โครงการธิ่งประดิษฐ์

รายงาน

ระบบคอมพิวเตอร์เพื่อการจินตทัศน์ข้อมูลแบบสามมิติสเตอริโอสลับเชิงเวลา

โดย

สมชาย ประสิทธิ์จูดระกูล

จห
วห 15
009466

ตุลาคม 2540

จุฬาลงกรณ์มหาวิทยาลัย

ทุนโครงการสิ่งประดิษฐ์

รายงาน

ระบบคอมพิวเตอร์เพื่อการจินตทัศน์ข้อมูลแบบสามมิติสเตอริโอสลับเชิงเวลา

โดย

สมชาย ประสิทธิ์จูตระกูล

ตุลาคม ๒๕๔๐

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

I19949625 23 S.A. 2541

ชื่อโครงการ ระบบคอมพิวเตอร์เพื่อการจินตทัศน์ข้อมูลแบบสามมิติสเตอริโอสลับเชิงเวลา

ชื่อผู้ดำเนินงาน สมชาย ประสิทธิ์จูตระกูล

เดือนและปีที่ทำวิจัยเสร็จ ตุลาคม 2540

บทคัดย่อ

รายงานนี้นำเสนอระบบคอมพิวเตอร์เพื่อการจินตทัศน์ข้อมูลแบบสามมิติสเตอริโอสลับเชิงเวลา ซึ่งอาศัยแนวคิดของการรับรู้ความลึกจากความแตกต่างทางแนวนอนของภาพที่ตามนุษย์เห็น การแสดงภาพสเตอริโอสลับเชิงเวลาบนจอภาพคอมพิวเตอร์ และการทำงานของแว่นตาเปิดปิดแบบผลึกเหลว มาใช้กับระบบการจินตทัศน์ Visualization ToolKit ของบริษัทเจนเนรัล อิเล็กตริก และระบบแบบจำลองสามมิติ 3D-Studio MAX ของบริษัทออโตเดกส์ โดยพัฒนาเป็นส่วนเสริมเพิ่มเข้าไปในซอฟต์แวร์ทั้งสองเพื่อให้มีคำสั่งการทำงานเพิ่มขึ้นในการแสดงภาพสามมิติสเตอริโอ โดยเลือกใช้เทคนิคการแสดงภาพแบบสอดสลับ โดยรวมภาพคู่จากภาพสำหรับตาซ้ายและตาขวาเป็นภาพเดียวแบบสลับเส้น ทั้งนี้เนื่องจากสามารถใช้ได้กับระบบคอมพิวเตอร์ทั่วไป โดยไม่ต้องปรับปรุงซอฟต์แวร์ระบบในส่วนแสดงผลที่มีอยู่เดิม แต่อาศัยการทำงานของวงจรถอดรหัสเพิ่มเติมภายนอกเพื่อควบคุมการเปิดปิดแว่นผลึกเหลวให้ถูกจังหวะ นอกจากนี้ยังได้พัฒนาโปรแกรมเพื่อแสดงภาพยนตร์จากแฟ้มคู่ในรูปแบบ FLI/FLC (ที่ได้มาจาก 3D-Studio MAX) ในลักษณะสามมิติสเตอริโอแบบพลิกหน้า เพื่อทำงานใต้ระบบปฏิบัติการ DOS ทั้งนี้เพื่องานการนำเสนอที่ให้รายละเอียดของภาพที่ดีกว่าแบบสอดสลับ แต่ต้องการระบบคอมพิวเตอร์ที่มีความเร็วและหน่วยแสดงผลที่ดีกว่า

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Project Title	Time-Multiplexed Three-Dimensional Stereoscopic Data Visualization Computer System
Name of the Inverstigator	Somchai Prasitjutrakul
Year	1997

Abstract

This report presents a time-multiplexed three-dimensional stereoscopic data visualization computer system. The system applies concepts of horizonal parallax for human eye depth perception, time-multiplexed stereoscopic computer image display, and liquid crystal shuttered glass for extending the capability of the Visualization Toolkit (VTK) visualization system software from General Electric corporation and the 3D-Studio MAX three-dimensional modelling software from Autodesk Inc. The extension lets both softwares display three-dimensional stereoscopic images using interlace display technique where an image pair for left and right eyes are merged into a single image by alternating each scan line from the image pair. With this technique, it can be used in most of the computer system without any modification to the existing display driver system software by using additional controlling circuit for synchronizing the liquid-crystal-shuttered glass. In addition, we developed a program for simultaneously displaying two FLI/FLC format animation files (generated from the 3D-Studio MAX) using page flipping technique running under DOS in order to achieve higher resolution quality image than the interlace technique, however it requies a faster and better computer processing and display units.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

กิตติกรรมประกาศ

ผู้วิจัยขอขอบคุณ คณะอนุกรรมการสิ่งประดิษฐ์ประเภทโปรแกรมคอมพิวเตอร์ ฝ่ายวิจัย จุฬาลงกรณ์มหาวิทยาลัย ที่ให้ทุนสนับสนุนงานสิ่งประดิษฐ์ ให้สามารถดำเนินการประดิษฐ์ได้อย่างดี และขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ให้ความอิสระในการใช้สถานที่ และอุปกรณ์เสริมบางอย่างในการดำเนินการประดิษฐ์ งานสิ่งประดิษฐ์นี้มิได้เพิ่งเริ่มทำในปีที่ผ่านมาระหว่างการรับทุน แต่เป็นโครงการที่ผู้วิจัยให้ความสนใจเป็นงานอดิเรกมานาน จากที่ได้เคยเล่นเกมสามมิติ และได้ซื้อหาอุปกรณ์ราคาถูกประกอบการเล่นเกมดังกล่าว จึงได้จัดให้มีหัวข้อโครงงานวิศวกรรมและวิทยานิพนธ์ของนิสิตภาควิชาวิศวกรรมคอมพิวเตอร์อย่างต่อเนื่องในห้าปีที่ผ่านมา เทคนิคการประดิษฐ์นั้นได้รับการเปลี่ยนแปลงอยู่เรื่อยๆ ตามเทคโนโลยีของเครื่องคอมพิวเตอร์ที่เปลี่ยนแปลงอย่างรวดเร็ว ผู้วิจัยจึงขอขอบคุณทีมงานที่ได้ร่วมงานกันมาในช่วงห้าปีที่ผ่านมาอย่างต่อเนื่อง ได้แก่ สุรเดช สามะบุตร, จีรวุธ คงอ่อน, วรุฒ ทวีทรัพย์, ทัชพล ไทยทองสุข, ชลธิชา พงศ์สุปราณี, เชษฐ์ วิศวิกรานต์, ศิริพร ผสมทรัพย์, วนัชสวัณณ์ นักร้อง, มงคล ภิญโญสโมสร, ไกรฤกษ์ ต๊ะวิชัย, ไพโรจน์ แสงสุทธิจริตกุล, พิรพัฒน์ ถิ่นสันติสุข, นภัทร จึงพัฒนปรีชา และ รุจิพันธ์ อัสสะรัตน์

สมชาย ประสิทธิ์จูตระกูล

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

บทคัดย่อ ii

Abstract iii

กิตติกรรมประกาศ iv

สารบัญ v

1. บทนำ 1

2. การรับรู้ความลึกและการแสดงภาพสามมิติ 4

2.1 การรับรู้ความลึก 4

2.2 การแสดงภาพสามมิติ 5

2.2.1 โฮโลแกรม 5

2.2.2 การแสดงแบบหลายระนาบ 5

2.2.3 ภาพคู่สเตอริโอ 6

2.3 การแสดงภาพคู่สเตอริโอสลับเชิงเวลา 8

2.4 การสร้างภาพคู่สามมิติ 12

2.4.1 การฉายภาพ 12

2.4.2 การฉายภาพนอกแกน 12

2.4.3 การฉายภาพบนแกน 13

2.4.4 บริเวณของมุมมอง 14

2.5 ปัจจัยที่มีผลกับคุณภาพของภาพ 15

2.5.1 ภาพหลอนและการรบกวนข้ามตา 15

2.5.2 อัตราการกวาดภาพของจอแสดงผล 16

2.6 การปรับขนาดของภาพ 16

3. ระบบจินตทัศน์ข้อมูล 18

3.1 Visualization Toolkit 19

3.2 3D-Studio MAX 23

4. ฮาร์ดแวร์สำหรับการแสดงภาพสามมิติสเตอริโอ สลับเชิงเวลา 28

4.1 แว่นตาผลึกเหลว 28

4.2 วงจรควบคุมแว่นเปิดปิดผลึกเหลว 29

4.2.1 วงจรควบคุมแว่นเปิดปิดผลึกเหลวแบบพลิกหน้า 29

4.2.2 วงจรควบคุมแว่นเปิดปิดผลึกเหลวแบบสอดสลับ 32

5. ส่วนเสริมแสดงภาพสามมิติแบบสอดสลับใน VTK 36

5.1 หน้าที่ของส่วนเสริม 36

5.2 โครงสร้างของส่วนเสริม 37
5.3 การพัฒนาส่วนเสริม 38
6. ส่วนเสริมแสดงภาพสามมิติแบบสอดสลับใน 3D-Studio MAX 44
6.1 ส่วนเสริมเพื่อการดูภาพสามมิติแบบสอดสลับระหว่างการออกแบบ 44
6.2 ส่วนเสริมเพื่อการสร้างแฟ้มภาพยนตร์สามมิติแบบสอดสลับ 48
7. โปรแกรมแสดงภาพยนตร์ FLI/FLC สามมิติ แบบพลิกหน้า 50
7.1 องค์ประกอบของโปรแกรม 50
7.2 ส่วนถอดรหัสภาพคู่ 51
7.3 ส่วนพลิกหน้าแสดงภาพ 53
7.4 การใช้มาตรฐาน VESA 54
8. บทสรุปและข้อเสนอแนะ 56
เอกสารอ้างอิง 60
ภาคผนวก ก การใช้งานส่วนเสริม 61
ภาคผนวก ข แฟ้มข้อมูลใน CD-ROM 67
ภาคผนวก ค การเผยแพร่ผลงาน 70

