่ายด้วยกล้องเพียงตัวเดียวจะต้องมีอุปกรณ์เสริมเพิ่มไปที่กล้องเพื่อให้ได้คุณสมบัติที่สามารถมองภาพเป็นสองมุมมองที่แตกต่างกันให้ได้ในภาพวิดีโอที่บันทึก ซึ่งเป็นหลักการของการสร้างภาพวิโอสามมิติ

ในส่วนของโครงงานนี้ จะเป็นส่วนของการประมวลผลภาพวิดีโอสามมิติที่ได้จากการบันทึก แล้วนำภาพที่ได้มาประมวลผลและแสดงบนจอมอนิเตอร์ของคอมพิวเตอร์เพื่อให้รับชมภาพวิดีโอสามมิติได้

วัตถุประสงค์ของโครงงาน

1. ใช้ในงานด้านการนำเสนอผลงาน (Presentation) ในระบบภาพ 3 มิติ
2. ใช้ในงานด้านเทคโนโลยีสารสนเทศได้ เช่นการสร้างเว็บไซต์ในลักษณะของภาพ 3 มิติ
3. ใช้ในงานด้านระบบความจริงเสมือน (Virtual Reality)



รูปที่ 1.1 ขั้นตอนทั้งหมดของการสร้างภาพวิดีโอสามมิติของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีความจริงเสมือน

2.1 ทฤษฎีความจริงเสมือน

ความจริงเสมือน (Virtual Reality) หรือ VR ซึ่งมีบางคนกล่าวไว้ว่า คือ การผสมผสานของเทคโนโลยีหลากหลายรูปแบบเข้าด้วยกัน ไม่ว่าจะเป็นหน้ากากแสดงผล (Head Mounted Display) หรือ ถุงมืออิเล็กทรอนิกส์ (Data Glove) บางกลุ่มอาจจะเน้นไปทางด้านภาพยนต์หรือทางบันเทิง ซึ่งแท้ จริงแล้วความจริงเสมือนก็คือหนทางที่จะนำมนุษย์ไปสู่การมองเห็น การสัมผัสและการโต้ตอบกับคอมพิวเตอร์ ซึ่งมีข้อมูลที่ซับซ้อนได้อย่างเต็มที่ โดยที่การมองเห็นภาพนั้นจะใช้จอคอมพิวเตอร์หรือจอโทรทัศน์มาเป็นตัวแสดงผล

2.2 แสงและการมองเห็น

การมองเห็นวัตถุใดๆนั้นเกิดจากแสงจากแหล่งกำเนิดตกกระทบวัตถุแล้วสะท้อนมาเข้าตาของเรา โดยที่การทำงานของตากับการทำงานของประสาทตาจะทำงานสัมพันธ์กันโดยสมองจะประมวลผลภาพที่ตามองเห็นทำให้คนเข้าใจในสิ่งที่ตนมองเห็นว่าสิ่งนั้นคืออะไร

2.3 การมองเห็นภาพสามมิติโดยธรรมชาติ

การมองเห็นสิ่งของต่างๆรอบตัวเราจะเป็นการมองเห็นในลักษณะ 3 มิติอยู่แล้ว ซึ่งจะเห็นว่าสิ่งนั้นมีความกว้าง ความยาว และความลึกหรือความหนา โดยการมองเห็นสิ่งต่างๆได้ในลักษณะ 3 มิตินั้นเกิดจากการมองวัตถุใน สองมุมมองที่ต่างกันหรือว่าเหลื่อมกันเล็กน้อย เท่ากับระยะห่างของตานั่นเอง โดยสมองก็จะทำการประมวลผลรวม 2 ภาพดังกล่าวเข้าด้วยกัน ซึ่งทำให้เราเห็นสิ่งของต่างๆเป็น 3 มิตินั่นเอง

2.4 การมองเห็นภาพจากจอคอมพิวเตอร์หรือจากจอโทรทัศน์

จะมีการมองเห็นในลักษณะ 2 มิติโดยจะไม่มีความลึก หรือ ความหนา ซึ่งเนื่องจากไม่สามารถแสดงภาพออกมานอกจอ หรือว่าทะลุเข้าไปในจอได้ การมองเห็นจึงมีลักษณะแบนๆไปตามจอดังกล่าว

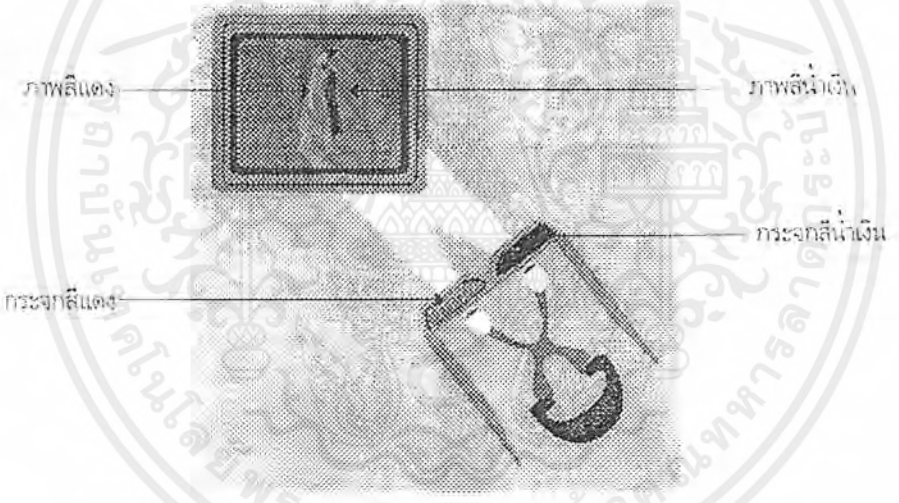
2.5 หลักพื้นฐานในการสร้างภาพสามมิติ ในจอคอมพิวเตอร์หรือจอโทรทัศน์

มีวิธีการทำด้วยกันหลายชนิด จะขอยกวิธีที่นิยมทำกันดังนี้

2.5.1 ภาพ 3 มิติแบบแว่นตากรองแสงสีแดงและน้ำเงิน วิธีการนี้จะใช้ความแตกต่างระหว่างภาพซ้ายกับภาพขวาที่มีความแตกต่างกันเพียงเล็กน้อย ซึ่งวิธีการแบบนี้จะแสดงภาพพร้อมกันที่ตาทั้ง 2 ข้างหรือเรียกอีกอย่างหนึ่งว่า Time-Parallel วิธีการดูภาพจะต้องใส่แว่นตาที่ข้างหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

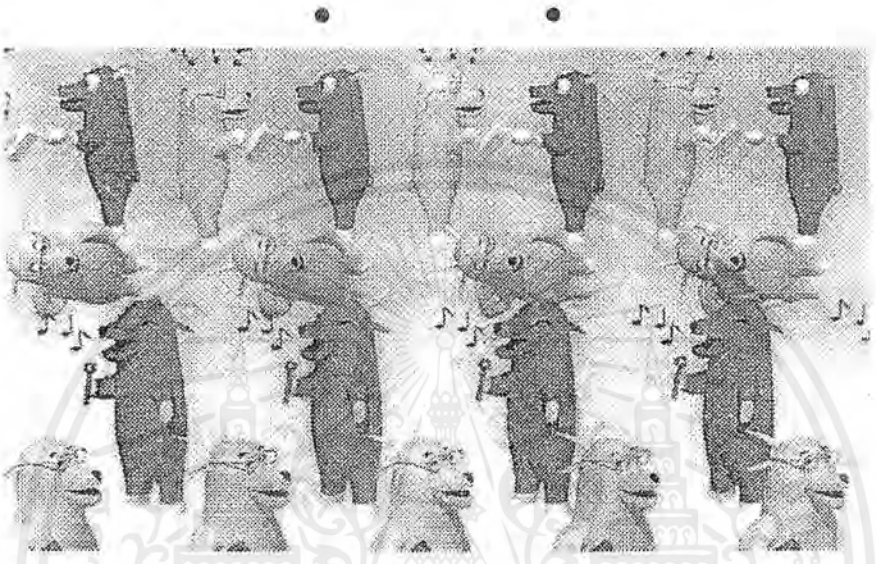
เป็นกระจกหรือกระดาษกรองแสงสีแดง และอีกข้างหนึ่งเป็นสีน้ำเงิน ซึ่งแว่นตาดังกล่าวนี้จะช่วยแยกภาพออกเป็น 2 ภาพโดยจะใช้สีของกระจก หรือกระดาษสีมาเป็นตัวกรองแสง เพื่อทำให้ตาทั้งสองข้างมองภาพอย่างอิสระต่อกันซึ่งโครงสร้างของภาพสามมิติแบบใช้แว่นตากรองแสงสีแดงและน้ำเงินแสดงดังรูปที่ 2.1 ถ้าลองถอดแว่นตาดังกล่าวออกแล้วทดลองดูภาพ จะเห็นได้ว่ามีภาพ 2 ภาพซ้อนกันอยู่ โดยภาพหนึ่งจะเป็นสีแดงและอีกภาพหนึ่งจะเป็นสีน้ำเงิน ซึ่งจากหลักการดังกล่าวนี้เองทำให้สามารถมองเห็นภาพเป็นภาพสามมิติโดยที่กระจกสีแดงจะเห็นภาพสีแดง และจะมองไม่เห็นภาพสีน้ำเงิน ส่วนที่กระจกสีน้ำเงินจะเห็นภาพสีน้ำเงินและไม่เห็นสีแดง จึงทำให้ปรากฏภาพสามมิติเกิดขึ้น แต่วิธีนี้มีข้อเสียคือ เมื่อดูภาพไปนาน ๆ จะทำให้เกิดอาการปวดตาขึ้นได้เนื่องจากกระตกสีที่กรองภาพ ไม่สามารถกรองภาพได้อย่างหมดจดจริงๆ ดังนั้น ตาแต่ละข้างจะเห็นเงาของอีกภาพหนึ่งเกิดขึ้นอยู่ด้วย ซึ่งจะเรียกปรากฏการณ์นี้ว่า Ghosting หรือ Cross Talk



รูปที่ 2.1 โครงสร้างของภาพสามมิติแบบแว่นตากรองแสงสีแดงและสีน้ำเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 ภาพสามมิติ แบบความเหลื่อมของภาพ จากรูปที่ 2.2 เป็นการสร้างภาพสามมิติในลักษณะที่ใช้ภาพ 2 ภาพมาทำให้เกิดภาพสามมิติ โดยภาพที่ใช้สร้างจะมีมุมมองที่แตกต่างกันเล็กน้อยทางด้านแนวนอน ซึ่งเทคนิคการมองภาพสามมิติด้วยตาเปล่านั้นมีเทคนิคคือ ให้มองจากด้านล่างของภาพในลักษณะที่ยึดจากจุด 2 จุดที่ใดก็ได้แต่ต้องอยู่บนภาพ มาเป็นตัวอ้างอิง



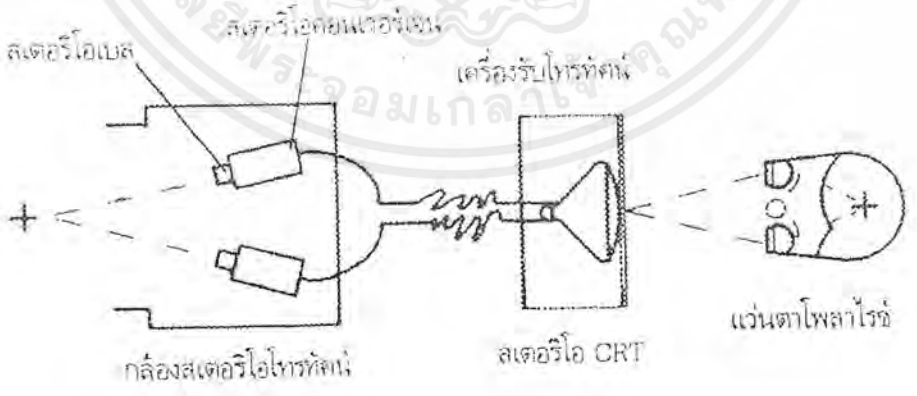
รูปที่ 2.2 ตัวอย่างของภาพสามมิติแบบเหลื่อมกัน

รูปที่ 2.3 ลำดับขั้นตอนการมองภาพสามมิติแบบเหลื่อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการมองเห็นภาพแบบเหลื่อมนี้ ต้องใช้ต้องใช้วิธีการมองภาพที่แตกต่างจากแบบแรกที่มองผ่านแว่นตากรองแสง สีแดงและสีน้ำเงิน การมองภาพทำได้โดยการบังคับสายตาแต่ละข้างให้เป็นอิสระ โดยการเพ่งสายตาไปในแนวตรงทั้ง 2 ข้างให้ขนานกัน การวางภาพให้วางห่างจากสายตาประมาณ 1 ช่วงแขน จากนั้นใช้สายตาทั้งสองข้างเพ่งมองไปยังภาพ โดยการมองแบบปกติก่อนแล้วค่อยๆ บังคับสายตาให้แยกแนวของสายตาออกจากกัน โดยตาข้างซ้ายเพ่งมองแนวตรงไปด้านซ้าย ตาด้านขวาเพ่งแนวตรงไปด้านขวา ถ้าสังเกตจุดด้านล่างจะเห็นเป็นจุดจากเดิมที่เป็น 2 จุด โดยที่มุมมองของภาพจากสายตาทั้งซ้ายและขวาจะเหมือนซ้อนเข้าหากัน ภาพที่เห็นจะเป็นภาพสามมิติ การมองภาพแบบนี้ไม่จำเป็นต้องใส่แว่นตาก็สามารถเห็นภาพสามมิติด้วยตาเปล่าได้ แต่จะมองได้ยากและต้องใช้เวลานานจึงจะสามารถมองเห็นเป็นภาพสามมิติได้