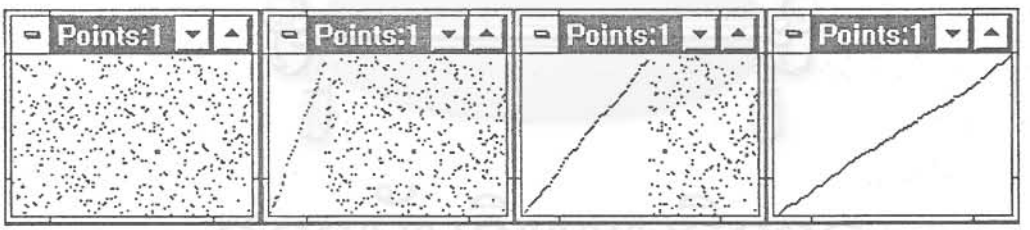
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

เลขหมู่ จส
ค 15
เลขทะเบียน 009466
วัน,เดือน,ปี ๘ ต.ค. 41

1. บทนำ

การแสดงภาพด้วยเครื่องคอมพิวเตอร์ได้กลายเป็นส่วนหนึ่งของการประยุกต์ใช้งานเครื่องคอมพิวเตอร์ในทุกแขนงงาน คอมพิวเตอร์กราฟิกส์ (computer graphics) จึงเป็นศาสตร์ที่ได้รับความสนใจอย่างมากทั้งในอดีต ปัจจุบัน และอนาคต โดยเฉพาะอย่างยิ่งคอมพิวเตอร์กราฟิกส์แบบสามมิติ ทั้งนี้สืบเนื่องมาจากวิวัฒนาการที่เติบโตไปอย่างคล้องจองกันทางด้านการพัฒนาซอฟต์แวร์ และความสามารถของฮาร์ดแวร์ทางด้านการคำนวณและการแสดงผล ทำให้การแสดงภาพสามมิติกระทำได้รวดเร็ว โต้ตอบกับผู้ใช้โดยตรง ในราคาที่ต่ำ การแสดงภาพสามมิติ ไม่ว่าจะเป็นภาพนิ่งหรือภาพเคลื่อนไหว เป็นที่ต้องการอย่างสูง ในการประยุกต์คอมพิวเตอร์ในการแสดงภาพทางการแพทย์ (medical imaging) การใช้คอมพิวเตอร์ช่วยออกแบบ (Computer-Aided Design) การวิจัยในเรื่องของความจริงเสมือน (virtual reality) การจินตทัศน์ข้อมูล (data visualization) และอื่นๆ เป็นต้น

การจินตทัศน์ข้อมูลคือการใช้ภาพนิ่งหรือภาพเคลื่อนไหวเป็นสื่อในการแสดงพฤติกรรมของกลุ่มข้อมูลที่สนใจศึกษา ข้อมูลที่กล่าวถึงนี้โดยทั่วไปจะเป็นกลุ่มข้อมูลขนาดใหญ่ (large data sets) และมักเป็นข้อมูลแบบหลายมิติ (multidimension) กลุ่มข้อมูลนี้อาจเป็นข้อมูลดิบที่สุ่มสำรวจมาได้ (sampling data) ผลลัพธ์ระหว่างการจำลองระบบ (simulation) หรือข้อมูลที่เปลี่ยนแปลงระหว่างการแก้ไขปัญหา เป็นต้น [1] ตัวอย่างเช่น รูปที่ 1-1 แสดงภาพที่แทนรายการข้อมูลตั้งแต่เริ่มจนจบของการเรียงลำดับข้อมูลแบบแทรก (Insertion sort) แต่ละข้อมูลถูกแทนด้วยจุดหนึ่งจุด ความสูงของจุดแปรตามค่าของข้อมูล ข้อมูลในรายการเรียงจากซ้ายไปขวา ดังนั้นถ้าข้อมูลในรายการเรียงลำดับจากซ้ายไปขวาเรียงตามค่าจากน้อยไปมาก จะได้จุดเรียงตามแนวทแยงความลาดเอียงเป็นค่าบวก (ดังแสดงในรูปขวาสุด) [2]



รูปที่ 1-1 ภาพการเปลี่ยนแปลงของข้อมูลระหว่างการเรียงข้อมูลแบบแทรก (Insertion sort)

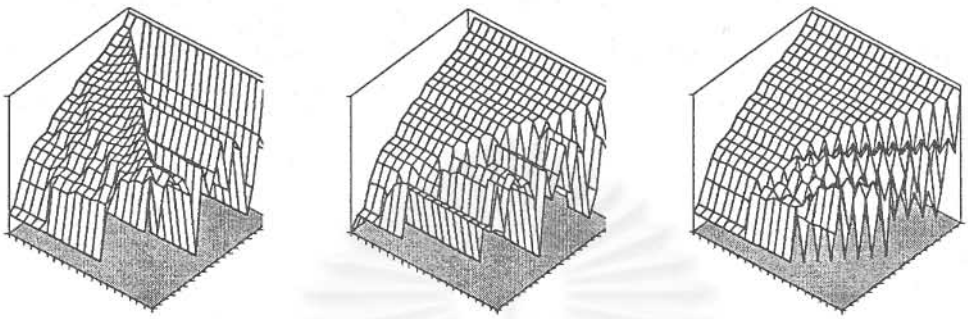
การจินตทัศน์ข้อมูลในรูปที่ 1-1 นั้นจับเฉพาะช่วงของการเปลี่ยนแปลงบางช่วงเท่านั้น ไม่ได้แสดงให้เห็นถึงภาพรวมการเปลี่ยนแปลง โดยทั่วไปผู้ใช้จะต้องสังเกตพฤติกรรมการเปลี่ยนแปลงของ

---

1 P. R. Keller and M. M. Keller, *Visual Cues: Practical Data Visualization*, IEEE Computer Society Press, June 1993.

2 S. Prasitjutrakul and W. Watcharawittayakul, "Algorithm Visualization System: An Overview of Internal Structure", *Proc. of Thrid ASEAN Regional Seminar on Microelectronics and Information Technology*, Bangkok, August 1994.

ข้อมูลระหว่างการเรียงลำดับข้อมูลจริงๆทางหน้าจอคอมพิวเตอร์ โดยจะเห็นเป็นภาพเคลื่อนไหวตลอดการทำงาน หากจะสรุปการทำงานรวมทั้งหมดของของการเรียงลำดับอาจกระทำได้ โดยการเพิ่มแกนเวลาและเก็บรวบรวมข้อมูลเป็นช่วงๆ แล้วนำมาวาดกราฟได้ ดังเช่นตัวอย่างในรูปที่ 1-2 ที่แสดงลักษณะการเปลี่ยนของข้อมูลระหว่างการเรียงลำดับข้อมูลสามวิธีคือแบบแทรก (insertion) แบบเลือก (selection) และแบบฟอง (bubble)



(ก) Insertion Sort (ข) Selection Sort (ค) Bubble Sort

รูปที่ 1-2 การเปรียบเทียบการเปลี่ยนแปลงข้อมูลระหว่างการเรียงลำดับข้อมูลสามวิธี

โครงการสิ่งประดิษฐ์นี้มีแนวคิดในการประยุกต์การรับรู้ความลึกโดยใช้การแสดงภาพแบบสามมิติสเตอริโอสลับเชิงเวลา เพื่อการจินตทัศน์ข้อมูล ทั้งนี้เพื่อทำให้ผู้ที่ต้องการศึกษาพฤติกรรมของกลุ่มข้อมูลที่สนใจ ได้เห็นภาพรวมของกลุ่มข้อมูลได้อย่างมีความหมายและเข้าใจอย่างลึกซึ้ง ลักษณะการนำเสนอจะไม่ใช่จุดมุ่งหมายของโครงการนี้ ทั้งนี้เนื่องจากเทคนิคการนำเสนอเพื่อการจินตทัศน์ที่ได้ความหมายนั้นจะขึ้นกับตัวข้อมูลและที่มาของข้อมูล แต่โครงการนี้จะเน้นการพัฒนาส่วนเสริมการแสดงภาพสามมิติ ให้กับระบบการจินตทัศน์ที่มีอยู่บนเครื่องคอมพิวเตอร์ส่วนบุคคลที่ใช้กันทั่วไป โดยใช้อุปกรณ์เสริมทางด้านการรับรู้ที่ราคาถูก ใช้งานได้อย่างมีประสิทธิภาพ ซึ่งแตกต่างจากการจินตทัศน์โดยทั่วไปที่มักกระทำบนเครื่องคอมพิวเตอร์แบบเวิร์คสเตชัน (Workstation) ที่มีภาคการแสดงที่มีประสิทธิภาพ แสดงสีได้มากมายด้วยความละเอียดสูง แต่ราคาแพง [3] ระบบจินตทัศน์ที่ได้รับการเสริมความสามารถการแสดงภาพสามมิติในโครงการนี้ คือ Visualization ToolKit (VTK) ของบริษัทเจนเนอรัลอิเล็กทริก และ 3D-Studio MAX ของบริษัทออโตเดกส์ เหตุผลที่เลือกการเสริมสามารถซอฟต์แวร์ทั้งสองนี้เนื่องจากเป็นซอฟต์แวร์ที่ได้รับความนิยม ใช้ได้กับเครื่องคอมพิวเตอร์ส่วนบุคคลราคาไม่แพง และการใช้แนวคิดเชิงวัตถุในการออกแบบ

เนื้อหาในส่วนที่เหลือของรายงานนี้พอสรุปได้ดังนี้ บทที่ 2 จะกล่าวถึงการรับรู้ความลึกของตามนุษย์ และวิธีการแสดงภาพสามมิติในปัจจุบัน โดยจะเน้นเทคนิคการแสดงภาพคู่สเตอริโอสลับเชิงเวลาที่ใช้โครงการนี้ บทที่ 3 นำเสนอคุณสมบัติและโครงสร้างของระบบจินตทัศน์ Visualization ToolKit

---

[3] Stereographics, Stereographics Resource Guide, Stereographics Corporation, 1994

(VTK) และ 3D-Studio MAX ที่จะได้รับการเสริมความสามารถในการแสดงภาพสามมิติ บทที่ 4 อธิบายการทำงานของอุปกรณ์สำหรับการแสดงภาพสามมิติสเตอริโอสลับเชิงเวลา อันประกอบด้วยแว่นตาเปิดปิดผลึกเหลว และวงจรควบคุมแว่น บทที่ 5 และ 6 จะนำเสนอหน้าที่ โครงสร้าง และวิธีการพัฒนาส่วนเสริมการแสดงภาพสามมิติ ที่พัฒนาให้กับซอฟต์แวร์ VTK และ 3D-Studio MAX ตามลำดับ บทที่ 7 นำเสนอรายละเอียดของโปรแกรมเพื่อการแสดงภาพยนตร์สามมิติ ที่อาศัยเทคนิคการพลิกหน้าหน่วยความจำแสดงผล และบทที่ 8 สรุปผลที่ได้ นำเสนอข้อจำกัด และข้อเสนอแนะประเด็นการพัฒนาในอนาคต (อนึ่ง โปรแกรมต้นฉบับ โปรแกรมที่ใช้งาน และเอกสารเสริมต่างๆที่ใช้ในโครงการสิ่งประดิษฐ์นี้ ถูกเก็บอยู่ในแผ่น CD-ROM ประกอบรายงานฉบับนี้)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2. การรับรู้ความลึกและการแสดงภาพสามมิติ

ในบทนี้จะได้กล่าวถึงพื้นฐานทางด้านการรับรู้ภาพสามมิติ การแสดงภาพสามมิติ และ การสร้างภาพคู่สเตอริโอ โดยในโครงการนี้จะเลือกใช้เทคนิคการรับรู้ภาพสามมิติจากความแตกต่างของภาพที่ฉายมาสู่ตาทั้งสองทางแนวนอน และใช้เทคนิคการแสดงภาพคู่สเตอริโอของตาทั้งสองนี้บนจอคอมพิวเตอร์แบบสลับเชิงเวลา

2.1 การรับรู้ความลึก [1]

ระบบการมองภาพของมนุษย์นั้นใช้การรับรู้หลายอย่าง เพื่อแยกความลึกของตำแหน่งของวัตถุที่มองเห็น การรับรู้ความลึกดังกล่าวนี้ประกอบด้วยสองหลักการใหญ่ๆคือ ในทางสรีรศาสตร์ (physiological) และในทางจิตศาสตร์ (psychological) ในทางสรีรศาสตร์สามารถจำแนกเป็น 4 กลุ่มย่อย คือ

1. Accommodation คือการเปลี่ยนแปลงระยะโฟกัสของเลนส์นัยน์ตาขณะที่เพ่งไปสู่ส่วนที่สนใจ
2. Convergence คือการเคลื่อนลูกตาเข้าหากัน ขณะซึ่งวัตถุเคลื่อนเข้าหาผู้มอง
3. Binocular disparity คือความแตกต่างของภาพที่ฉายมาสู่ตาข้างซ้าย และตาข้างขวา
4. Motion parallax คือการรับรู้การเปลี่ยนแปลงของระยะทางสัมพัทธ์ของวัตถุ เมื่อวัตถุหรือผู้มองมีการเคลื่อนที่

ส่วนการรับรู้ความลึกในทางจิตศาสตร์นั้นมีผลมาจากสิ่งต่อไปนี้

1. Linear perspective คือคุณสมบัติของการมองที่ทำให้ขนาดของวัตถุที่มองมีขนาดเล็กลง เมื่อห่างจากผู้มองมากขึ้น
2. Shading and shadowing คือคุณสมบัติทางแสงในภาพที่ช่วยให้ผู้มองรับรู้ตำแหน่งทางลึกเทียบกับตำแหน่งของแสง
3. Aerial perspective คือคุณสมบัติของการมองที่ทำให้เห็นวัตถุที่อยู่ไกล ชัดเจนน้อยกว่าที่อยู่ใกล้
4. Interposition (occlusion) คือการบดบังกันของวัตถุที่ทำให้ผู้มองรับรู้ว่าวัตถุที่อยู่ใกล้จะบังวัตถุที่อยู่ไกลกว่า
5. Texture gradient คือความลาดของเนื้อวัตถุที่ช่วยให้ผู้มองรับรู้ความลึกจากความละเอียดของเนื้อวัตถุที่เห็น

---

[1] David F. Mcallister, "3-D Displays", *BYTES*, May 1992, pp.183-188.

หอสมุดกลาง จุฬาลงกรณ์มหาวิทยาลัย

6. Color คือสีของวัตถุซึ่งสามารถช่วยเน้นความลึกได้ โดยวัตถุที่อยู่ใกล้มักมีสีที่สดใสกว่า

โดยทั่วไปแล้วการรับรู้ความลึกที่ได้กล่าวมานี้จะมีคุณสมบัติที่เสริมซึ่งกันและกัน ในบางสถานการณ์คุณสมบัติบางอย่างอาจเด่นกว่าคุณสมบัติอย่างอื่น ตัวอย่างเช่น การบดบังกันของวัตถุจะมีอิทธิพลต่อการรับรู้ความลึกมากกว่า ความแตกต่างของภาพสำหรับตาทั้งสอง เป็นต้น

2.2 *การแสดงภาพสามมิติ* [2]

ด้วยเทคโนโลยีทางการแสดงภาพสามมิติที่ถูกประดิษฐ์คิดค้นขึ้นใหม่มากมาย เป็นผลมาจากประสิทธิภาพของฮาร์ดแวร์ที่เพิ่มสูงขึ้นและราคาลดลง การเปลี่ยนแปลงทางด้านความเร็ว ความละเอียด และราคาของอุปกรณ์ ตลอดจนการพัฒนาเทคโนโลยีใหม่ๆในการแสดงผล อาทิเช่น หน่วยเปิดปิดบังคับแสงทางเดียวแบบผลึกเหลว (Liquid-crystal Polarizing Shutter) แผ่นบังแถบขนานแบบผลึกเหลว (Liquid-crystal Parallax Barrier) และเครื่องแสดงผลหลายระนาบ (Multiplanar Display) ต่างก็เป็นส่วนผลักดันให้การพัฒนาการแสดงภาพสามมิติแบบทันที มีประสิทธิภาพสูงมากขึ้น อุปกรณ์แสดงภาพสามมิติพอแบ่งออกได้เป็นสามลักษณะคือ แบบโฮโลกราฟิกส์ (Holographics) แบบหลายระนาบ (Multi-planar) และแบบภาพคู่สเตอริโอ (Stereo pair)

2.2.1 โฮโลแกรม

โฮโลแกรม (Hologram) เห็นจะเป็นการแสดงภาพสามมิติที่เป็นที่คุ้นเคยกันโดยทั่วไป ตัวอย่างเช่นรูปที่ปรากฏอยู่บนบัตรเครดิตทั่วไป ในการสร้างภาพโฮโลแกรมนั้น ผู้สร้างจะทำการบันทึกรูปแบบการแทรกซ้อนที่เกิดจากลำแสงเลเซอร์สองลำ ที่ยิงจากแหล่งเดียวกันลงบนสื่อบันทึกภาพความละเอียดสูง ลำแสงหนึ่งยิงตรงเข้าหาสื่อบันทึกภาพ ในขณะที่อีกลำแสงหนึ่งสะท้อนจากวัตถุและเกิดการแทรกซ้อนกับลำแสงอ้างอิงลำแรก รูปแบบการแทรกซ้อนของลำแสงนี้เอง ที่เอื้ออำนวยการสร้างภาพสามมิติของวัตถุนั้นในภายหลังได้ ผู้มองสามารถโยกย้ายมุมมอง เพื่อเปลี่ยนมุมมองการเห็นภาพสามมิติของวัตถุได้ อย่างไรก็ตาม เทคโนโลยีทางด้านโฮโลแกรมยังก้าวไปไม่ถึงจุดที่เอื้ออำนวยการแสดงภาพแบบทันทีได้ เนื่องจากต้องการความเร็วของการประมวลผลที่สูง

2.2.2 การแสดงแบบหลายระนาบ

การแสดงภาพแบบหลายระนาบนั้น จะใช้การแบ่งวัตถุออกเป็นร้อยๆ หรืออาจเป็นพันๆ ระนาบ (เหมือนกับที่ใช้ในเครื่อง CAT scan) จากนั้นทำการวาดจุดต่างๆ ของวัตถุในแต่ละระนาบ โดยทั่วไประบบการแสดงภาพแบบนี้จะใช้กระจกสะท้อนจุดที่วาดจากแสงเลเซอร์ หรือจากหลอดภาพ ตัวอย่างเช่นเทคนิคการใช้กระจกแปรจุดรวม (Varifocal Mirror Technique) จะทำการแบ่งภาพสามมิติออกเป็นหลายพัน