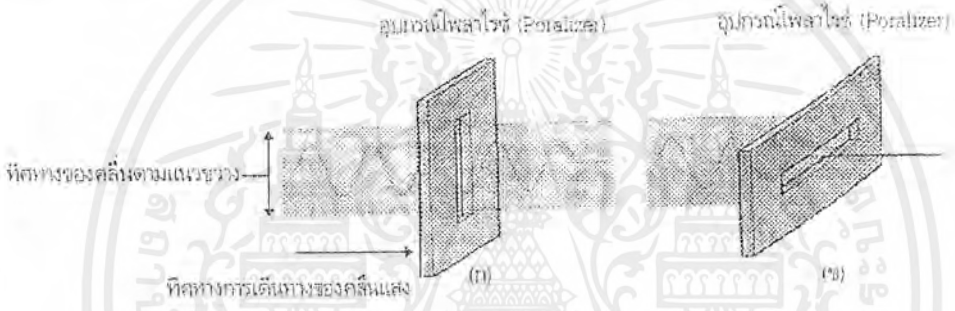
2.5.3 ภาพสามมิติ ระบบสเตอริโอสโคปิค หลักการทำงานของระบบนี้ภาพจะถูกเก็บด้วยกล้องโทรทัศน์สเตอริโอสโคปแบบพิเศษ ภายในจะประกอบไปด้วยกล้องโทรทัศน์สี 2 ชุด ที่หัวกล้องโทรทัศน์ สามารถจะปรับระยะห่างระหว่างจุด 2 จุดของภาพที่เก็บไว้ เรียกว่า สเตอริโอเบส จะสามารถปรับค่าความเข้มของภาพได้ โดยใช้อุปกรณ์ที่เรียกว่า สเตอริโอคอนเวอร์เจนต์ หลังจากบันทึกภาพที่ได้จากกล้องสเตอริโอสโคป สัญญาณของภาพทั้งสองจะถูกส่งผ่านไปยังทางด้านอินพุทของเครื่องรับโทรทัศน์ ซึ่งระบบโทรทัศน์ดังกล่าวนี้ จะเป็นโทรทัศน์ระบบพิเศษคือจะมีจอภาพ CRT ชนิดพิเศษ ซึ่งมี CRT 2 ชุดในหนึ่งจอภาพ กล่าวคือที่จอภาพ CRT ด้านหนึ่งจะแสดงภาพทางด้านแนวตั้งที่รับภาพมาจากกล้องสเตอริโอสโคป ตัวแรก และที่จอภาพ CRT อีกด้านหนึ่งจะแสดงภาพทางด้านแนวนอนที่รับภาพมาจากกล้องสเตอริโอสโคปของอีกตัวหนึ่ง



รูปที่ 2.4 การสร้างภาพ 3 มิติในระบบสเตอริโอสโคปิค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยลักษณะของจอภาพชนิดพิเศษนี้จะถูกเรียกว่า จอภาพระบบสเตอริโอ CRT การมองภาพผ่านทางจอภาพดังกล่าว จะต้องใช้อุปกรณ์ชนิดพิเศษคือ สามารถที่จะมองภาพทั้งสอง ได้อย่างอิสระต่อกัน ซึ่งจากหลักการที่ผ่านมา ถ้าสามารถมองภาพ 2 ภาพได้อย่างเป็นอิสระต่อกันแล้ว ภาพดังกล่าวจะสามารถเห็นเป็น ภาพสามมิติได้ ซึ่งจากหลักการดังกล่าวนี้เอง การที่ใช้อุปกรณ์มองภาพ 2 มุมมองอย่างอิสระต่อกันทำให้ภาพที่ได้เกิดเป็นภาพสามมิติ และอุปกรณ์ดังกล่าวนี้เรียกว่า แว่นตาโพลาไรซ์ (Poralize Glasses) ซึ่งหลักการทำงานของแว่นตาโพลาไรซ์คือ อุปกรณ์ที่สามารถจะยอมให้แสงผ่านไปในลักษณะเพียงด้านใดด้านหนึ่งเท่านั้น (แนวตั้ง กับ แนวนอน) โดยภายในแว่นตาโพลาไรซ์หนึ่งอันจะประกอบไปด้วยเลนส์โพลาไรซ์ (Poralize Lens) สองตัว ซึ่งการทำงานของเลนส์โพลาไรซ์มีหลักการดังรูปที่ 2.5



รูปที่ 2.5 ลักษณะการเดินทางของคลื่นแสงโดยผ่านอุปกรณ์โพลาไรซ์

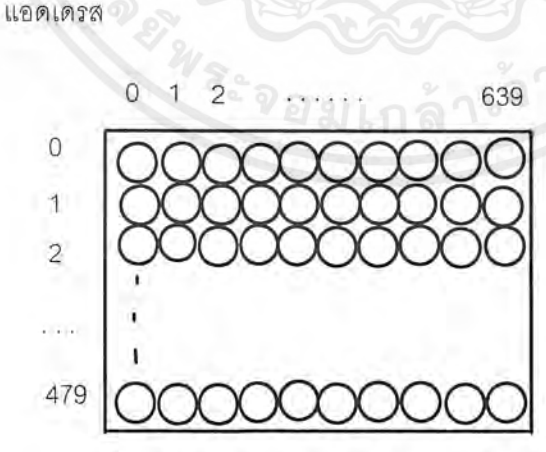
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การสร้างภาพ 3 มิติบนจอมอนิเตอร์โดยวิธีลำดับภาพ

3.1 ทฤษฎีการสร้างภาพที่หน้าจอคอมพิวเตอร์

จอคอมพิวเตอร์ หรือ จอมอนิเตอร์ คืออุปกรณ์ทางด้านเอาท์พุทของคอมพิวเตอร์ ที่มีหน้าที่แสดงผลลัพธ์ที่ได้จากการประมวลผล ให้แสดงออกมาเป็นตัวอักษร ข้อความ หรือ ภาพกราฟิก บนหน้าจอของ คอมพิวเตอร์ สำหรับจอมอนิเตอร์ของคอมพิวเตอร์นั้นจะมีอุปกรณ์ต่อพ่วง นั่นก็คือ การ์ดแสดงผล หรือ วีจีเอการ์ด มีหน้าที่ควบคุมการทำงานของจอมอนิเตอร์ เพื่อแสดงผลของข้อมูลได้อย่างถูกต้อง

ลักษณะการเก็บข้อมูลภาพลงสู่หน่วยความจำ จะมีการแสดงขนาดภาพที่จอมอนิเตอร์อยู่หลายโหมด เช่น 480 x 640 , 600 x 800 ,768 x 1024 พิกเซล (pixel) เป็นต้น โดยถ้าการจัดพื้นที่หน่วยความจำมีขนาด 480 x 640 หมายความว่า ในหนึ่งหน้าจอทั้งหมดจะแสดงจำนวนพิกเซล ทั้งหมด 307,200 พิกเซล โดยแบ่งเป็นทางด้านแนวตั้ง (Column) ทั้งหมด 640 คอลัมน์ และทางด้านแนวนอน (Row) ทั้งหมด 480 แถว ซึ่งการ์ดจอส่วนมากจะทำการสแกนโหมด 480 x 640 และ 600 x 800 อยู่ใน Non – Interlacing Mode และจะทำการสแกนโหมด 768 x 1024 และที่มีความละเอียดสูงกว่านี้ อยู่ใน Interlacing Mode คือถ้าโหมดในการแสดงภาพยิ่งมีความละเอียดมาก จำนวนจุด (pixel) ก็มีมากขึ้น การสแกนก็ต้องทำอยู่ใน Interlacing Mode เพื่อรองรับจำนวนจุดที่จะแสดงมาก ๆ ได้

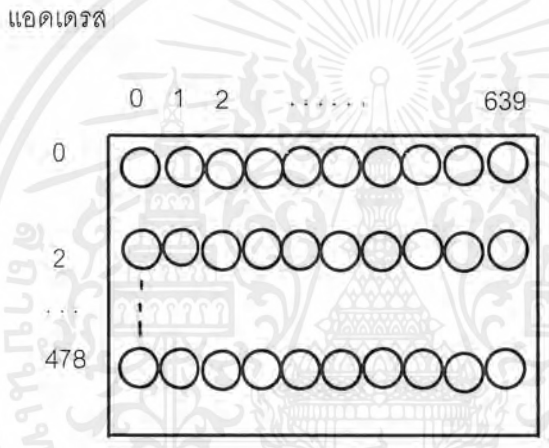


รูปที่ 3.1 แสดงการเก็บข้อมูลภาพลงสู่หน่วยความจำ

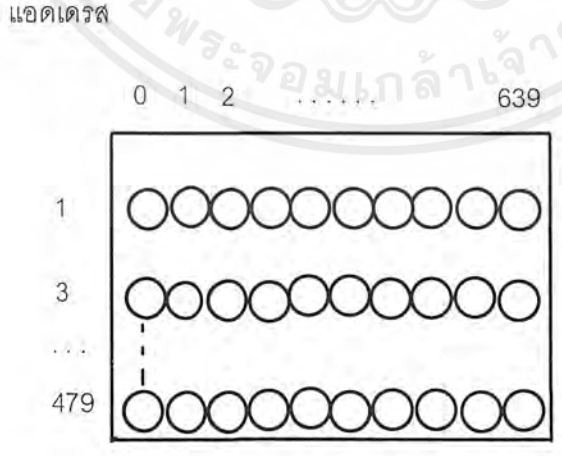
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 โหมด อินเทอร์แลซ (Interlaced Mode) และ นอนอินเทอร์แลซ (Non - Interlaced Mode) กับการแสดงภาพ 3 มิติ บนจอมอนิเตอร์

การแสกนแบบไม่สลับเส้นสแกน (Non – Interlacing Mode) มีวิธีการคือจะทำการสแกนภาพทั้งหมด 1 ภาพหรือเรียกว่า 1 เฟรม (Frame) ออกหน้าจอโดยไม่มีการแบ่งภาพออกเป็นฟิลด์ (Field) ย่อย ๆ ซึ่งจะทำให้ภาพมีความละเอียดต่ำกว่าแบบสลับเส้นสแกน (Interlacing Mode) โดยวิธีของการสแกนภาพโดยการสลับเส้นสแกน มีวิธีการคือ จะแบ่งภาพ 1 ภาพ หรือที่ 1 เฟรม ออกเป็นฟิลด์ย่อย ๆ 2 ฟิลด์ คือ ฟิลด์คู่และฟิลด์คี่ โดยเริ่มจากการสแกนภาพจาก ฟิลด์คี่ก่อนพอครบ 1 ฟิลด์ ก็จะสแกนต่อด้วย ฟิลด์คู่ แล้วก็สลับกันไปเรื่อยๆ



รูปที่ 3.2 แสดงการเก็บข้อมูลภาพฟิลด์คู่ลงหน่วยความจำ



รูปที่ 3.3 แสดงการเก็บข้อมูลภาพ ฟิลด์คี่ลงหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่หลักการของ Interlacing Scanning Mode สามารถพัฒนาให้ มีการมองเห็นภาพ 3 มิติในจอมอนิเตอร์ได้ ซึ่งหลักการพื้นฐานของการมองภาพ 3 มิติซึ่งได้กล่าวไปแล้วในบทที่ 2 ก็คือ การทำให้ตาทั้งสองข้างมองภาพ 2 ภาพที่เหลื่อมกันตามแนวระนาบในมุมมองที่เป็นอิสระต่อกัน โดยตาข้างซ้ายจะมองเห็นภาพวัตถุทางด้านซ้าย และตาข้างขวาจะมองเห็นภาพวัตถุทางด้านขวา ซึ่งก็คือใน 1 ภาพถ้าทำให้ประกอบด้วย 2 ภาพย่อยโดย ภาพย่อยแรกคือภาพที่มองจากตาด้านซ้ายด้านเดียว และอีกภาพย่อยหนึ่ง คือภาพที่มองจากตาด้านขวาเพียงด้านเดียว

วิธีการต่อมาก็คือ ทำการแมทช์ (Match) กันระหว่าง ภาพย่อยจากตาซ้าย กับ ภาพฟิลด์คี่ (หรือ ฟิลด์คู่) และ ภาพย่อยจากตาขวา กับ ภาพฟิลด์คู่ (หรือ ฟิลด์คี่) หลังจากนั้นก็ทำการแสดงออกหน้าจอมอนิเตอร์ในโหมดอินเทอร์แลส ซึ่งจะต้องแสดงออกที่ละฟิลด์ สลับกันโดยใช้การเขียนโปรแกรมควบคุม