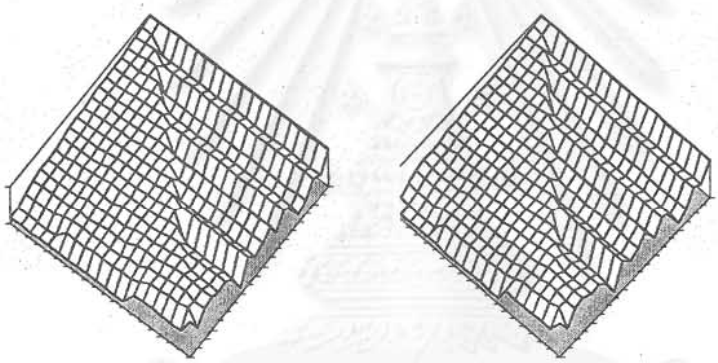
---

2 David F. Mcallister, "3-D Displays", *BYTES*, May 1992, pp.183-188.

ระนาบ จากนั้นวาดจุดต่างๆบนแต่ละระนาบด้วยหลอดภาพ แล้วใช้กระจกโค้งที่สั่นด้วยความถี่ (เช่น 30Hz) สะท้อนภาพจากหลอดภาพเข้าสู่ตาผู้มอง ตัวรับภาพในตามนุษย์จะจดจำภาพที่มองเห็นในช่วงเวลาต่างๆ (temporal persistent) ดังนั้นภาพที่เห็นจากกระจกที่เลื่อนได้ (อย่างรวดเร็ว) จะทำให้โสตประสาทการมองเห็นของมนุษย์รวมภาพต่างๆ (ในแต่ละระนาบ) เข้าด้วยกันจนเกิดการรับรู้ภาพสามมิติที่มีปริมาตร ภาพสามมิติที่แสดงด้วยวิธีนี้จะเป็นภาพที่โปร่งใสและมีเพียงสีเดียว

2.2.3 ภาพคู่สเตอริโอ

เทคนิคการแสดงภาพคู่สเตอริโอนี้คือ การแสดงภาพที่เหมาะสมสำหรับตาทั้งสองโดยอาศัยหลักการรับรู้ความลึกนูนจากความแตกต่างของภาพที่มองเห็นของตาทั้งสอง นั่นคือแสดงภาพๆ หนึ่งสำหรับตาซ้าย และอีกภาพหนึ่งสำหรับตาขวา ถ้าภาพทั้งสองมีความแตกต่างกันของระยะวัตถุตามแนวนอน จะทำให้ผู้มองรับรู้ความลึกของวัตถุจากความแตกต่างของระยะวัตถุดังกล่าว ตัวอย่างเช่น รูปที่ 2-1 แสดงภาพของระนาบสามมิติสองภาพสำหรับตาทั้งสอง ผู้สังเกตสามารถรับรู้ความลึกได้ หากเพ่งบนรูปดังกล่าว ปรับระยะโฟกัสของตา (หรือปรับระยะของกระดาษกับนัยน์ตา) จนทำให้ภาพทั้งสองรวมกันตรงกลาง ซึ่งคือตำแหน่งซึ่งตาทั้งสองมองเห็นเฉพาะภาพสำหรับตาแต่ละข้าง ผู้สังเกตจะรับรู้ความลึกได้

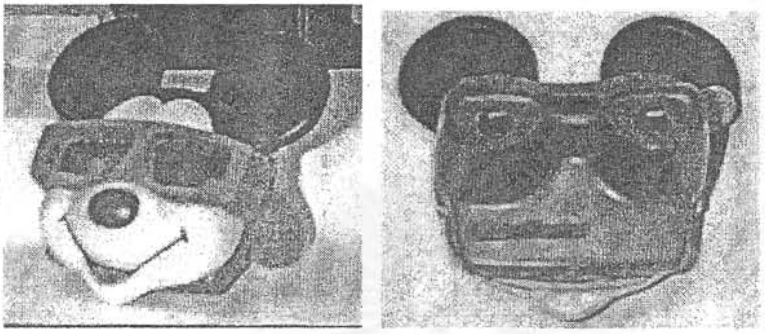


รูปที่ 2-1 ภาพคู่สเตอริโอสำหรับการรับรู้ภาพสามมิติ

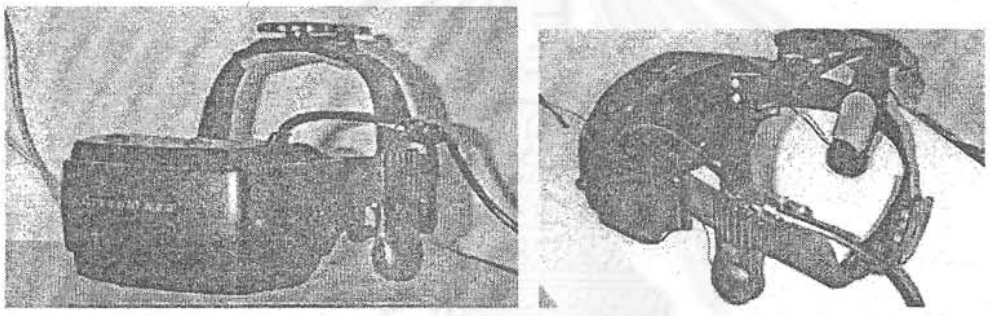
เทคนิควิธีการแสดงภาพทั้งสองนี้ แบ่งออกได้เป็น การแสดงภาพคู่พร้อมกัน (Time-parallel) และการแสดงภาพคู่สลับเชิงเวลา (Time-multiplexed)

การแสดงภาพคู่พร้อมกันที่เป็นที่รู้จักกันมากคือ กล้องดูรูปภาพนิ่งการ์ตูนสามมิติ ViewMaster® (รูปที่ 2-2) ผู้มองมองผ่านช่องมองสองช่อง แต่ละช่องจะมีภาพสองภาพสำหรับตาแต่ละข้าง นอกจากนี้ยังมีผู้ผลิตกล้องดูภาพนิ่งและภาพเคลื่อนไหวสามมิติที่ใช้จอภาพผลึกเหลวขนาดเล็กสองจอภาพสำหรับตาแต่ละข้าง ประกอบกันเป็นหมวกเพื่อให้ผู้ชมสวม โดยที่ตำแหน่งของจอภาพทั้งสองจะตรงกับดวงตาของผู้ชมพอดี ตัวอย่างเช่นหมวกสวมหัว CyberMaxx® ของบริษัท VictorMax (รูปที่ 2-3) เป็นต้น อีกวิธีหนึ่งที่เป็นที่รู้จักคือการแสดงภาพยนตร์สามมิติในสมัยก่อน ซึ่งใช้วิธีที่เรียกว่า Anaglyph ที่แสดงภาพสำหรับตาทั้งสองบนจอภาพหนึ่งจอพร้อมๆกัน โดยใช้สีที่ต่างกันสำหรับภาพทั้งสอง เช่นสีแดงกับสีเขียว เป็นต้น โดยที่ผู้มองจะต้องสวมแว่นที่มีตัวกรองสีเขียวและแดง วิธีนี้มีปัญหาเรื่องภาพหลอน เนื่องจากตัวกรองแสงไม่

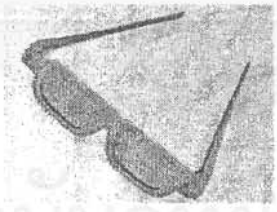
สามารถกรองแสงได้หมดจริง จึงทำให้เกิดการรับภาพสำหรับตาอีกข้างหนึ่งด้วย การแสดงภาพยนตร์สามมิติแบบ Anaglyph นี้ได้รับการปรับปรุงให้ดีขึ้น ด้วยการฉายภาพจากกล้องฉายสองกล้องผ่านแผ่นกรองบังคับทิศทางที่ต่างกันของคลื่นแสง โดยที่ทิศทางคลื่นของการฉายภาพทั้งสองจะตั้งฉากซึ่งกันและกัน แว่นกรองทิศทางแสง (รูปที่ 2-4) ที่ผู้ชมสวมจะกรองทิศของคลื่นที่ต่างๆ สำหรับตาทั้งสอง ดังนั้น แผ่นกรองจะกั้นภาพที่ไม่ตรงทิศออกได้ [3]



รูปที่ 2-2 กล้องดูรูปภาพนิ่งสามมิติ ViewMaster®



รูปที่ 2-3 หมวกสวมพร้อมจอภาพคู่ CyberMaxx®



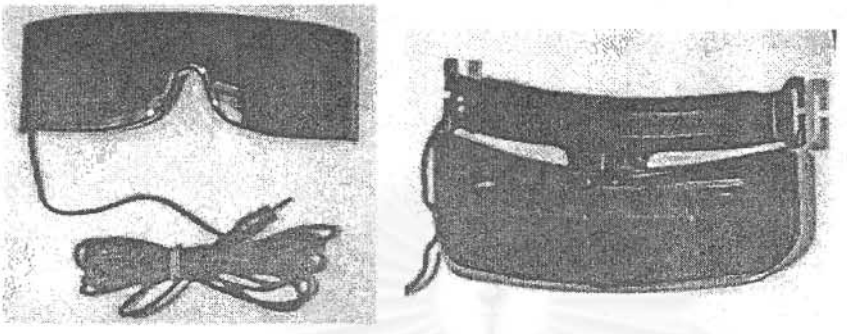
รูปที่ 2-4 แว่นกรองทิศทางแสง

การแสดงภาพคู่สลับเชิงเวลา มีหลักการทำงานคือการแสดงภาพสำหรับตาซ้ายในขณะที่เปิดตาซ้ายแต่ปิดตาขวา สลับกับการแสดงภาพตาขวาในขณะที่เปิดตาขวาแต่ปิดตาซ้าย หากการสลับดังกล่าวกระทำได้ในเวลาอันสั้น โสตประสาทการรับรู้จะสามารถรวมภาพทั้งสองและรับรู้ความลึกได้ การเปิดปิดตากระทำได้ในสองลักษณะคือ การใช้แว่นแบบ Active กับแบบ Passive สำหรับแบบ Active (รูปที่ 2-5) นั้นตัวแว่นเองจะทำการเปิดหรือบังการมองของนัยน์ตา โดยการใช้แว่นที่ทำจากแผ่นผลึกเหลวที่สามารถ

---

[3] ภาพสามมิติในรูปแบบต่างๆสามารถหาชมได้จาก http://www.3d-web.com

กั้นหรือยอมให้แสงผ่านตามลักษณะสัญญาณที่ป้อนเข้าที่แผ่นผลึกเหลว การเปิดหรือปิดแว่นนี้จะกระทำแบบประสานจังหวะไปกับการแสดงภาพตาซ้ายและขวาสลับกันบนจอภาพ สำหรับแว่นแบบ Passive (รูปที่ 2-4) นั้นการเปิดปิดภาพจะกระทำที่หน้าจอโดยที่ฉากหน้าจอภาพจะสามารถควบคุมทิศทางคลื่นแสงได้ โดยให้ภาพตาซ้ายและภาพตาขวาถูกแสดงสลับกันไป โดยมีคลื่นแสงที่ถูกบังคับทิศให้ตั้งฉากซึ่งกันและกัน ผู้ชมก็เพียงสวมแว่นกรองทิศคลื่น ในลักษณะเดียวกันกับการชมภาพยนตร์สามมิติ



รูปที่ 2-5 แว่นตาผลึกเหลวสำหรับมองภาพสามมิติสลับเชิงเวลา

เทคนิคการใช้ภาพคู่สเตอริโอนี้จะมีราคาย่อมเยาที่สุด วิธีนี้สามารถใช้ภาพสีเหมือนจริงที่สร้างขึ้นด้วยเทคนิคทางคอมพิวเตอร์อื่นๆได้

2.3 การแสดงภาพคู่สเตอริโอสลับเชิงเวลา [4]

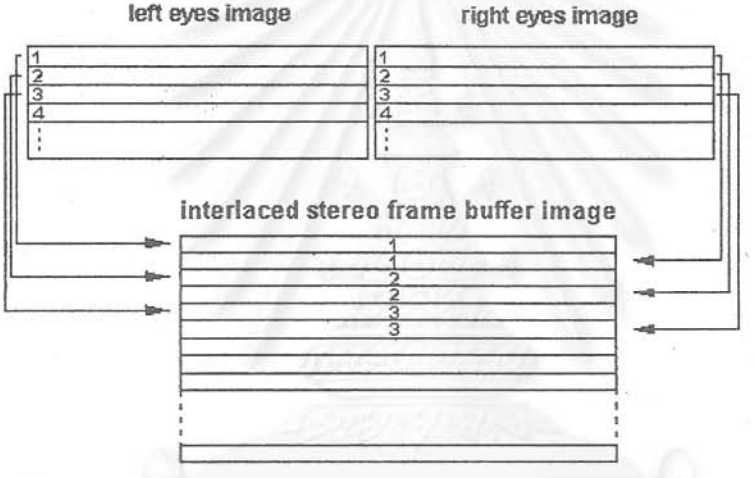
ขบวนการการแสดงภาพคู่สเตอริโอสลับเชิงเวลาประกอบด้วยการนำภาพตาซ้ายแสดงบนจอภาพในขณะที่ปิดแผ่นบังตาขวาให้ทึบ และจากนั้นนำภาพตาขวาแสดงบนจอภาพในขณะที่ปิดแผ่นบังตาซ้ายให้ทึบ ทำเช่นนี้สลับกันไปตามความถี่ของการกวาดภาพตามแนวตั้ง (vertical retrace) ของจอภาพที่ใช้ วิธีการแสดงภาพคู่สลับเชิงเวลาดังกล่าวกระทำได้หลายวิธีดังนี้

1. *การแสดงภาพเต็มจอภาพแบบพลิกหน้า* วิธีนี้แสดงภาพสำหรับตาซ้ายและปิดแผ่นบังตาขวา จากนั้นรอจนได้รับสัญญาณกวาดภาพตามแนวตั้ง จึงเริ่มแสดงภาพสำหรับตาขวาและปิดแผ่นบังตาซ้าย กระทำเช่นนี้สลับกันไป โดยทั่วไปผลิตภัณฑ์แสดงภาพสามมิติที่ใช้แว่นตาผลึกเหลวสำหรับการเล่นวิดีโอเกมจะใช้วิธีนี้
2. *การแสดงภาพผสานเส้นคู่เส้นคี่แบบสอดสลับ* วิธีนี้จะนำภาพคู่สเตอริโอทั้งสองรูปมาผสานกันเป็นภาพเดียวกันก่อน โดยจะกำหนดให้ภาพสำหรับตาซ้ายใช้เนื้อที่เฉพาะเส้นวาดตามแนวนอนที่มีหมายเลขกำกับเป็นเลขคี่เท่านั้น ส่วนภาพสำหรับตาขวาใช้เส้นวาดตามแนวนอนที่มีหมายเลขกำกับเป็นเลขคู่เท่านั้น ดังนั้นเมื่อดูภาพที่ผสานแล้ว จะเห็นภาพคู่ถูกรวมเข้าด้วย

---

[4] StereoGraphics Developers' Handbook : Background on Creating Images for CrystalEyes® and SimulEyes®, http://www.stererographics.com

กันโดยใช้เส้นวาดตามแนวนอนสลับกันไป (รูปที่ 2-6) จากนั้นแสดงภาพผสานดังกล่าวออกจอภาพแต่จะต้องกำหนดให้จอภาพแสดงภาพแบบสอดสลับ (interlace) การแสดงภาพในลักษณะแบบนี้ จอภาพจะกวาดลำแสงอิเล็กตรอน เพื่อแสดงเส้นกวาดตามแนวนอนเฉพาะเส้นคี่ทีละเส้นจากเส้นบนลงมาเรื่อย เมื่อกวาดเส้นคี่หมดแล้ว จึงจะเริ่มแสดงเส้นกวาดตามแนวนอนเฉพาะเส้นคู่ทีละเส้น เมื่อครบเส้นคู่ ก็กลับไปแสดงเส้นคี่ กระทำเช่นนี้ไปเรื่อยๆ เมื่อใดที่จอภาพแสดงภาพของเส้นกวาดหมายเลขคี่ (ภาพสำหรับตาซ้าย) ก็ทำการปิด แผ่นบังตาขวา และในทางกลับกัน เมื่อจอภาพแสดงภาพของเส้นกวาดหมายเลขคู่ (ภาพสำหรับตาขวา) ก็ทำการปิด แผ่นบังตาซ้าย ตัวอย่างผลิตภัณฑ์ที่ใช้วิธีนี้ในการแสดงภาพมิติ มีอาทิ แว่นผลึกเหลว SimulEyes® ของบริษัท StereoGraphics [5] แว่นผลึกเหลว 3D-Max® ของบริษัท Kasan Electronics [6] เป็นต้น