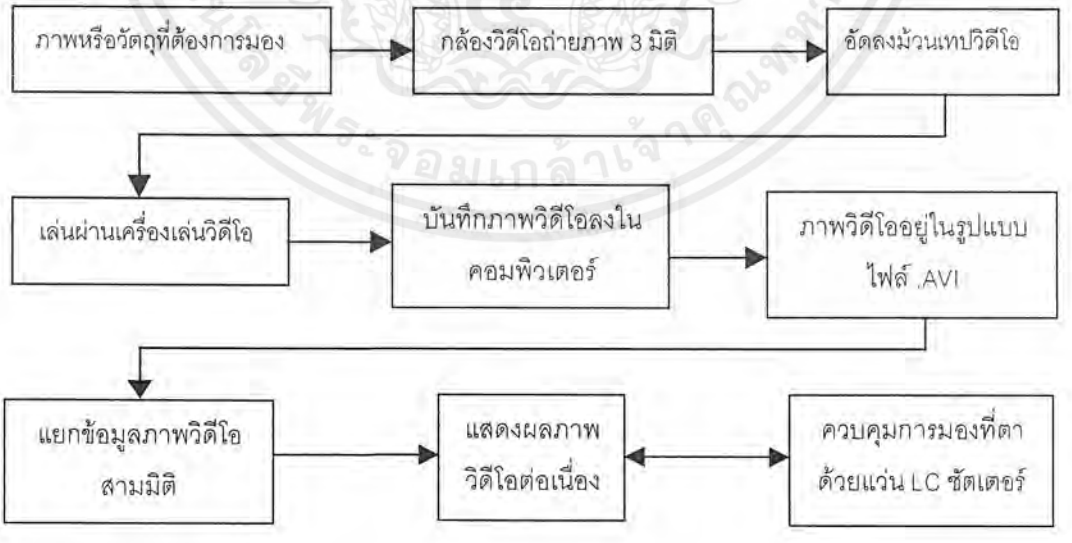
วิธีการต่อมาในส่วนควบคุมการมองเห็น คือต้องทำให้การมองเห็นของตาทั้งสองข้างเป็นอิสระซึ่งกันและกัน โดยเมื่อทำให้ตาข้างซ้ายเห็นภาพฟิลด์คี่ที่แสดงออกมาที่จอมอนิเตอร์ ซึ่งจะต้องไม่ให้ตาข้างซ้ายเห็นภาพของฟิลด์คู่โดยเด็ดขาด และในทางกลับกัน ก็ต้องทำให้ตาข้างขวาเห็นภาพฟิลด์คู่ที่แสดงออกมาที่จอมอนิเตอร์ ซึ่งจะต้องไม่ให้ตาข้างขวาเห็นภาพของฟิลด์คี่โดยเด็ดขาด โดยจะกล่าวรายละเอียดในบทที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ขั้นตอนการดำเนินงานสร้างภาพวิดีโอสามมิติ

ขั้นตอนการดำเนินงานในโครงงานการสร้างและประมวลผลภาพวีดีโอสามมิติมีอยู่ทั้งหมด 4 ขั้นตอนคือ

1) การรับข้อมูลภาพ (Video Capture) เป็นการรับข้อมูลภาพวีดีโอสามมิติที่ได้รับการบันทึกลงบนวีดีโอเทป โดยผ่าน Video Capture Card ภาพที่ได้รับการบันทึกจะอยู่ในรูปแบบไฟล์ .AVI (Audio Video Interleaved) ซึ่งเป็นรูปแบบไฟล์วีดีโอมาตรฐานบน Microsoft Windows
2) การจัดเรียงข้อมูลภาพ (Field Separater) เป็นการแยกข้อมูลภาพที่ได้จากการบันทึกภาพจากสองมุมมองออกจากกัน เนื่องจากภาพที่ได้มีลักษณะทับซ้อนกันอยู่ รายละเอียดจะกล่าวถึงในหัวข้อถัดไป
3) การแสดงผลข้อมูลภาพ (Video Playback) เป็นการแสดงผลภาพที่ได้จากการประมวลผลในขั้นตอนที่ผ่านมาในลักษณะภาพต่อเนื่อง ซึ่งมีลักษณะเช่นเดียวกับโปรแกรม Video Playback ทั่วไป เช่น Media Player เป็นต้น
4) การสร้างอุปกรณ์ควบคุมการชัตเตอร์ของแว่นตาสามมิติ (3D Glasses Shuttering Controller) เป็นการสร้างชุดอุปกรณ์ควบคุมการโปร่งแสง/ทึบแสงของ LC (Liquid Crystal) ภายในแว่นตาเพื่อการซิงโครไนซ์กับการแสดงผลบนหน้าจอคอมพิวเตอร์



รูปที่ 4.1 ขั้นตอนทั้งหมดของการสร้างภาพสามมิติบนจอมอนิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การออกแบบและพัฒนาซอฟแวร์ประมวลผลภาพวิดีโอสามมิติ

จากที่ได้กล่าวไว้ในขั้นตอนการดำเนินการของโครงงานนี้ว่า ภาพที่ได้จากวีดีโอเทป เป็นภาพสามมิติที่มีลักษณะซ้อนกันของภาพทั้งสองมุมมองแบบฟิลด์ต่อฟิลด์ ดังรูปที่ 5.1

รูปที่ 5.1 ภาพสามมิติที่มีลักษณะซ้อนทับกันอยู่

จากรูปที่ 5.1 จะเห็นว่าภาพที่ได้จากสองมุมมองมีการทับซ้อนกัน โดยภาพที่ได้จากมุมมองทางด้านซ้ายจะแสดงอยู่บนฟิลด์คู่ (กลุ่มของจุดสีแถวคู่) และภาพที่ได้จากมุมมองทางด้านขวาจะแสดงอยู่บนฟิลด์คี่ (กลุ่มของจุดสีแถวคี่) ซึ่งทั้งสองภาพนั้นสามารถแยกออกจากกันได้ ดังที่แสดงในรูปที่ 5.2

รูปที่ 5.2 ก ภาพฟิลด์คู่

รูปที่ 5.2 ข ภาพฟิลด์คี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1 การออกแบบและพัฒนาซอฟต์แวร์ จะแบ่งออกเป็น 2 ขั้นตอนคือ

5.1.1 โปรแกรมจัดเรียงข้อมูลภาพวีดีโอสามมิติ ทำหน้าที่แยกข้อมูลภาพทั้งสองฟิลด์ออกจากกันทุกเฟรมแล้วทำการจัดเรียงข้อมูลใหม่ โดยบันทึกกลับเป็นไฟล์ข้อมูลใหม่ในอีกรูปแบบหนึ่ง

5.1.2 โปรแกรมแสดงข้อมูลภาพวีดีโอสามมิติ ทำหน้าที่นำภาพที่ได้จากการบันทึกมาแสดงแบบเฟรมต่อเฟรม โดยภาพฟิลด์คู่จะแสดงอยู่บนเส้นสแกนแถวคู่ และภาพฟิลด์คี่จะแสดงอยู่บนเส้นสแกนแถวคี่ ต่อเนื่องกันไป

โดยทั้งสองโปรแกรมดังที่กล่าวไว้นั้น ได้รับการออกแบบและพัฒนาแบบ Object Oriented (OO) ซึ่งเป็นเทคโนโลยีในการพัฒนาโปรแกรมเชิงวัตถุ โดยใช้ Visual C++ 6.0 ในการพัฒนาโปรแกรมใช้งานบนระบบปฏิบัติการ Microsoft Windows 95/98 ซึ่งรายละเอียดของโปรแกรมนั้นจะกล่าวถึงในหัวข้อถัดไป และสำหรับเทคนิคที่สำคัญอย่างยิ่งที่ใช้ในการพัฒนาซอฟต์แวร์คือ การใช้โปรแกรม HexEditor ซึ่งเป็นโปรแกรมที่ใช้ในการ View Code ของไฟล์ทุกรูปแบบ ซึ่งสามารถใช้ในการตรวจสอบ/แก้ไข ข้อมูลในไฟล์ภาพได้ ซึ่งวิธีการนี้นับว่าเป็นประโยชน์อย่างยิ่งในการออกแบบอัลกอริทึมในการพัฒนาโปรแกรม

5.2โปรแกรมจัดเรียงข้อมูลภาพวีดีโอสามมิติ

หลักการทำงาน ภาพต่อเนื่องที่ได้จากโปรแกรมบันทึกภาพ จะถูกจัดเก็บอยู่ในรูปแบบไฟล์ .AVI ซึ่งมีวิธีการจัดเก็บข้อมูลทั้งแบบบีบอัดและไม่บีบอัด ในที่นี้จะนำไฟล์ที่ได้รับการบันทึกแบบไม่บีบอัดมาจัดเรียงข้อมูลใหม่ โดยการจัดเก็บจะมีลักษณะเดียวกับไฟล์ .BMP แบบเฟรมต่อเฟรม ซึ่งจะขออธิบายรูปแบบการจัดเก็บไฟล์ .BMP และ .AVI ดังต่อไปนี้

ไฟล์ข้อมูลภาพ BMP รายละเอียดของข้อมูลภาพจะแบ่งเป็น 3 ส่วนคือ

1. ส่วนหัวไฟล์ (File Header) มีรายละเอียดดังต่อไปนี้

- Btype เป็นค่า BMP Structure ต้องมีค่าเท่ากับ 19778 ขนาด 2 ไบต์
- Bsize ค่าของขนาดไฟล์โดยรวม ขนาด 4 ไบต์
- Reserved1 ไม่ใช้ (ค่าเท่ากับ 0) ขนาด 2 ไบต์
- Reserved2 ไม่ใช้(ค่าเท่ากับ 0) ขนาด 2 ไบต์
- BoffBit ระยะห่างเป็นไบต์จากส่วนหัวไฟล์ถึงข้อมูลรูปตัวแรก

2. ส่วนข้อมูลของส่วนหัวไฟล์ (File Information Header) มีรายละเอียดดังต่อไปนี้

- BISize ขนาดของส่วน BMP Info Header เป็นไบต์ ขนาด 4 ไบต์
- Bwidth ค่าความกว้างของรูป ขนาด 4 ไบต์
- Bheight ค่าความสูงของรูป ขนาด 4 ไบต์
- Bplane จำนวนเพลนที่ใช้ (ต้องเท่ากับ 1) ขนาด 2 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- BBitPP จำนวนบิตต่อจุดสี ขนาด 2 ไบต์
- Bcomp วิธีบีบอัดข้อมูลที่ใช้ (0 คือไม่มีการบีบอัด) ขนาด 4 ไบต์
- BimSize ขนาดของข้อมูลส่วนรูปภาพเป็นไบต์ ขนาด 4 ไบต์
- BXRes ความละเอียดในแกน X (จุดต่อนิ้ว) ขนาด 4 ไบต์
- BYRes ความละเอียดในแกน Y (จุดต่อนิ้ว) ขนาด 4 ไบต์
- BCUsed จำนวนสีที่ใช้ในรูปภาพ ขนาด 4 ไบต์
- BClmp จำนวนสีที่สำคัญ (0 คือสำคัญทุกสี) ขนาด 4 ไบต์
- Pal พาเลตต์ของสี

3. ส่วนข้อมูลภาพ (Picture Data) ข้อมูลในส่วนของรูปภาพจะถูกเก็บเป็นกลุ่มๆ แต่ละกลุ่มจะมีข้อมูลสำหรับหนึ่งแถว กลุ่มของข้อมูลจะเก็บในลักษณะจากล่างขึ้นบน หมายความว่าข้อมูลของแถวสุดท้ายจะอยู่ก่อนแถวที่อยู่ข้างบนและแถวบนสุดจะอยู่หลังสุดในไฟล์ เพื่อให้เกิดความเข้าใจยิ่งขึ้น สามารถอธิบายได้ดังรูปที่ 5.3 และ 5.4

11111111111111111111	55555555555555555555
22222222222222222222	44444444444444444444
33333333333333333333	33333333333333333333
44444444444444444444	22222222222222222222
55555555555555555555	11111111111111111111

รูปที่ 5.3 การจัดเรียงข้อมูลในไฟล์ รูปที่ 5.4 การจัดเรียงข้อมูลของรูปภาพ

ไฟล์ข้อมูลภาพวีดีโอ AVI พัฒนาโดยบริษัทไมโครซอฟต์ มีรูปแบบการจัดเก็บข้อมูลลักษณะเช่นเดียวกับไฟล์ BMP แบบเฟรมต่อเฟรม สำหรับโครงสร้างข้อมูลสามารถอธิบายได้โดยรูปแบบของภาษาซีดังนี้

```
RIFF ('AVI'
        LIST ('hdrl'
                'avih' (<Main AVI Header>)
        LIST ('strl'
                'strh'(<Stream header>)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    'strf' (<Stream format>)
                    'strd' (<additional header data>)
                        .
                        .
                        .
                )
                .
                .
                .
)
LIST ('movi'
                {SubChunk | LIST ('rec'
                    SubChunk1
                    SubChunk2
                    .
                    .
                    .
                    )
                    .
                    .
                    .
                }
                .
                .
                .
        )
        ['idx1'<AVIIndex>]
)
```

จะเห็นว่าโครงสร้างข้อมูลของไฟล์ประกอบด้วยส่วนหลัก 3 ส่วนคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. 'RIFF' เป็นส่วนที่เก็บข้อมูลจำเพาะของไฟล์ เช่น ความกว้างของภาพ ความสูงของภาพ ขนาดสี จำนวนเฟรม ภายใน RIFF ประกอบด้วย

1.1) 'hdrl' ประกอบด้วย Main AVI Header ซึ่งมีโครงสร้างข้อมูลดังนี้

```
typedef struct {
        DWORD       dwMicroSecPerFrame ;
        DWORD       dwMaxBytesPerSec ;
        DWORD       dwReserved1 ;
        DWORD       dwFlags ;
        DWORD       dwTotalFrames ;
        DWORD       dwInitialFrames ;
        DWORD       dwStreams ;
        DWORD       dwSuggestedBufferSize ;
        DWORD       dwWidth ;
        DWORD       dwHeight ;
        DWORD       dwScale ;
        DWORD       dwRate ;
        DWORD       dwStart ;
        DWORD       dwLength ;
    } MainAVIHeader ;
```

1.2) 'strl' ประกอบด้วย Stream List แบ่งออกเป็น 3 ส่วนคือ

1.2.1) 'strh' หรือ Stream Header มีโครงสร้างข้อมูลดังนี้

```
typedef struct {
        FOURCC      fccType ;
        FOURCC      fccHandler ;
        DWORD       dwFlags ;
        DWORD       dwReserved1 ;
        DWORD       dwInitialFrames ;
        DWORD       dwScale ;
        DWORD       dwRate ;
        DWORD       dwStart ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    DWORD       dwLength ;
    DWORD       dwSuggestedBufferSize ;
    DWORD       dwQuality ;
    DWORD       dwSampleSize ;
} AVIStreamHeader ;
```