รูปที่ 2-6 การผสานภาพคู่สเตอริโอเส้นคู่เส้นคี่แบบสอดสลับ

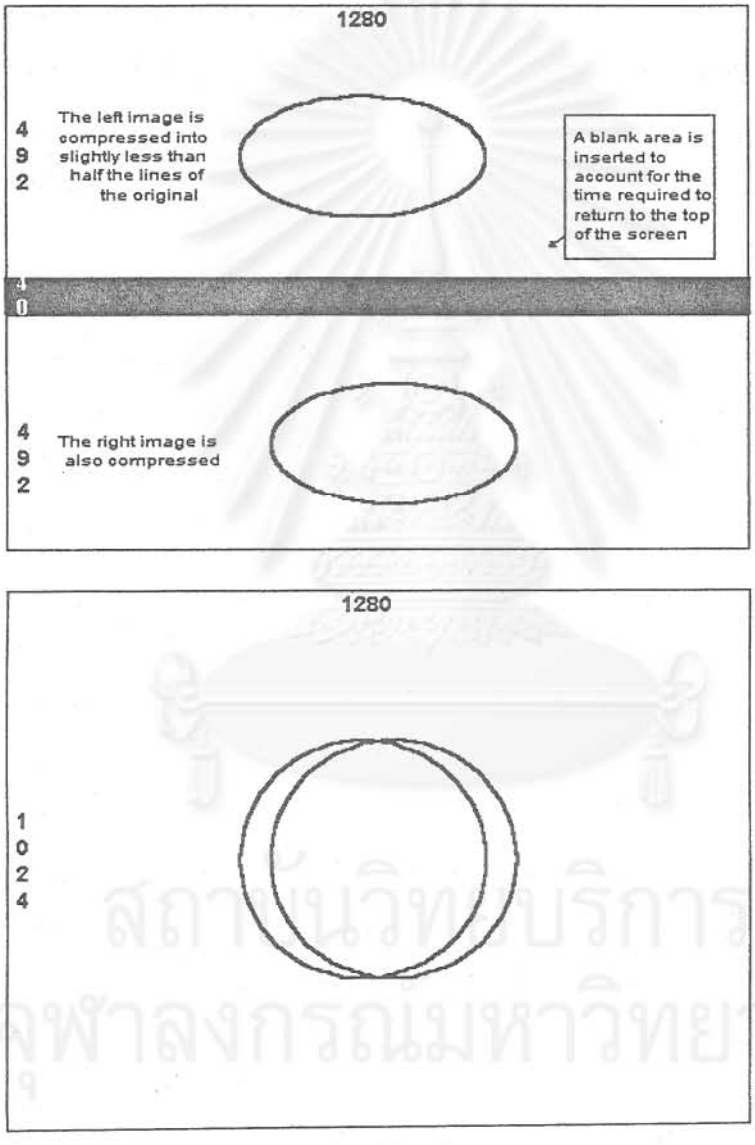
3. *การแสดงภาพแบบซ้ายบนขวาล่าง* วิธีนี้จะนำภาพคู่สเตอริโอทั้งสองภาพแสดงบนจอภาพโดยแสดงภาพซ้ายที่ครึ่งบนของจอภาพ และแสดงภาพขวาที่ครึ่งล่างของจอภาพ (รูปที่ 2-7) สัญญาณจากแผงวงจรแสดงผลของเครื่องคอมพิวเตอร์จะถูกส่งให้วงจรควบคุมการแสดงภาพสามมิติภายนอก ที่ส่งผ่านสัญญาณภาพดังกล่าวไปยังจอภาพตามปกติ เพียงแต่ว่าเมื่อใดสัญญาณที่ได้รับเป็นสัญญาณที่บ่งบอกว่าได้แสดงภาพครึ่งบนไปแล้ว ตัววงจรภายนอกนี้จะสร้างสัญญาณกวาดภาพแนวตั้งไปยังจอภาพ เสมือนกับเป็นการหลอกจอภาพว่าแสดงภาพทั้ง

---

5 http://www.stereographics.com/simueyes.html

6 http://pcvr.kasan.co.kr/3d_max.html

จอภาพเสร็จแล้ว ดังนั้นเมื่อสัญญาณของภาพครึ่งล่างถูกส่งผ่านไปยังจอภาพก็เป็นการแสดงภาพทั้งภาพครึ่งล่างนี้เริ่มต้นจากเส้นบนสุดของจอภาพ ดังนั้นหากภาพที่แสดงออกจากเครื่องคอมพิวเตอร์มีความถี่การกวาดภาพตามแนวตั้ง 60Hz จอภาพจะได้รับสัญญาณเป็นความถี่ 120Hz (เทคนิคนี้บางครั้งเรียกว่าการทวีคูณสัญญาณซิงค์ - Sync Doubling) เนื่องจากภาพแต่ละครึ่งที่ถูกวาดจะถูกขยายให้เต็มจอ ดังนั้นการวาดภาพจะต้องปรับขนาดของรูปให้เล็กลงด้วยสัดส่วนที่ถูกต้อง ผลิตภัณฑ์ที่ใช้วิธีนี้ในการแสดงภาพสามมิติคือ แว่นผลึกเหลว CrystalEyes® ของบริษัท StereoGraphics [7] ที่ใช้กับเครื่องคอมพิวเตอร์ระดับสถานีวิศวกรรม เป็นต้น



รูปที่ 2-7 การแสดงภาพซ้ายบนขวาล่าง และผลที่ได้

7 http://www.stereographics.com

จะเห็นได้ว่าเทคนิคแบบพลิกหน้านั้นใช้เนื้อที่ของหน่วยความจำการแสดงภาพเป็นสองเท่าของภาพ ดังนั้นหากต้องการแสดงภาพที่มีความละเอียดสูงจะใช้ปริมาณหน่วยความจำในการแสดงผลมากเป็นพิเศษ สำหรับเทคนิคแบบสอดสลับนั้นภาพทั้งสองถูกจัดเก็บในหน่วยความจำแสดงผลพร้อมๆกัน จึงไม่ต้องเสียเวลาสลับภาพซ้ายและขวาเข้าและออกด้วยซอฟต์แวร์ (ที่ทำในแบบพลิกหน้า) จากหน่วยความจำการแสดงผล แต่ต้องเสียเวลาการผสานภาพคู่ให้เป็นภาพเดียว และได้ความละเอียดของภาพเป็นครึ่งหนึ่งของความละเอียดจอภาพ ในขณะที่วิธีแบบบนล่างนั้นไม่ต้องเสียเวลาการผสานภาพ ซอฟต์แวร์ที่สร้างภาพคู่นั้นสามารถสร้างภาพบนแต่ละครึ่งของจอภาพได้เลย (ให้สังเกตว่าในวิธีที่สอง จะใช้โปรแกรมสร้างภาพคู่บันทึกภาพในหน่วยความจำแสดงผลแบบสลับเส้นคู่และคี่โดยตรงนั้น ไม่เหมาะสมในประเด็นทางด้านความเร็ว) แต่อย่างไรก็ตามวิธีที่สามนั้นต้องใช้วงจรเสริมภายนอกที่ยุ่งยากกว่าสองแบบแรก

สำหรับงานสิ่งประดิษฐ์นี้จะใช้วิธีแบบพลิกหน้าและแบบสอดสลับในการสร้างระบบการแสดงผลสามมิติ ทั้งนี้เนื่องจากต้องการอุปกรณ์และวงจรเสริมภายนอกไม่ยุ่งยากมากนัก อีกทั้งความก้าวหน้าทางเทคโนโลยีของเครื่องคอมพิวเตอร์ส่วนบุคคลที่มีประสิทธิภาพสูงขึ้น ในราคาที่ถูกลง สามประการดังนี้

1. *ความจุของหน่วยความจำแสดงผล* ปัจจุบันแผงวงจรแสดงผลสามารถบรรจุหน่วยความจำแสดงผลได้จำนวนมาก (โดยทั่วไปได้ถึง 4MB) ในราคาถูก [8] ยังผลสามารถบรรจุทั้งภาพสำหรับตาซ้ายและภาพสำหรับตาขวาเต็มทั้งจอภาพได้ในหน่วยความจำแสดงผล ดังนั้นเมื่อต้องการสลับภาพที่จะแสดงออกจอภาพก็กระทำได้โดยการใช้เทคนิคการพลิกหน้าของแผงวงจรแสดงผลตามมาตรฐาน VESA

2. *ความเร็วของหน่วยประมวลผล* หน่วยประมวลผลหลักของเครื่องคอมพิวเตอร์ส่วนบุคคลมีความเร็วเพิ่มขึ้นเรื่อยๆ ในราคาที่ถูกลง ทั้งนี้เนื่องมาจากการปรับปรุงทั้งทางด้านสถาปัตยกรรม เทคโนโลยีการผลิตที่ทำให้รับความถี่สัญญาณนาฬิกาที่สูงขึ้น ระบบการส่งถ่ายข้อมูล ขนาดของหน่วยความจำความเร็วสูงประเภทแคช และอื่นๆ ทำให้การคำนวณ วาด และย้ายภาพ ไปยังหน่วยความจำแสดงผลกระทำได้ทัน

3. *ความถี่การกวาดภาพของจอภาพ* เนื่องจากการแสดงภาพสามมิติแบบสเตอริโอสลับเชิงเวลาดังที่ได้กล่าวมาแล้วนั้น จำนวนภาพที่แสดงบนจอภาพต่อหนึ่งวินาที มีครึ่งหนึ่งเป็นภาพสำหรับตาซ้าย อีกครึ่งหนึ่งเป็นภาพสำหรับตาขวา ทำให้ความถี่การแสดงภาพบนจอภาพลดลงครึ่งหนึ่ง ความถี่ต่ำจะมีผลทำให้เกิดภาพกระพริบเป็นที่รำคาญการมองภาพ ความถี่นี้ถูกกำหนดโดยคุณสมบัติของทั้งแผงวงจรแสดงผลและตัวจอภาพ ซึ่งในปัจจุบันสามารถเพิ่มความถี่ดังกล่าวได้สูงประมาณ 100 Hz แบบไม่สอดสลับ (non-interlace) ในราคาถูก

---

8 ในขณะที่เขียนรายงานวิจัยนี้ (วันที่ 25 ตุลาคม 2540) หน่วยความจำแสดงผลแบบ EDO RAM ราคาประมาณ 500 บาท ต่อ 1MB ใช้กับแผงวงจรที่ใช้ตัวประมวลผลการแสดงผล S3 Trio64+

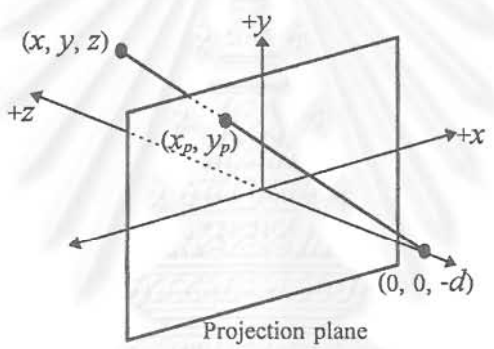
2.4 การสร้างภาพคู่สามมิติ [9]

ในระบบการแสดงภาพสเตอริโอสามมิตินั้น จะต้องสร้างภาพสองภาพสำหรับตาแต่ละข้าง ภาพทั้งสองเกิดจากการคำนวณภาพฉาย (Projection) ของวัตถุบนระนาบการฉาย ซึ่งสามารถคำนวณได้ในสองลักษณะคือ การฉายภาพนอกแกน (Off-axis Projection) และการฉายภาพบนแกน (On-axis Projection) ถึงแม้ว่าวิธีทั้งสองจะเหมือนกันในทางคณิตศาสตร์ก็ตาม แต่ในทางปฏิบัติแล้วจะได้ภาพที่แตกต่าง ดังจะได้นำเสนอต่อไป