ซึ่ง DWORD หมายถึงข้อมูลชนิด Double Word ขนาด 4 ไบต์และ FOURCC หมายถึงข้อมูลชนิด Character จำนวน 4 ตัวอักษร

1.2.2) 'strf' หรือ Stream Format เป็นตัวบ่งบอกลักษณะของข้อมูลว่าเป็นข้อมูลภาพหรือเสียง ถ้าเป็นข้อมูลภาพ ข้อมูลจะอยู่ในรูปแบบโครงสร้างของไฟล์ .BMP

1.2.3) 'strd' หรือ Additional header data เป็นข้อมูลเพิ่มเติมในส่วนของ header

2. 'movi' เป็นส่วนที่เก็บข้อมูลของภาพในแต่ละเฟรม ลักษณะเช่นเดียวกับการเก็บไฟล์ .BMP ทั้งแบบบีบอัดและไม่บีบอัดทุกประการ แต่ละเฟรมจะถูกคั่นด้วยรหัส '00db' เป็นการบ่งบอกการสิ้นสุดในแต่ละเฟรม

3. 'idx1' เป็นส่วนท้ายสุดของไฟล์ เป็นตัวบ่งบอกค่า Address Offset ของในแต่ละเฟรม

5.3 ขั้นตอนการทำงานของโปรแกรม ดังที่กล่าวไว้ข้างต้นว่า ไฟล์ภาพแบบ AVI มีลักษณะการจัดเก็บเช่นเดียวกับไฟล์ภาพแบบ BMP แบบเฟรมต่อเฟรม ฟังก์ชันหลักในการทำงานประกอบด้วย

5.3.1 OpenAVIFile เป็นฟังก์ชันที่ใช้ในการอ่านจำนวนเฟรม ความกว้างของภาพ ความสูงของภาพ ขนาดของสีที่ใช้ต่อหนึ่งจุดสี ซึ่งข้อมูลที่ได้จะถูกนำมาใช้ในการประมวลผลต่อไป

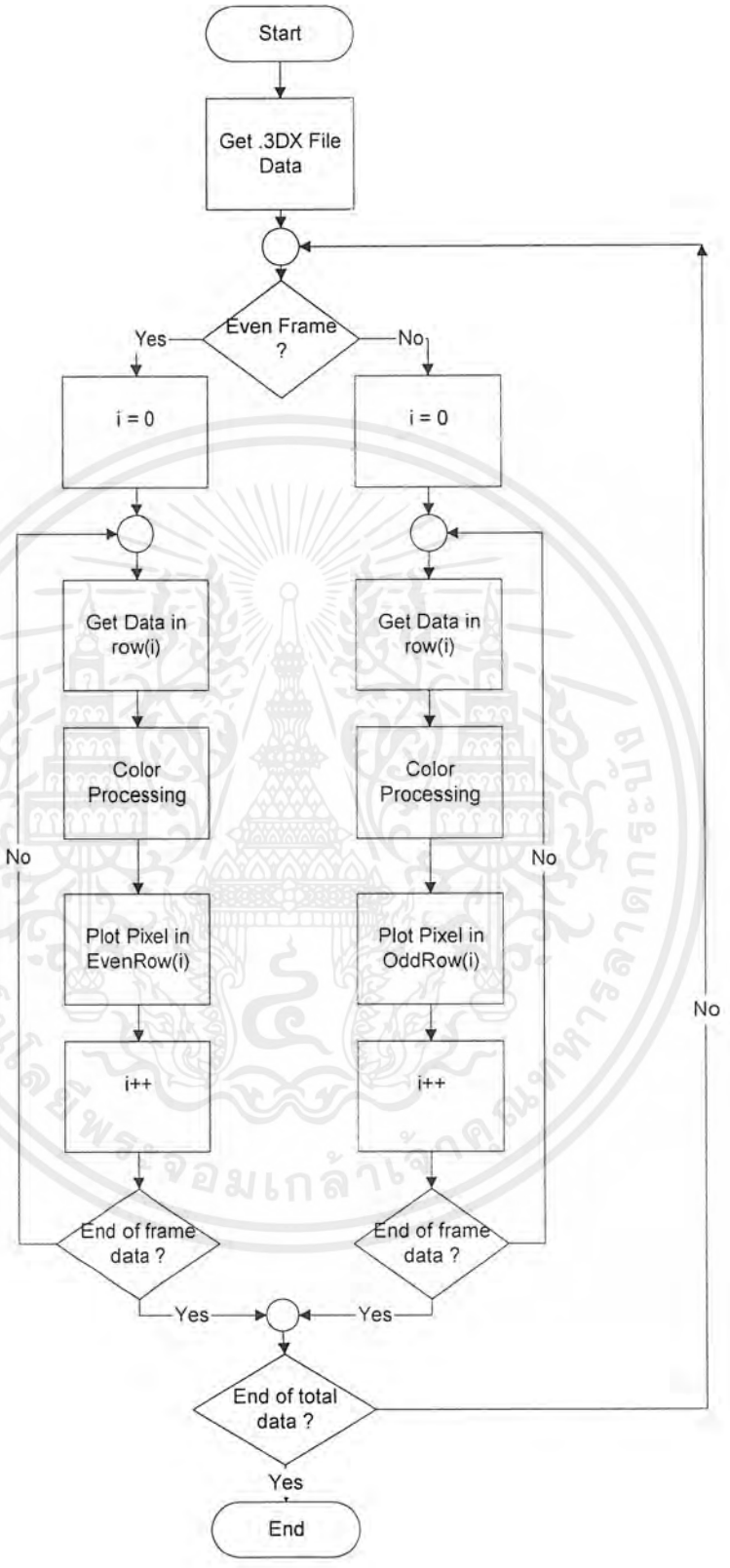
5.3.2 Encode3DAVI เป็นฟังก์ชันที่ใช้ในการแยกฟิลด์ภาพทางด้านซ้ายและด้านขวาออกจากกันต่อภาพหนึ่งเฟรม โดยภาพที่แยกออกมาจะมีความสูงเป็นครึ่งหนึ่งของภาพเดิมโดย ทำเช่นนี้ทุกๆเฟรม สำหรับขนาดของภาพในแต่ละเฟรมสามารถคำนวณได้จาก ความกว้าง*ความสูง* จำนวนบิตต่อจุดสี ซึ่งในที่นี้จะใช้ภาพขนาด 640*480*24 ภาพฟิลด์คู่และฟิลด์คี่จะถูกเก็บไว้ใน Buffer แล้วนำไปบันทึกเป็นไฟล์รูปแบบใหม่ต่อไป

1.SaveFile เป็นฟังก์ชันที่ทำหน้าที่บันทึกข้อมูลของภาพชนิดใหม่คือ จำนวนเฟรม ความกว้างของภาพ ความสูงของภาพ ขนาดของสี และข้อมูลที่เก็บไว้ใน Buffer เป็นไฟล์ .3DX

แผนผังขั้นตอนการทำงานของโปรแกรมแสดงในรูปที่ 5.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 5.5 แผนผังขั้นตอนการทำงานของโปรแกรมแยกข้อมูลภาพวิดีโอสามมิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
36926

5.4 โปรแกรมแสดงข้อมูลภาพวีดีโอสามมิติ

หลักการทำงานโปรแกรมถอดรหัสข้อมูล หรือโปรแกรมแสดงภาพต่อเนื่อง (VideoPlayback) ทำหน้าที่ นำภาพที่ได้ทำการแยกข้อมูลออกมา ซึ่งได้รับการบันทึกเป็นไฟล์ .3DX ออกมาแสดงให้ดู โดยภาพฟิลด์คู่จะแสดงบนเส้นสแกนเส้นคู่ และภาพฟิลด์คี่จะแสดงบนเส้นสแกนเส้นคี่ โดยทั้งนี้ทั้งนั้นการทำงานของจอภาพจะต้องทำงานอยู่ใน Interrace Mode ซึ่งเป็นโหมดการทำงานที่มีการสแกนภาพทีละฟิลด์สลับกันไป โดยจังหวะในการสแกนภาพจะเป็นจังหวะเดียวกับการมองเห็นภาพในมุมมองนั้นๆผ่านแว่นตาสามมิติ ซึ่งการมองเห็นนั้นจะเป็นอิสระต่อกันทั้งตาซ้ายและตาขวา ภาพที่ได้จะถูกนำไปประมวลผลเพื่อทำให้เกิดการมองเห็นเป็นภาพสามมิติต่อไป

5.5 ขั้นตอนการทำงานของโปรแกรม

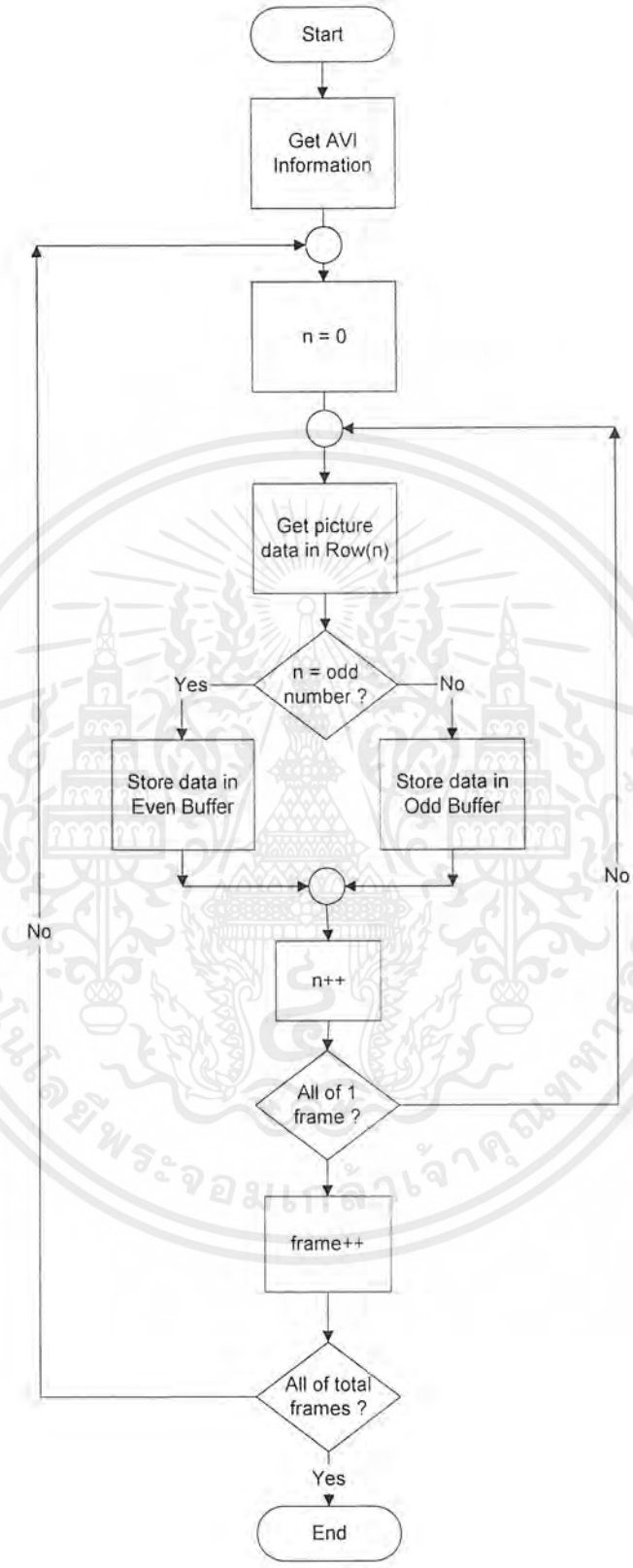
โปรแกรมแสดงข้อมูลภาพวีดีโอสามมิติประกอบด้วยฟังก์ชันหลักในการทำงานดังนี้

5.5.1 OpenFileInfo เป็นฟังก์ชันที่ใช้ในการอ่านข้อมูลที่จะนำมาใช้ในการประมวลผล ประกอบด้วย ความกว้างของภาพ ความสูงของภาพ จำนวนเฟรม ขนาดสี

5.5.2 OpenAVI เป็นฟังก์ชันที่ใช้ในการอ่านข้อมูลของแต่ละจุดสี โดยแต่ละจุดสีที่มองเห็นจะประกอบด้วยแม่สีแสง คือ สีแดง สีเขียว สีน้ำเงิน ตัวอย่างเช่น ถ้าแต่ละจุดสีประกอบด้วย สีแดง สีเขียว สีน้ำเงินอย่างละ 8 บิต เราสามารถมองเห็นสีได้ถึง 2^{24} สี หรือ 16.7 ล้านสี (True Color)

5.5.3 PlotEvenPel และ PlotOddPel เป็นฟังก์ชันที่ใช้ในการแสดงจุดสีบนหน้าจอ ภายในฟังก์ชันประกอบด้วยชุดคำสั่งในการผสมสี การแสดงจุดบนหน้าจอ โดยเรียกใช้การทำงานของ Microsoft DirectX SDK Version 6.0 ซึ่งเป็น Software Development Kit ที่ใช้ในการติดต่อและทำงานเกี่ยวกับ Computer Graphic พัฒนาโดยบริษัทไมโครซอฟต์ ซึ่งภาพฟิลด์คู่จะแสดงอยู่บนจุดสีแถวคู่ และภาพฟิลด์คี่จะแสดงอยู่บนจุดสีแถวคี่ แบบเฟรมต่อเฟรมต่อเนื่องกันไป แผนผังขั้นตอนการทำงานของโปรแกรมแสดงในรูปที่ 5.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 แผนผังขั้นตอนการทำงาน ของโปรแกรมแสดงภาพวิดีโอสามมิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
อุปกรณ์ควบคุมการมองเห็นภาพสามมิติบนจอมอนิเตอร์

จากวิธีการสร้างภาพ 3 มิติ ที่ได้กล่าวมาแล้วในบทที่ 3 โดยวิธีการลำดับภาพฟิลด์คี่ เข้าสู่ตาข้างซ้าย และฟิลด์คู่เข้าสู่ตาข้างขวา ซึ่งในบทนี้จะพูดถึงรายละเอียดทั้งหมด

6.1 หลักการทำงานของการ์ดแสดงผล

หรือ วีจีเอการ์ด ทำหน้าที่แปลงคำสั่งที่ได้จากซีพียู ให้อยู่ในรูปแบบสัญญาณที่จอมอนิเตอร์เข้าใจ โดยโครงสร้างของการ์ดแสดงผล (VDO Graphic Array) มีมาตรฐานในการแสดงผลของจอภาพซึ่งบริษัทผู้ผลิตได้กำหนดไว้ โดยมีหลายรูปแบบดังนี้ คือ 640 x 480 จุดภาพ แบบ 16 สีและ 2 สี , 720 x 400 จุดภาพ แบบ 16 สี และโมโนโครมเฉพาะตัวอักษร 360 x 400 จุดภาพแบบ 16 สี แบบตัวอักษร 320 x 200 จุดภาพ แสดงสีได้ถึง 250 สี ในโหมดการสแกน 200 เส้น ข้อมูลได้รักการสแกนซ้ำ 2 ครั้ง ดังนั้นข้อมูลรายละเอียดภายในจึงเป็นแบบ 400 เส้น

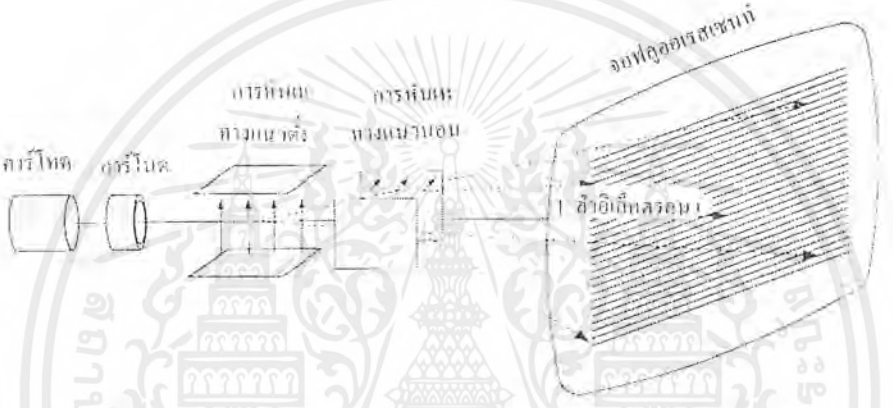
โครงสร้างของวีจีเอการ์ด สามารถอธิบายได้คร่าวๆคือ การ์ดแสดงผลจะใช้หน่วยความจำเก็บตามแพลนสี ซึ่งเมื่อนำมารวมกันจะได้จำนวนสีมากมายหลายสี ส่วนในโหมดแสดงแบบตัวอักษรจะให้ซีพียูเขียนรหัสแอสกีลงมายังหน่วยความจำ และวีจีเอจะนำไปแสดงผลโดยนำตัวอักษรจากรอม (ROM) มาแสดงผล การแสดงผลจะเป็นการนำเอาข้อมูลจากหน่วยความจำมาจัดใหม่ ให้อยู่ในรูปแบบที่ใช้ในการแสดงผล แล้วจึงส่งให้วงจรแปลงสัญญาณอนาล็อก สีแดง เขียว และน้ำเงิน เพื่อส่งต่อให้กับมอนิเตอร์สีอีกครั้งหนึ่งและหากต้องการแสดงผลแบบโมโนโครมก็ทำคล้ายกัน คือ นำข้อมูลทั้งหมดมาจัดรูปแบบใหม่ แล้วให้วงจรแปลงสัญญาณดิจิตอลเป็นอนาลอกแปลงสัญญาณออกทางเอาต์พุตแบบอนาลอกเฉพาะสีเขียวเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 สัญญาณที่ควบคุมการสแกนของการ์ดแสดงผลในโหมดอินเทอร์แลซ

สัญญาณที่ใช้ในการสแกนภาพในจอมอนิเตอร์มี 2 ชนิดคือ สัญญาณซิงค์ทางแนวตั้ง (Vertical Sync) และสัญญาณซิงค์ทางแนวนอน (Horizontal Sync)

การแสดงภาพ (รวมทั้งข้อความ หรือ กราฟิกส์) บนจอมอนิเตอร์ มีลักษณะคล้ายกับวิธีที่ใช้ในทีวี จากรูปที่ 6.1 แสดงให้เห็นถึงการสร้างภาพ ด้วยหลอดรังสีคาร์โทด (Cathode ray tube : CRT)

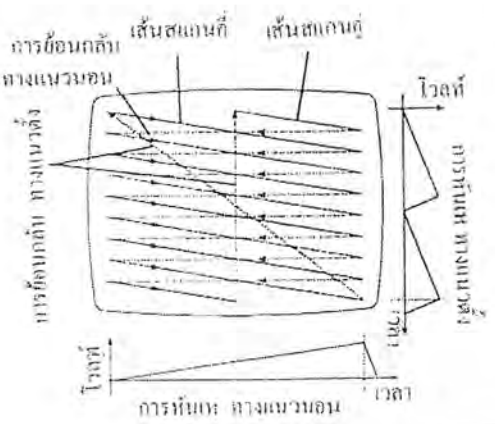


รูปที่ 6.1 แสดงการสร้างภาพบนจอมอนิเตอร์ด้วยหลอดรังสีคาร์โทด

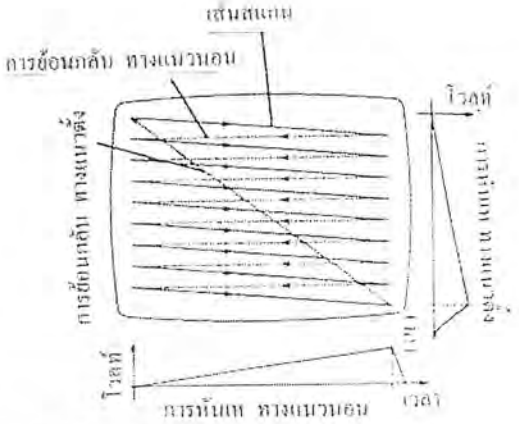
โดยหน้าจอจะแบ่งเป็นหลาย ๆ แถวในแนวนอนซึ่งแต่ละแถวจะประกอบด้วยจุดที่เรียกว่าพิกเซล (Pixel) กับหลายๆแถวในแนวตั้ง ซึ่งได้กล่าวมาแล้วในบทที่ 2 ในกรณีที่ลำอิเล็คตรอนมาถึงตำแหน่งมุมด้านขวาล่างสุดของแถว จะต้องทำให้ย้อนกลับไปเริ่มต้นในแถวถัดไปใหม่ เรียกขบวนการนี้ว่า การย้อนกลับทางแนวนอน (Horizontal retrace) ซึ่งจะใช้ สัญญาณซิงค์ทางแนวนอน (Horizontal Sync Signal) เป็นสัญญาณควบคุมลำอิเล็คตรอนในขบวนการดังกล่าว และในทางเดียวกันลำอิเล็คตรอน จะต้องย้อนกลับไปที่ตำแหน่งมุมซ้ายบนสุด อีกครั้งเมื่อมันมาถึงตำแหน่งมุมล่างขวาสุดแล้ว ซึ่งจะเรียกขบวนการนี้ว่า การย้อนกลับทางแนวตั้ง (Vertical retrace) ซึ่งจะใช้สัญญาณซิ้งค์ทางแนวตั้ง (Vertical Sync Signal) เป็นสัญญาณควบคุมลำอิเล็คตรอนในขบวนการดังกล่าว

ซึ่งในโหมดอินเทอร์แลซ การย้อนกลับทางแนวตั้งจะเกิดขึ้นมากกว่า ในโหมดนอนอินเทอร์แลซ 2 เท่า โดยจะสแกนเส้นคี่ถึงตำแหน่งกึ่งกลางล่างสุดแล้ว จะย้อนขึ้นมาที่ตำแหน่งกึ่งกลางด้านบนสุด เพื่อสแกนเส้นคู่ต่อไปดังรูปที่ 6.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



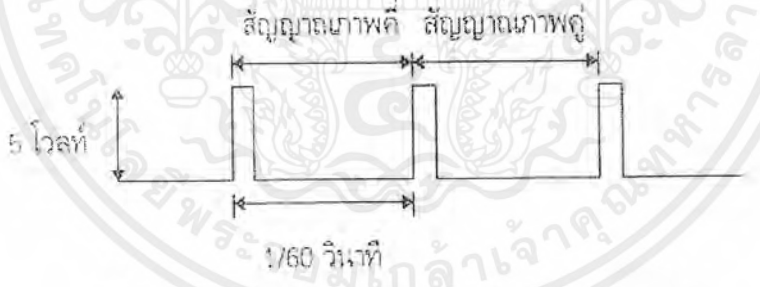
รูปที่ 6.2 ก โหมด อินเทอร์แลซ



รูปที่ 6.2 ข โหมด นอนอินเทอร์แลซ

6.3 วิธีการนำ สัญญาณซิงค์ทางแนวตั้ง มาใช้ควบคุมการมองเห็นภาพ 3 มิติ

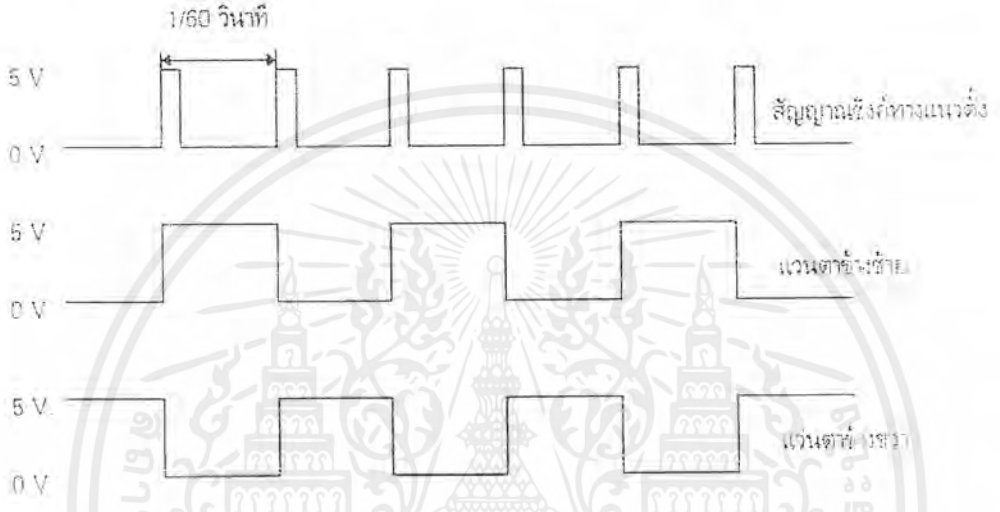
สัญญาณซิงค์ทางแนวตั้งของการ์ดแสดงผลที่โหมดอินเทอร์แลซ (Interlaced Mode) ซึ่งตั้งค่าความละเอียดของหน้าจอไว้ที่ 1024 x 768 จุด จะมีความถี่ประมาณ 60 เฮิร์ตซ์ และมีระดับแรงดันถึงยอด 5 โวลต์ แสดงดังรูปที่ 6.3



รูปที่ 6.3 ภาพสัญญาณซิงค์ทางแนวตั้งของการ์ดแสดงผล

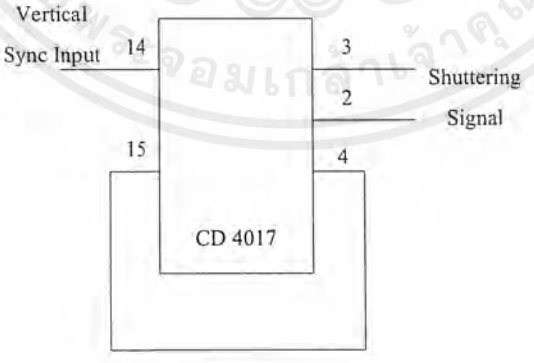
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากในโหมดอินเทอร์แลซ จะแสดงภาพฟิลด์คี่ สลับกับฟิลด์คู่ ไปเรื่อยๆซึ่งเราได้ทำการแมทช์ให้ภาพจากมุมมองจากตาซ้ายอยู่ฟิลด์คี่ และภาพจากมุมมองจากตาขวาอยู่ฟิลด์คู่ซึ่งได้กล่าวรายละเอียดแล้วในบทที่ 3 ดังนั้นถ้าเราทำการหารความถี่ของสัญญาณซิงค์ทางแนวตั้ง ซึ่งควบคุมการสแกนภาพทั้งฟิลด์คี่ และฟิลด์คู่ ลง 2 เท่า เราก็จะได้สัญญาณควบคุมการมองภาพทั้ง 2 ภาพได้อย่างอิสระกัน ดังรูปที่ 6.4



รูปที่ 6.4 สัญญาณที่ออกจากอุปกรณ์ควบคุมการชัตเตอร์ ไปยังแว่น LC

6.4 วงจรในส่วนอุปกรณ์ควบคุมการชัตเตอร์



รูปที่ 6.5 วงจรหารความถี่ลง 2 เท่าโดยใช้ ไอซีเบอร์ CD4017

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5 การทำงานของแว่นตาสามมิติ (3D Glasses) : LC แบบเซลล์เดียว