2.4.1 การฉายภาพ

กำหนดให้พิกัดของจุด p ในสามมิติ คือ (x, y, z) การฉายจุด p ลงบนระนาบ $(x, y, 0)$ โดยมีจุดศูนย์กลางการฉาย (center of projection) อยู่ที่ $(0, 0, -d)$ ดังแสดงในรูปที่ 2-8 จะได้จุดฉาย (x_p, y_p) บนระนาบดังกล่าว ดังนี้

$$x_p = \frac{xd}{d+z} \qquad y_p = \frac{yd}{d+z} \tag{1}$$



รูปที่ 2-8 การฉายภาพ

2.4.2 การฉายภาพนอกแกน

การฉายภาพนอกแกนเพื่อสร้างภาพสองภาพสำหรับตาซ้ายและตาขวานั้น ใช้จุดศูนย์กลางการฉายสองจุด คือจุดศูนย์กลางการฉายด้านขวา (Rcop) และจุดศูนย์กลางการฉายด้านซ้าย (Lcop) จุดสองจุดนี้อยู่นอกแกน z เป็นระยะทาง $e/2$ นั่นคือพิกัดของ Rcop และ Lcop เป็น $(e/2, 0, -d)$ และ $(-e/2, 0, -d)$ ตามลำดับ โดยที่ e คือระยะห่างระหว่างจุดทั้งสอง (ระยะห่างนี้มีผลต่อการแสดงภาพแบบสเตอริโอซึ่งจะได้กล่าวในภายหลัง) เมื่อใช้ Rcop เป็นจุดศูนย์กลางการฉายจุดบนระนาบสำหรับภาพตาขวา และใช้ Lcop เป็นจุดศูนย์กลางการฉายจุดบนระนาบสำหรับตาซ้าย จะได้ภาพคู่ของวัตถุบนระนาบสำหรับตาทั้งสองเพื่อการรับรู้ภาพสามมิติแบบสเตอริโอ สมมติให้ p เป็นจุดบนวัตถุ มีพิกัด (x, y, z) การฉายภาพนอกแกนของ

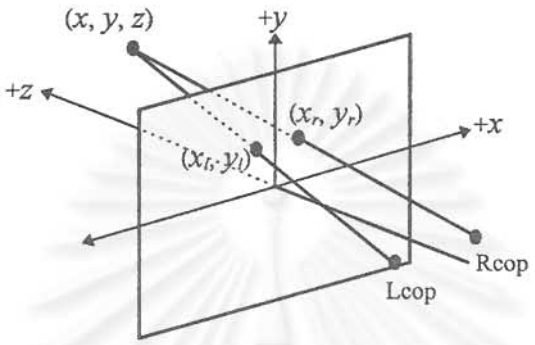
---

9 Larry F. Hodges, "Tutorial: Time-Multiplexed Stereoscopic Computer Graphics", *IEEE Computer Graphics & Applications*, March 1996, pp.20-30.

จุด p จะได้จุดสองจุดคือ จุด (x_r, y) และจุด (x_l, y) บนระนาบ ซึ่งได้การฉายโดยใช้จุด Rcop และ Lcop เป็นจุดศูนย์กลางการฉายตามลำดับ ดังนี้ (ดูรูปที่ 2-9 ประกอบ)

$$x_l = \frac{xd - \frac{ze}{2}}{d+z} \qquad y_l = \frac{yd}{d+z} \tag{2}$$

$$x_r = \frac{xd + \frac{ze}{2}}{d+z} \qquad y_r = \frac{yd}{d+z} \tag{3}$$



รูปที่ 2-9 การฉายภาพนอกแกน

2.4.3 การฉายภาพบนแกน

วิธีนี้จะใช้จุดศูนย์กลางการฉายที่พิกัด (0, 0,-d) ที่อยู่บนแกน z แต่จะมีขั้นตอนการย้ายจุดที่วัตถุก่อนการฉาย และย้ายจุดที่ระนาบหลังการฉาย เป็นขั้นตอนตามตารางข้างล่างนี้

ขั้นตอนการฉายภาพบนแกน	*ภาพตาขวา*	*ภาพตาซ้าย*
1. เริ่มด้วยการย้ายจุด (x, y, z) ตามแนวนอนไป $e/2$ (หรือ $-e/2$)	$x - \frac{e}{2}$	$x + \frac{e}{2}$
2. จากนั้นฉายจุดที่ย้ายตำแหน่งแล้วลงบนระนาบ โดยใช้ (0, 0,-d) เป็นจุดศูนย์กลางการฉาย (ซึ่งอยู่บนแกน z)	$\frac{d\left(x - \frac{e}{2}\right)}{d+z}$	$\frac{d\left(x + \frac{e}{2}\right)}{d+z}$
3. แล้วจึงย้ายจุดฉายบนระนาบกลับตามแนวนอนไป $-e/2$ (หรือ $e/2$)	$\frac{d\left(x - \frac{e}{2}\right)}{d+z} + \frac{e}{2}$	$\frac{d\left(x + \frac{e}{2}\right)}{d+z} - \frac{e}{2}$

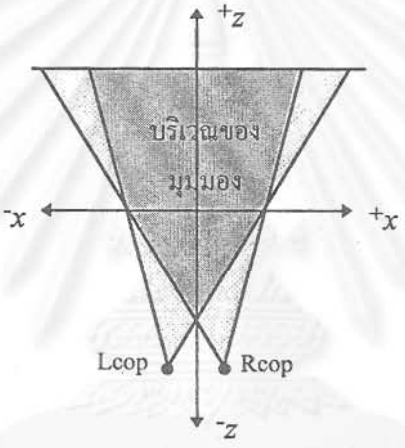
จากวิธีดังกล่าวจะสังเกตได้ว่าจะได้ผลเช่นเดียวกับ การฉายภาพนอกแกนในหัวข้อที่แล้ว ซึ่งมีสมการการฉายภาพนอกแกนที่เขียนได้ใหม่ ในรูปแบบเช่นเดียวกับสมการการฉายภาพบนแกน ดังต่อไปนี้

$$x_l = \frac{xd - \frac{ze}{2}}{d+z} = \frac{d\left(x + \frac{e}{2}\right)}{d+z} - \frac{e}{2} \quad (4)$$

$$x_r = \frac{xd + \frac{ze}{2}}{d+z} = \frac{d\left(x - \frac{e}{2}\right)}{d+z} + \frac{e}{2} \quad (5)$$

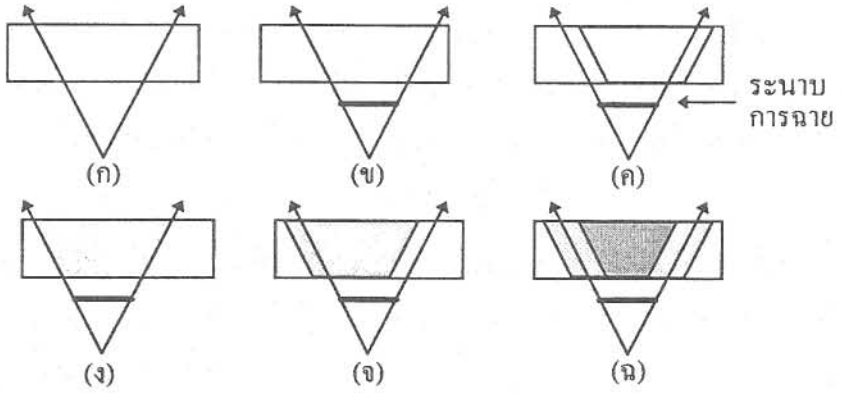
2.4.4 บริเวณของมุมมอง

การฉายภาพแบบนอกแกน และแบบบนแกน มีรูปลักษณะทางคณิตศาสตร์ที่เหมือนกัน แต่จะต่างกันในเรื่องของบริเวณที่ถูกฉายบนระนาบของตาทั้งสองซึ่งเรียกว่า บริเวณของมุมมอง (Field of view) ในกรณีของการฉายภาพนอกแกนนั้น บริเวณของมุมมองหาได้จากบริเวณร่วมของการฉายภาพจากจุดศูนย์กลางทางซ้ายและทางขวา ที่ฉายลงบนพื้นที่ร่วมกันบนระนาบ ดังแสดงในรูปที่ 2-10



รูปที่ 2-10 บริเวณของมุมมอง สำหรับการฉายภาพนอกแกน

สำหรับการฉายภาพบนแกนนั้น ให้พิจารณาจากรูปที่ 2-11 กล่องสี่เหลี่ยมผืนผ้าแทนขอบเขตของวัตถุที่สนใจที่ถูกฉายลงบนระนาบการฉาย ในกรณีของภาพฉายสำหรับตาขวานั้นหาได้จากการย้ายตำแหน่งของทุกๆ วัตถุในขอบเขตที่สนใจไปทางซ้าย เป็นระยะทาง $e/2$ (รูปที่ 2-11 ข) จากนั้นฉายภาพลงบนระนาบแล้วจึงเลื่อนภาพฉายบนระนาบทั้งภาพกลับไปทางขวาเป็นระยะทาง $e/2$ (รูปที่ 2-11 ค) ให้สังเกตว่าการย้ายภาพกลับนี้เอง ที่ทำให้เกิดการสูญเสียข้อมูลที่ทำการฉายแล้วทางด้านซ้ายของจอภาพ นั่นคือเกิดช่องว่างทางด้านซ้าย ในทำนองเดียวกันการสร้างภาพฉายสำหรับตาซ้ายนั้น หาได้จากการย้ายตำแหน่งของทุกๆ วัตถุในขอบเขตที่สนใจไปทางขวา ฉายภาพลงบนระนาบ (รูปที่ 2-11 ง) แล้วย้ายภาพที่ได้กลับไปทางซ้าย (รูปที่ 2-11 จ) เมื่อรวมภาพฉายทั้งสอง จะได้บริเวณของมุมมองร่วมของตาซ้ายและขวาในรูปที่ 2-11 ฉ บริเวณของมุมมองที่ได้นี้จะครอบคลุมบริเวณได้น้อยกว่าการฉายภาพแบบนอกแกน ซึ่งโดยทั่วไปอาจแตกต่างได้มากถึง 40%