LC ย่อมาจาก Liquid Crystal ซึ่งหลักการทำงานของอุปกรณ์นี้คือ เมื่อมีระดับแรงดันประมาณ 5-15 โวลท์ป้อนเข้ามาที่เซลล์ LC จะทำให้เกิดสนามไฟฟ้าขึ้นที่เซลล์ ของ LC ซึ่งส่งผลให้โมเลกุลที่อยู่ภายใน LC แบบเซลล์เดียววิ่งมาจับตัวกันเป็นเส้นตรงทำให้ช่วงเวลานั้นไม่สามารถมองเห็นภาพผ่านทางหน้าจอของ LC โดยจากหลักการทำงานของ LC แบบเซลล์เดียวถ้านำอุปกรณ์ LC ดังกล่าวมาใช้ ปิด-เปิดการมองเห็นภาพฟิลด์คี่และฟิลด์คู่ให้อิสระต่อกันได้โดยใช้สัญญาณที่ได้จากรูปที่ 6.4 มาควบคุมการปิด-เปิด LC ชัตเตอร์ซึ่งก็จะสามารถมองเห็นเป็นภาพ 3 มิติได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

ผลการทดลอง

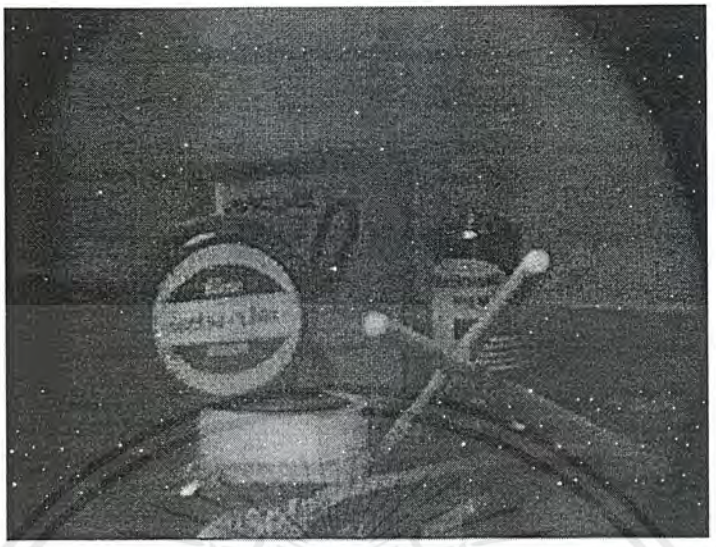
7.1 ภาพที่ได้จากการบันทึกเป็นไฟล์ .AVI

รูปที่ 7.1 ภาพ 3 มิติซึ่งยังไม่ได้แยกฟิลด์

7.2 ภาพที่ได้จากการแยกข้อมูลภาพสองมุมมองออกจากกัน

รูปที่ 7.2 ภาพมุมมองจากตาขวา ซึ่งอยู่บนฟิลด์คู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.3 ภาพมุมมองจากตาซ้าย ซึ่งอยู่บนฟิลด์คี่

7.3 ลำดับภาพที่เกิดขึ้นในการประมวลผลโดยสมอง

ภาพที่มองจากตาขวา

ภาพที่มองจากตาซ้าย

ภาพที่มองจากตาขวา

ภาพที่มองจากตาซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
สรุปผลการทดลอง

จากผลการทดลองจะเห็นได้ว่าองค์ประกอบหลักของโครงงานนี้ซึ่งประกอบด้วย 2 ส่วน คือ ฮาร์ดแวร์และซอฟต์แวร์ จะต้องทำงานสัมพันธ์กัน จึงจะสามารถมองเห็นเป็นภาพสามมิติได้ แต่ทั้งนี้ทั้งนั้นโครงงานนี้ยังมีข้อจำกัดอยู่บางประการเช่น

8.1 ความละเอียดของสีในการแสดงผล สำหรับความละเอียดของสีสูงสุดในการแสดงผลในโครงงานนี้คือ 65536 สี หรือ High Color โดยที่ภาพที่มีความละเอียดของสีมากกว่านี้ ซึ่งเป็นภาพที่ได้จากการบันทึกคือ 16777216 สี หรือ True Color ทำให้ภาพวีดีโอสามมิติที่นำมาแสดงใหม่นั้น มีความผิดเพี้ยนในเรื่องสี แต่ก็อยู่ในเกณฑ์ที่ยอมรับได้

8.2 อัตราการแสดงผลภาพ ไฟล์ภาพ .AVI ซึ่งเป็นไฟล์ภาพต้นฉบับนั้นสามารถควบคุมอัตราการแสดงผลภาพได้ ซึ่งสามารถแสดงผลภาพได้ช้าหรือเร็วตามต้องการได้ แต่ไฟล์ภาพ .3DX ซึ่งเป็นไฟล์ภาพที่ได้รับการแยกข้อมูลแล้วไม่สามารถควบคุมอัตราการแสดงผลได้เนื่องจากข้อจำกัดในโปรแกรมแสดงผลที่พัฒนาขึ้นมาใช้งาน แต่อย่างไรก็ตามก็ยังสามารถมองเห็นเป็นภาพสามมิติได้เช่นกัน

8.3 ขนาดของข้อมูลภาพ ไฟล์ภาพ .AVI เป็นไฟล์ภาพที่มีขนาดใหญ่เนื่องจากไม่มีการบีบอัดข้อมูล (Compression) นับว่าเป็นข้อจำกัดในเรื่องการบันทึก การส่งผ่านข้อมูล ตลอดจนการแสดงผลมาก ซึ่งสมควรอย่างยิ่งที่จะนำเทคโนโลยีการบีบอัดข้อมูลมาใช้งาน

โดยบทสรุปแล้วนั้น หากโครงงานนี้ได้รับการพัฒนาเพิ่มเติมโดยยึดแนวทางหลัก 3 ประการดังที่ได้กล่าวไว้ข้างต้น ก็จะสามารถพัฒนาเป็นระบบโรงภาพยนต์สามมิติภายในบ้าน (3D Video Home Theater) ได้สมบูรณ์แบบยิ่งขึ้น อันจะทำให้เกิดอรรถรสและความบันเทิงมากขึ้นในการรับชม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการใช้งาน

1) เชื่อมต่อฮาร์ดแวร์เข้ากับการ์ดแสดงผล โดยนำสายสัญญาณที่เชื่อมต่อกับจอมอนิเตอร์ต่อเข้ากับอินพุตของฮาร์ดแวร์ และนำสายสัญญาณแบบ DB-16 เชื่อมต่อเอาท์พุตของฮาร์ดแวร์กับอินพุตของการ์ดแสดงผล
2) ปรับโหมดการแสดงผลไปเป็นแบบ Interlaced Mode ซึ่งปกติจะเป็นโหมดที่มีความละเอียดสูง เช่น 1024*768*24
3) เรียกโปรแกรมแสดงผลภาพวิดีโอสามมิติขึ้นมาใช้งาน โดยเรียกไฟล์ .3DX ซึ่งเป็นไฟล์ที่ได้รับการจัดเรียงข้อมูลใหม่
4) กรณีที่ต้องการสร้างภาพวิดีโอสามมิติ ให้เรียกใช้โปรแกรมแยกข้อมูลภาพวิดีโอสามมิติ โดยอินพุตเป็นไฟล์ .AVI ที่ได้รับการบันทึกจากวิดีโอเทป แล้วบันทึกเป็นไฟล์ .3DX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมแยกข้อมูลภาพสามมิติ / ชุดคำสั่งใน Library

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
#include "testWin2.h"
int width,bytecolor,fname,height,framenum,color ;
void OpenAVIFile(char fname[100])
{
////////* Variable Declaretion *//////////////////

        FILE *f ;

        unsigned char a[20] ;
        char n[20][50] ;
        char strwidth[10],strheight[10],strcolor[10],strframe[10] ;
        int i ;
        long j;
        int check ;
////////////////////////////////////////////////

////// * Initialize Variable ////////////////////*/
        i = 0 ;
        int False = 0 ;
        int True = 1 ;
        check = false ;

////////////////////////////////////////////////
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
f = fopen(fname,"r") ;

while((check ==False)) {

        fseek(f,i*sizeof(char),SEEK_SET);
        fread(a,1,1,f) ;
        i++ ;
        if(a[0] == 0x61) {

                fread(a,3,1,f) ;
                if((a[0] ==0x76) && (a[1] == 0x69) && (a[2] == 0x68)) {
                        fseek(f,20*sizeof(char),SEEK_CUR) ;
                ////////// Show the number of frame //////////////

                        fread(a,4,1,f) ;
                        itoa(a[0],n[0],16) ;
                        itoa(a[1],n[1],16) ;
                        itoa(a[2],n[2],16) ;
                        itoa(a[3],n[3],16) ;
                        strcat(n[1],n[0]) ;
                        strcat(n[2],n[1]) ;
                        strcat(n[3],n[2]) ;
                        j = strtol(n[3],NULL,16) ;
                        itoa(j,strframe,10) ;
                        framenum = atol(strframe) ;
                //      MessageBox(NULL,strframe,NULL,MB_OK) ;
                        check = true ;
                }
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }

        fclose(f) ;
                        ////////////////////////////////////////////////

//      f = fopen("c:\\Project\\conv4\\tech2.avi","r") ;
        f = fopen(fname,"r") ;
        i = 0 ;
        check = False ;
        while((check == False)) {

                fseek(f,i*sizeof(char),SEEK_SET);
                fread(a,1,1,f) ;
                i++ ;
                if(a[0] == 0x73) {

                        fread(a,3,1,f) ;
                        if((a[0] ==0x74) && (a[1] == 0x72) && (a[2] == 0x66)) {
                                fseek(f,8*sizeof(char),SEEK_CUR) ;
                ////////// Show the width of frame //////////////

                                fread(a,4,1,f) ;
                                itoa(a[0],n[0],16) ;
                                itoa(a[1],n[1],16) ;
                                itoa(a[2],n[2],16) ;
                                itoa(a[3],n[3],16) ;
                                strcat(n[1],n[0]) ;
                                strcat(n[2],n[1]) ;
                                strcat(n[3],n[2]) ;
                                j = strtol(n[3],NULL,16) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
itoa(j,strwidth,10) ;
width = atol(strwidth) ;
//      MessageBox(NULL,strwidth,NULL,MB_OK) ;

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

////////// Show the height of frame ////////////

fread(a,4,1,f) ;
itoa(a[0],n[0],16) ;
itoa(a[1],n[1],16) ;
itoa(a[2],n[2],16) ;
itoa(a[3],n[3],16) ;
strcat(n[1],n[0]) ;
strcat(n[2],n[1]) ;
strcat(n[3],n[2]) ;
j = strtol(n[3],NULL,16) ;
itoa(j,strheight,10) ;
height = atol(strheight) ;
//      MessageBox(NULL,strheight,NULL,MB_OK) ;

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

////////// Show the color of frame ////////////

fread(a,4,1,f) ;
itoa(a[2],n[2],16) ;
itoa(a[3],n[3],16) ;
strcat(n[3],n[2]) ;
j = strtol(n[3],NULL,16) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    itoa(j,strcolor,10) ;
                    color = atol(strcolor) ;
                    bytecolor = int(color/8) ;
                //  MessageBox(NULL,strcolor,NULL,MB_OK) ;

            ////////////////////////////////////////////////
                    check = True ;
                }
            }
        }
        fclose(f) ;

}

void Encode3DAVI(char fname[100],char savename[100])
{
        // TODO: Add your control notification handler code here

        FILE *f,*newfile ;

        int widthx[2],heightx[2],framex[2],colorx[2] ;
        unsigned char a[10] ;
        unsigned char *EvenData[1200],*OddData[1200] ;
        unsigned char *Data00db ;
        unsigned char *buffer1 ;
        unsigned char *buffer;
        int i,j,yy,y,ii,jj,kk ;
        int temp ;
        unsigned char *Blackdata ;
        int check ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int False = 0 ;
int True = 1 ;
i = 0 ;
jj = 1 ;
kk = 1 ;
temp = width*bytecolor ;
check = False ;

f = fopen(fname,"r") ;
//      newfile = fopen("c:\\ProjectCDX\\CDXtech1-3.3dx","w") ;
newfile = fopen(savename,"w") ;
widthx[0] = width ;
heightx[0] = height ;
framex[0] = framenum ;
colorx[0] = bytecolor ;

fwrite(widthx,2,1,newfile) ;
fwrite(heightx,2,1,newfile) ;
fwrite(framex,2,1,newfile) ;
fwrite(colorx,2,1,newfile) ;

Data00db = (unsigned char *)malloc(8) ;
buffer = (unsigned char *)malloc(temp) ;
buffer1 = (unsigned char *)malloc(8) ;
Blackdata = (unsigned char *)malloc(temp) ;
for(j=0;j<=temp-1;j++) {
        Blackdata[j] = 0x00 ;
}
for(j=1;j<=1000;j++) {
        EvenData[j] =(unsigned char *)malloc(temp) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
for(j=1;j<=1000;j++) {
        OddData[j] =(unsigned char *)malloc(temp) ;
}

while((check == false)) {

        fseek(f,i*sizeof(char),SEEK_SET) ;
        fread(a,1,1,f) ;
        i++ ;
        if(a[0] == 0x6D) {
                fread(a,3,1,f) ;
                if((a[0] == 0x6F) && (a[1] == 0x76) && (a[2] == 0x69)) {
                        fseek(f,8*sizeof(char),SEEK_CUR) ;
                        for(y=1;y<=framenum;y++) {
                                jj = 1 ;
                                kk = 1 ;

                                for(ii=height;ii>=1;ii--) {
                                        fread(buffer,temp,1,f) ;
                                        if(ii % 2 ==0) {
                                                memcpy(EvenData[jj],buffer,temp) ;
                                                jj++ ;
                                        }
                                        else {
                                                memcpy(OddData[kk],buffer,temp) ;
                                                kk++ ;
                                        }
                                //      fseek(f,2,SEEK_CUR) ;
                                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    check = True ;
                    fread(buffer1,8,1,f) ;

                    if(y==1) {
                            memcpy(Data00db,buffer1,8) ;
                    }

                    for(yy=1;yy<=jj-1;yy++) {

                            fwrite(EvenData[yy],temp,1,newfile) ;
                    }

                    for(yy=1;yy<=kk-1;yy++) {

                            fwrite(OddData[yy],temp,1,newfile) ;
                    }

                }
            }
        }
}
fclose(f) ;
fclose(newfile) ;

free(Data00db) ;
free(Blackdata) ;
free(buffer) ;
free(buffer1) ;
for(j=1;j<=1000;j++) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            free(EvenData[j]) ;
        }
        for(j=1;j<=1000;j++) {
            free(OddData[j]) ;
        }
}
```

โปรแกรมแยกข้อมูลภาพสามมิติ / โปรแกรมหลัก

```
#include <windows.h>
#include "resource.h"
#include <stdio.h>
#include <stdlib.h>

#include "testWin2.h"

LRESULT CALLBACK WindowFunc(HWND,UINT,WPARAM,LPARAM) ;
char szWinName[] = "MyWin" ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char fname[100],savename[100] ;

LPSTR GetFileName(HWND hwnd,char *szFile,int StringSize )
{

        OPENFILENAME OpenFileName;
        char szDirName[256] ;
        char szFileTitle[256] ;

        szFile[0] = 0 ;
        memset(&OpenFileName,0,sizeof(OPENFILENAME)) ;
        strcpy(szDirName,"c:\\") ;
// Fill in the OPENFILENAME structure to support a template and hook.
   OpenFileName.lStructSize      = sizeof(OPENFILENAME);
   OpenFileName.hwndOwner        = hwnd;
   OpenFileName.lpstrFilter      = "AVI Files (*.avi)\0*.avi\0" ;
   OpenFileName.nFilterIndex     = 1;
   OpenFileName.lpstrFile        = szFile;
   OpenFileName.nMaxFile         = StringSize;
   OpenFileName.lpstrFileTitle   = szFileTitle;
   OpenFileName.nMaxFileTitle    = sizeof(szFileTitle);
   OpenFileName.lpstrInitialDir  = szDirName ;
   OpenFileName.Flags            = OFN_FILEMUSTEXIST;
        GetOpenFileName(&OpenFileName) ;
        return szFile ;
}
BOOL LoadFile(HWND hwnd)
{

        memset(&fname,0,100) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        GetFileName(hwnd,fname,100) ;

        return TRUE ;

}
void GetSaveName(HWND hwnd)
{
        OPENFILENAME OpenFileName;
        char szDirName[256] ;
        char szFile[256],szFileTitle[256] ;

        memset(&OpenFileName,0,sizeof(OPENFILENAME)) ;
        szFile[0] = '\0' ;
        strcpy(szDirName,"c:\\") ;

        OpenFileName.lStructSize      = sizeof(OPENFILENAME);
  OpenFileName.hwndOwner       = hwnd;
  OpenFileName.lpstrFilter     = "3D Video Files (*.3dx)\0*.3dx\0" ;
  OpenFileName.nFilterIndex    = 1;
  OpenFileName.lpstrFile       = szFile;
  OpenFileName.nMaxFile        = sizeof(szFile) ;
  OpenFileName.lpstrFileTitle  = szFileTitle;
  OpenFileName.nMaxFileTitle   = sizeof(szFileTitle);
  OpenFileName.lpstrInitialDir = szDirName ;
  OpenFileName.Flags           = OFN_OVERWRITEPROMPT;

        GetSaveFileName(&OpenFileName) ;
        strcpy(savename,OpenFileName.lpstrFile) ;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LRESULT CALLBACK WindowFunc(HWND hwnd, UINT message,
                                        WPARAM wParam,LPARAM
lParam)
{

    switch(message) {
    case WM_COMMAND :
        switch(LOWORD(wParam)) {
        case ID_FILE_OPENFILE :

            LoadFile(hwnd) ;
            OpenAVIFile(fname) ;

            break ;

        case ID_ENCODE_ENCODENOW :
            GetSaveName(hwnd) ;
            Encode3DAVI(fname,savename) ;

            break ;
        }
        break ;
        case WM_DESTROY :
            PostQuitMessage(0) ;
            break ;
        default :

            return DefWindowProc(hwnd,message,wParam,lParam) ;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        return 0 ;
}

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInst,
                                    LPSTR lpCmdLine, int nCmdShow)
{

////////////// Window and Menu Initial //////////////

        HWND hwnd ;
        MSG msg ;
        WNDCLASS wcl ;

        wcl.hInstance = hInstance ;
        wcl.lpszClassName = szWinName ;
        wcl.lpfnWndProc = WindowFunc ;
        wcl.style = 0 ;

        wcl.hIcon = LoadIcon(NULL,IDI_APPLICATION) ;
        wcl.hCursor = LoadCursor(NULL,IDC_ARROW) ;
        wcl.lpszMenuName = "MENU" ;

        wcl.cbClsExtra = 0 ;
        wcl.cbWndExtra = 0 ;

        wcl.hbrBackground = (HBRUSH) GetStockObject(WHITE_BRUSH) ;

        if(!RegisterClass (&wcl)) return 0 ;

        hwnd = CreateWindow(
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                szWinName,
                "3D AVI Encoder " ,
                WS_OVERLAPPEDWINDOW,
                CW_USEDEFAULT,
                CW_USEDEFAULT,
                CW_USEDEFAULT,
                CW_USEDEFAULT,
                HWND_DESKTOP ,
                NULL,
                hInstance ,
                NULL
        ) ;

    ShowWindow(hwnd,nCmdShow) ;
    UpdateWindow(hwnd) ;

    while(GetMessage(&msg,NULL,0,0)) {
            TranslateMessage(&msg) ;
            DispatchMessage(&msg) ;
    }
    return msg.wParam ;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมแสดงข้อมูลภาพสามมิติ

```
// ------------------------------------------------------------------
// testCDX8.cpp : Defines the entry point for the application.
// ------------------------------------------------------------------


// ------------------------------------------------------------------
// Generated by the CDX Application Wizard
// By: Michael Rich [istan@alltel.net]
//
// With valuable help from:
//  Ioannis Karagiorgos, for the debugging help
//  Bil Simser, for the good ideas and constructive criticism
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// ------------------------------------------------------------------

#define WIN32_LEAN_AND_MEAN

#include <windows.h>
#define CDXINCLUDEALL    // this define includes all headers, otherwise include one by one
#include <cdx.h>
#include <iostream.h>
#include <fstream.h>
#include <commdlg.h>
#include <string.h>
#include <direct.h>
#include "resource.h"

// ------------------------------------------------------------------
// Global Variables
// ------------------------------------------------------------------
//////////////////////////////////////////////////

ifstream fp ;
int rowdata,xxx ;
int width,height,framenum,color ;
unsigned char *EvenData[500][500],*OddData[500][500] ;
char s[100] ;
//////////////////////////////////////////////////
char     szAppName[]   = "testCDX8";
char     szClassName[]  = "TestCDX8WndClass";

HINSTANCE  g_hInst;                    // instance handle
HWND     g_hWnd;                       // window handle
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BOOL      g_bFullScreen   = FALSE;     // running fullscreen?

// ---------------------------------------------------------------
// CDX Objects
// ---------------------------------------------------------------
CDXScreen   * Screen    = 0;          // The screen object, every program must have one
CDXSurface      * Splash    = 0;

// ---------------------------------------------------------------
// Forward declarations
// ---------------------------------------------------------------
void AdjustWinStyle();



// ---------------------------------------------------------------
// cdx_Init - handles initialization of the CDX objects
// ---------------------------------------------------------------
BOOL cdx_Init()
{
        // Create the CDXSreen object
        Screen = new CDXScreen();
        if (Screen==NULL)
    CDXError( NULL , "Could not create CDXScreen object" );


        // start app in a window
        if (!Screen->CreateWindowed(g_hWnd, 640, 480))
    CDXError( Screen , "Could not set windowed mode 640x480" );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        // TODO: Initialize your own CDX objects here

        Splash = new CDXSurface() ;
        Splash->Create(Screen,640,480) ;
        return TRUE;
}


// ------------------------------------------------------------------
// cdx_DeInit - handles cleanup of CDX objects
// ------------------------------------------------------------------

void cdx_DeInit( void )
{
// TODO: Destroy your CDX objects here

        SAFEDELETE( Splash );
        SAFEDELETE( Screen );
}

LPSTR GetFileName(HWND hwnd,char *szFile,int StringSize )
{

        OPENFILENAME OpenFileName;
        char szDirName[256] ;
        char szFileTitle[256] ;

        szFile[0] = 0 ;
        memset(&OpenFileName,0,sizeof(OPENFILENAME)) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        strcpy(szDirName,"c:\\") ;
// Fill in the OPENFILENAME structure to support a template and hook.
        OpenFileName.lStructSize      = sizeof(OPENFILENAME);
   OpenFileName.hwndOwner        = hwnd;
   OpenFileName.lpstrFilter      = "3D Video Files (*.3dx)\0*.3dx\0" ;
   OpenFileName.nFilterIndex     = 1;
   OpenFileName.lpstrFile        = szFile;
   OpenFileName.nMaxFile         = StringSize;
   OpenFileName.lpstrFileTitle   = szFileTitle;
   OpenFileName.nMaxFileTitle    = sizeof(szFileTitle);
   OpenFileName.lpstrInitialDir  = szDirName ;
   OpenFileName.Flags            = OFN_FILEMUSTEXIST;
        GetOpenFileName(&OpenFileName) ;
        return szFile ;
}

void openfileinfo()
{
        unsigned char data[10] ;
        char strdata[10][10] ;
        char strj[10][10] ;
        int j ;

        fp.read(data,8) ;

        itoa(data[0],strdata[0],16) ;
        itoa(data[1],strdata[1],16) ;
        itoa(data[2],strdata[2],16) ;
        itoa(data[3],strdata[3],16) ;
        itoa(data[4],strdata[4],16) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        itoa(data[5],strdata[5],16) ;
        itoa(data[6],strdata[6],16) ;
        itoa(data[7],strdata[7],16) ;

        strcat(strdata[1],strdata[0]) ;
        strcat(strdata[3],strdata[2]) ;
        strcat(strdata[5],strdata[4]) ;
        strcat(strdata[7],strdata[6]) ;

        j = strtol(strdata[1],NULL,16) ;
        width = j ;
        itoa(j,strj[1],10) ;

        j = strtol(strdata[3],NULL,16) ;
        height = j ;
        itoa(j,strj[2],10) ;

        j = strtol(strdata[5],NULL,16) ;
        framenum = j ;
        itoa(j,strj[3],10) ;

        j = strtol(strdata[7],NULL,16) ;
        color = j ;
        itoa(j,strj[4],10) ;

}
void OpenAVI()
{
        int i,j ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//        fp.open("c:\\ProjectCDX\\CDXtech1-1.3dx",ios::in,0) ;
          openfileinfo() ;
          rowdata = width*color ;
          for(j=1;j<=framenum;j++) {
                    for(i=1;i<=int(height/2);i++) {
                              EvenData[j][i] = (unsigned char *)malloc(rowdata) ;
                              fp.read(EvenData[j][i],rowdata) ;
                    }
                    for(i=1;i<=int(height/2);i++) {
                              OddData[j][i] = (unsigned char *)malloc(rowdata) ;
                              fp.read(OddData[j][i],rowdata) ;
                    }
          }
}
BOOL LoadFile(HWND hwnd )
{
          memset(&s,0,100) ;
          GetFileName(hwnd,s,100) ;
          fp.open(s,ios::in,0) ;
          OpenAVI() ;
          return TRUE ;
}
void PlotEvenPel(int frame)
{
          unsigned long PelData[2000] ;
          int i,j,k,ii,jj,kk ;

//////////// Get Pixel Data ////////////////////

          Splash->DrawFast(0, 0, Screen->GetFront());
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        jj = 240;

        for(i=1;i<=int(height/2);i++) {
                j = 0 ;
                k = 1 ;
                        for(ii=1;ii<=width;ii++) {
                                unsigned char aaa = EvenData[frame][i][j] ;
                                unsigned char bbb = EvenData[frame][i][j+1] ;
                                unsigned char ccc = EvenData[frame][i][j+2] ;
                                PelData[k] = RGB(aaa,bbb,ccc) ;
                                j+=3  ;
                                k++ ;
                        }

                        Splash->Lock() ;

                        for(kk=1;kk<=width;kk++) {
                                Splash->PutPixel(kk,jj,PelData[kk]) ;
                        }
                        jj-=1;
                        Splash->UnLock() ;

        }
}

void PlotOddPel(int frame)
{
        unsigned long PelData[2000] ;
        int i,j,k,ii,jj,kk ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
////////// Get Pixel Data ////////////////////

        Splash->DrawFast(0, 0, Screen->GetFront());
        jj = 480;

        for(i=1;i<=int(height/2);i++) {
                j = 0 ;
                k = 1 ;
                        for(ii=1;ii<=width;ii++) {
                                int aaa = OddData[frame][i][j] ;
                                int bbb = OddData[frame][i][j+1] ;
                                int ccc = OddData[frame][i][j+2] ;
                                PelData[k] = RGB(aaa,bbb,ccc) ;
                                j+=3  ;
                                k++ ;
                        }

                        Splash->Lock() ;

                        for(kk=1;kk<=width;kk++) {
                                Splash->PutPixel(kk,jj,PelData[kk]) ;
                        }
                        jj-=1;
                        Splash->UnLock() ;

        }
}

// ------------------------------------------------------------------
// cdx_DoFrame - performs drawing of the current frame
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// ----------------------------------------------------------------
void cdx_DoFrame()
{

        int j ;

// TODO: Add code to draw your objects during each frame

        for(j=1;j<=framenum;j++) {
                PlotEvenPel(j) ;
                PlotOddPel(j) ;

        }

}

// ----------------------------------------------------------------
// AdjustWinStyle - adjusts the window style according to the mode
// ----------------------------------------------------------------
static void AdjustWinStyle()
{
        if (g_bFullScreen)  // In fullscreen mode the window must cover the whole desktop
        {
                DWORD      dwStyle;

                // Change window attributes

                dwStyle = WS_POPUP | WS_VISIBLE;

                SetWindowLong(g_hWnd, GWL_STYLE, dwStyle);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SetWindowPos(g_hWnd, HWND_TOPMOST, 0, 0, GetSystemMetrics
(SM_CXFULLSCREEN),
                        GetSystemMetrics(SM_CYFULLSCREEN), SWP_NOACTIVATE |
                        SWP_NOZORDER);
        }
        else
        {
                RECT      rect = {0, 0, 640, 480};
                DWORD     dwStyle;

                // Change window attributes

                dwStyle = GetWindowStyle(g_hWnd);
                dwStyle &= ~WS_POPUP;
                dwStyle |= WS_OVERLAPPED | WS_CAPTION | WS_MINIMIZEBOX |
WS_SYSMENU;

                SetWindowLong(g_hWnd, GWL_STYLE, dwStyle);

                // Resize the window so that the client area is 640x480

                AdjustWindowRectEx(&rect, GetWindowStyle(g_hWnd), GetMenu(g_hWnd)
!= NULL,
                        GetWindowExStyle(g_hWnd));

                // Just in case the window was moved off the visible area of the
                // screen.

                SetWindowPos(g_hWnd, NULL, 0, 0, rect.right-rect.left,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                rect.bottom-rect.top, SWP_NOMOVE | SWP_NOZORDER |
                SWP_NOACTIVATE);

            SetWindowPos(g_hWnd, HWND_NOTOPMOST, 0, 0, 0, 0,
                SWP_NOSIZE | SWP_NOMOVE | SWP_NOACTIVATE);
        }
}


// ---------------------------------------------------------------
// WinProc - handles application messages
// ---------------------------------------------------------------
static long PASCAL WinProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM
lParam)
{

    switch(message)
    {
    case WM_COMMAND :
        switch (LOWORD(wParam)) {
        case ID_FILE_OPENUP :
            LoadFile(hWnd) ;
            break ;
        case ID_FILE_SHOWNOW :
            if(1) {
                g_bFullScreen = !g_bFullScreen;
                Splash->Fill(0) ;
                if( Screen->CreateFullScreen(g_hWnd, 640, 480, 24) ==
FALSE )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    CDXError( Screen , "Could not set 640x480x24 video mode" );
                }
                break ;
            }

            case WM_KEYDOWN:  switch(wParam)
                        {
                            case VK_ESCAPE:       // if ESC key is hit, quit program
                                SendMessage(hWnd, WM_CLOSE, 0, 0);
                                break;
                        }
                        return  0;

            case WM_SYSKEYUP:  if (wParam == VK_RETURN)    // Alt+Enter switches
modes
                        {
                            g_bFullScreen = !g_bFullScreen;
                            if( g_bFullScreen )
                            {
                                AdjustWinStyle();
                                // we need to destroy and recreate the
supporting
                // CDX objects that can't survive intact when
                // we switch back to fullscreen mode
                {
                                    // TODO: add your own CDX
objects that would
                    // have to be destroyed on a mode switch
                    // (they will be recreated below)
                                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        // enable triple buffering
                                        Screen->SetTripleBuffering(TRUE);
                    if( Screen->CheckIfVideoModeExists( 640 , 480 , 24 ) == TRUE )
                    {
                      if( Screen->CreateFullScreen(g_hWnd, 640, 480, 24)  == FALSE )
                        CDXError( Screen , "Could not set 640x480x24 video mode" );
                    }
                    else
                    {
                      if( Screen->CreateFullScreen(g_hWnd, 640, 480, 32)  == FALSE )
                        CDXError( Screen , "Could not set 640x480x32 video mode" );
                    }

                                        {
                                            // TODO: recreate your CDX objects
that
                      // were destroyed above
                                        }
                                    }
                                    else
                                    {
                                        Screen->GetDD()->RestoreDisplayMode();
                                        AdjustWinStyle();
                                        // we need to destroy and recreate the
supporting
                    // CDX objects that can't survive intact when
                    // we switch back to windowed mode
                                        {
                                            // TODO: add your own CDX
objects that would
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    // have to be destroyed on a mode switch
                    // (they will be recreated below)
                                        }
                                        Screen->CreateWindowed(g_hWnd, 640, 480);
                                        {
                                                // TODO: recreate your CDX objects that
                    // were destroyed above
                                        }

                                }
                                return 0;
                        }
                        return 1;
                        case WM_CLOSE:          cdx_DeInit();

                        case WM_DESTROY:    PostQuitMessage(0);
                                return 0;

                        default:            return DefWindowProc(hWnd, message, wParam, lParam);
                }
}

// ------------------------------------------------------------------
// InitApp - Create the window and the CDX objects
// ------------------------------------------------------------------
static BOOL InitApp(int nCmdShow)
{
        WNDCLASS WndClass;

        WndClass.style = CS_HREDRAW | CS_VREDRAW;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        WndClass.lpfnWndProc = WinProc;
        WndClass.cbClsExtra = 0;
        WndClass.cbWndExtra = 0;
        WndClass.hInstance = g_hInst;
        WndClass.hIcon = LoadIcon(g_hInst, "APPICON");
        WndClass.hCursor = LoadCursor(0, IDC_ARROW);
        WndClass.hbrBackground = (HBRUSH)GetStockObject(WHITE_BRUSH);
        WndClass.lpszMenuName = (LPCSTR)IDR_MENU1 ;
        WndClass.lpszClassName = szClassName;
        RegisterClass(&WndClass);

        g_hWnd = CreateWindowEx(
                0,
                szClassName,
                szAppName,
                WS_OVERLAPPED | WS_CAPTION | WS_MINIMIZEBOX |
WS_SYSMENU,
                0,
                0,
                320,
                240,
                NULL,
                NULL,
                g_hInst,
                NULL);

    // when hWnd = -1 there was an error creating the main window
    // CDXError needs a CDXScreen object, if there is none at this early
    // program stage, pass it NULL
        if( !g_hWnd )
```

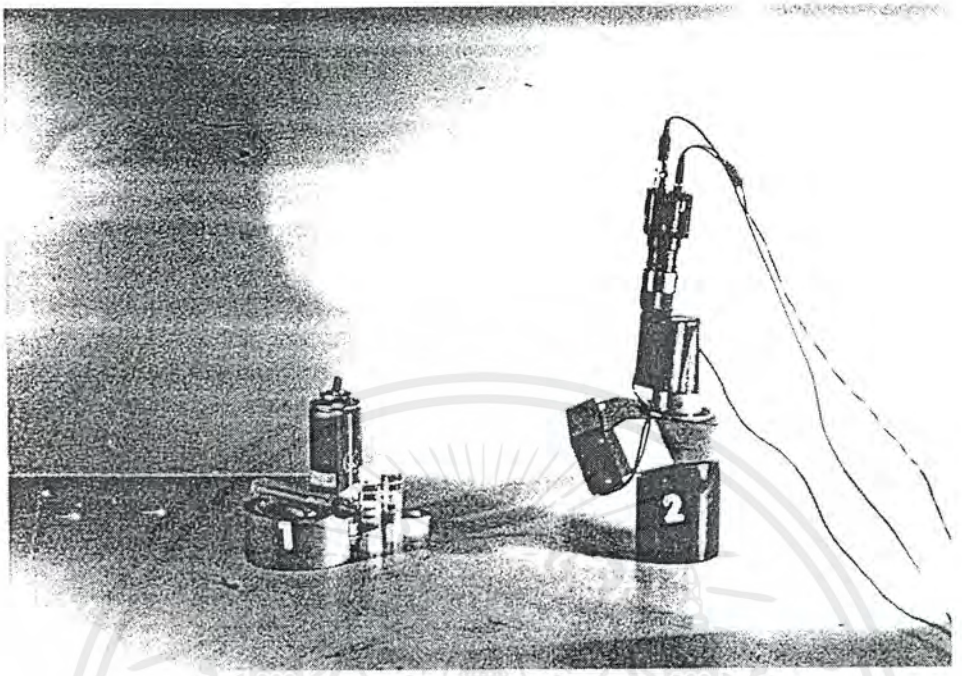
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    CDXError( NULL , "could not create the main window" );


        AdjustWinStyle();

        ShowWindow(g_hWnd, nCmdShow);
        UpdateWindow(g_hWnd);

        return TRUE;
}


// ---------------------------------------------------------------
// WinMain - inital function called by windows
// ---------------------------------------------------------------
int PASCAL WinMain(HINSTANCE hInstance, HINSTANCE hPrevInst,
                                    LPSTR lpCmdLine, int nCmdShow)
{

        MSG msg ;
        int i,j ;

  // save the app instance
        g_hInst = hInstance;

        if(!InitApp(nCmdShow))
     CDXError( NULL , "could not initialize application" );

        if(!cdx_Init())
        {
```

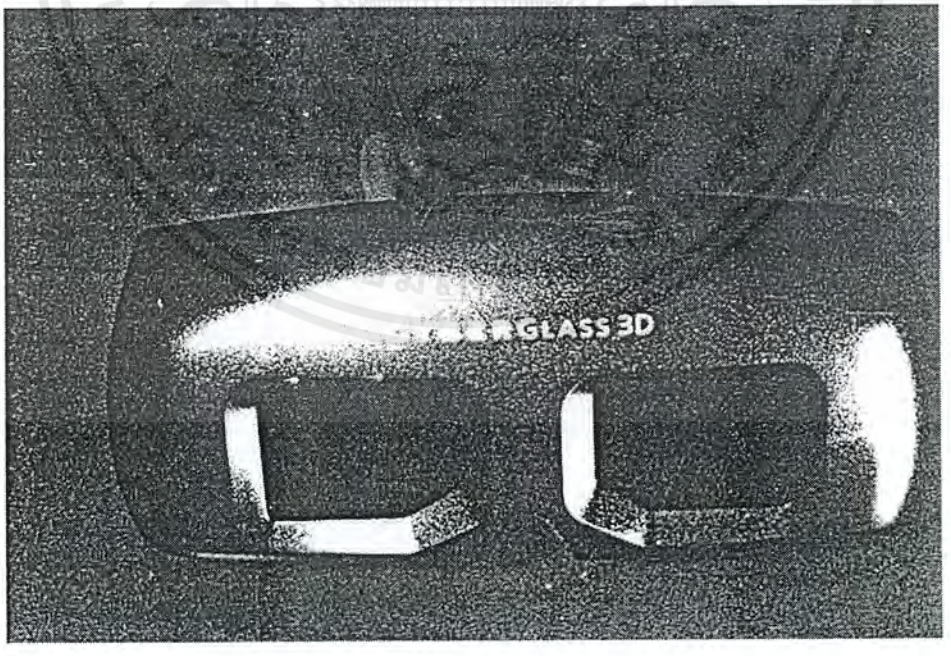
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            PostQuitMessage(0);
            return FALSE;
    }

    while(1)
    {

            if(PeekMessage(&msg, NULL, 0, 0, PM_NOREMOVE))
            {
                    if(!GetMessage(&msg, NULL, 0, 0 )) return msg.wParam;
                    TranslateMessage(&msg);
                    DispatchMessage(&msg);
            }
            else
            {
                    if( g_bFullScreen )
                    cdx_DoFrame();

            }
    }

    for(j=1;j<=framenum;j++) {
            for(i=1;i<=int(height/2);i++) {
                    free(EvenData[j][i]) ;
                    free(OddData[j][i]) ;
            }
    }
    fp.close() ;
    return 0 ;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



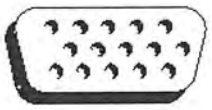
: รูปการถ่ายภาพวิดีโอสามมิติโดยอุปกรณ์ทั้งหมดซึ่งเป็นส่วนเริ่มต้นของโครงงานนี้ ในการนำภาพซึ่งอัดลงม้วนวิดีโอมาทำในโครงงาน



: รูปของแว่นตาสามมิติ (Cyber Glass 3D) ที่เป็นอุปกรณ์อีกส่วนหนึ่งในการมองภาพวิดีโอสามมิติของโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VGA Connector (15 pin): Latitude XPi CD



5 4 3 2 1
10 9 8 7 6
15 14 13 12 11

Pin Assignments for VGA Connector (15 pin)

Pin	Signal	I/O	Definition
1	RED	O	Red video
2	GREEN	O	Green video
3	BLUE	O	Blue video
4	MID2	I	Monitor ID bit 2
5	GND	N/A	Signal ground
6	GND	N/A	Red return (ground)
7	GND	N/A	Green return (ground)
8	GND	N/A	Blue return (ground)
9	GND	N/A	key (no pin)
10	GND	N/A	Sync return (ground)
11	MID0	I	Monitor ID bit 0
12	MID1	I	Monitor ID bit 1
13	HSYNC	O	H sync
14	VSYNC	O	V sync
15	NC	N/A	No connection
Shell	N/A	N/A	Chassis ground

Copyright 2000 Dell Computer Corporation.
Online Privacy Practices : Site Terms of Use

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

หนังสือภาษาไทย

Animate Group, ภาพสามมิติมหัศจรรย์ กรุงเทพมหานคร : Animate Printing And Design

หนังสือภาษาอังกฤษ

Joshua Edding, How Virtual Reality Works : Bookshelf Category General Computing

B.K Johnson, Opic and Optical Instrument :Dover Publication, Inc.

Michel Jones & Allen Wyatt, 3D Madness :Sams Publishing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้