รูปที่ 2-11 บริเวณของมุมมอง สำหรับการฉายภาพบนแกน

แม้ว่าการฉายภาพแบบบนแกนจะคลอบคลุมบริเวณได้น้อยกว่า การฉายภาพแบบนอกแกนนั้น วิธีการฉายภาพแบบบนแกนง่ายต่อการดำเนินการ โดยเฉพาะอย่างยิ่งกับส่วนแสดงผลกราฟิกส์ระดับเครื่องเวอร์คสเตชันที่สามารถทำการฉายภาพ การย้ายจุด และการย้ายภาพได้ด้วยฮาร์ดแวร์ ดังนั้นจะสามารถฉายภาพคู่สเตอริโอได้รวดเร็วกว่า

2.5 ปัจจัยที่มีผลกับคุณภาพของภาพ

คุณภาพของการรับรู้ภาพสามมิติโดยใช้วิธีการแสดงแบบภาพคู่สลับเชิงเวลานั้นมีหลายปัจจัยอันได้แก่ เรื่องของภาพหลอนและการรบกวนข้ามตา อัตราการแสองภาพของจอ และการเปลี่ยนขนาดของภาพ

2.5.1 ภาพหลอนและการรบกวนข้ามตา

ในระบบภาพสลับเชิงเวลาที่สมบูรณ์ เมื่อตาขวาจะต้องเห็นเฉพาะภาพสำหรับตาขวา และในทางกลับกัน ตาซ้ายจะต้องเห็นเฉพาะภาพสำหรับตาซ้าย แต่ในความเป็นจริงแล้ว การรบกวนข้ามตาระหว่างภาพจะเกิดขึ้น นั่นคือตาขวาอาจเห็นภาพของตาซ้ายบ้าง หรือในทางกลับกัน ปัจจัยที่มีผลต่อปริมาณการรบกวนข้ามตานั้นได้แก่

1. อัตราการส่งผ่านแสงของแว่นตาในสภาวะเปิดต่อสภาวะปิด (Dynamic Range) ถ้ามีค่าต่ำแสดงว่าขณะที่กำลังแสดงภาพสำหรับตาข้างหนึ่งนั้น ตาอีกข้างหนึ่งที่ถูกบังไว้นั้นกลับมองเห็นภาพนั้นบ้างลางๆ นั่นคือแว่นตาปิดบังตาข้างนั้นไม่สนิท
2. การคงอยู่ของฟอสฟอรัสบนจอภาพ (Phosphor Persistance) ในขณะที่เริ่มสลับการแสดงภาพสำหรับตาข้างใหม่ จะพบว่าภาพของตาอีกข้างหนึ่งที่แสดงไว้ก่อนหน้านี้จะยังไม่เลือนหายไปจากจอภาพ ปริมาณแสงจากฟอสฟอรัสที่ตกค้างมาจากภาพตรงข้ามจะวัดโดยใช้ระยะเวลาที่ความสว่างของฟอสฟอรัสลดลงเหลือ 10 เปอร์เซ็นต์ จากความสว่างเริ่มต้น ซึ่งมักขึ้นอยู่กับสีของวัตถุเช่น ฟอสฟอรัสหมายเลข P22 ที่ใช้กันมากในจอสีทั่วไป ฟอสฟอรัสสีแดงและน้ำเงินจะลดความสว่างลงเร็วกว่าฟอสฟอรัสสีเขียว

3. ตำแหน่งทางแนวตั้งของภาพ ผลกระทบเรื่องการคงอยู่ของฟอสฟอรัสยังขึ้นกับตำแหน่งของวัตถุบนจอภาพแบบราสเตอร์สแกน (Raster Scan) ฟอสฟอรัสที่อยู่ด้านบนของจอภาพจะมีเวลาในการสลายตัวมากกว่าบริเวณอยู่ด้านล่าง เมื่อชัตเตอร์เริ่มเปลี่ยนสถานะ ดังนั้นภาพที่อยู่ส่วนบนๆ จะได้รับผลกระทบน้อยกว่าบริเวณที่อยู่ด้านล่าง
4. ปัจจัยที่มีผลต่อการได้เห็นภาพหลอน (Ghosting) ระหว่างภาพของตาซ้ายและขวา ซึ่งเป็นผลมาจากการรบกวนข้ามตาได้แก่ ความสว่างของภาพ, ความซับซ้อนของพื้นผิว, และ การเหลื่อมใน แนวนอน

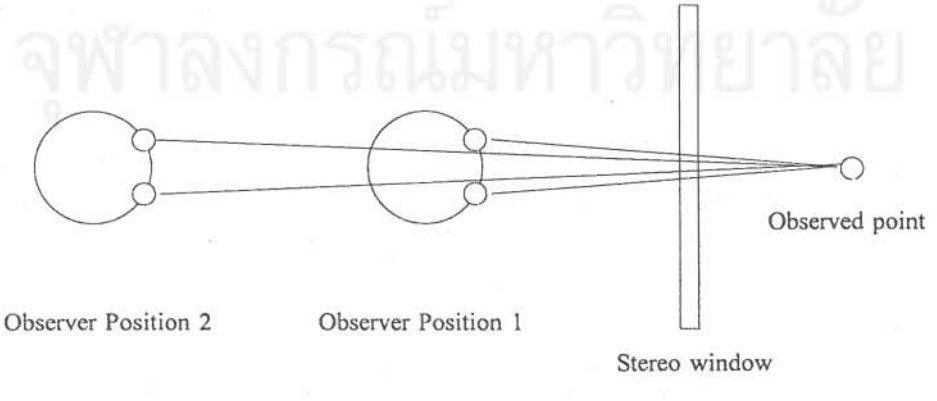
2.5.2 อัตราการกวาดภาพของจอแสดงผล

อัตราการกวาดภาพของจอแสดงผลจะมีผลต่อการกระพริบของภาพที่แสดงในจอส่วนใหญ่ที่มีอัตราการกวาดภาพเป็น 60 ภาพต่อวินาที จะทำให้ แต่ละตาได้รับภาพในอัตรา 30 ภาพต่อวินาที การสลับภาพมักจะทำโดยใช้ ฮาร์ดแวร์ และมีสองบัฟเฟอร์ (Buffer) การทำระบบแบบตอบโต้มักทำได้ยากเนื่องจากการเตรียมภาพสำหรับแต่ละตานั้น ต้องทำให้ได้ภายใน 1/60 วินาที เพื่อให้สามารถเปลี่ยนแปลงภาพได้ทันที โดยทั่วไปอาจลดความละเอียดตามแนวตั้งของจอภาพลง และเพิ่มบัฟเฟอร์อีกสองบัฟเฟอร์สำหรับการเตรียมภาพที่ใช้แสดงผล

2.6 การปรับขนาดของภาพ

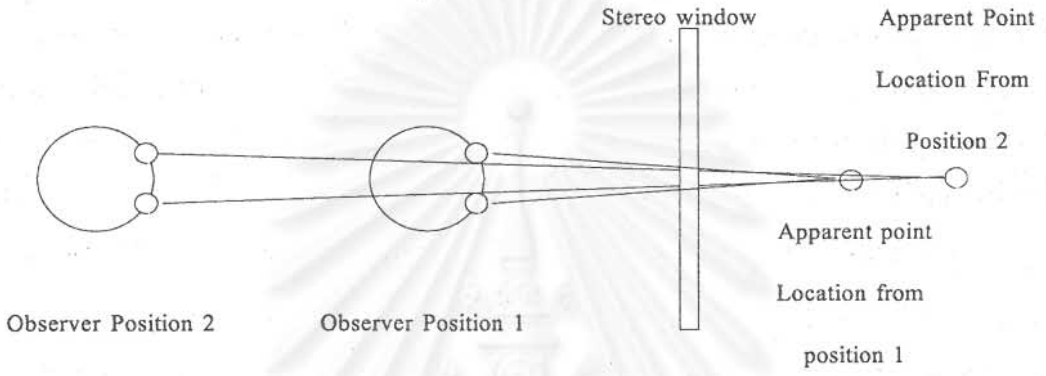
การปรากฏความลึกของจุดหรือวัตถุในระบบแสดงผลแบบสเตอริโอขึ้นกับตำแหน่งของผู้สังเกตเทียบกับจอภาพ ความสัมพันธ์นี้ได้แสดงไว้ในรูปที่ 2-12 เมื่อผู้สังเกตอยู่ห่างจากจอภาพมากขึ้น การเหลื่อมในแนวนอนสำหรับจุดคงที่จุดหนึ่งจะลดลง เมื่อผู้สังเกตอยู่ใกล้จอมากขึ้นการเหลื่อมในแนวนอนก็จะมากขึ้น

เนื่องจากระยะห่างของผู้สังเกตกับจอภาพมักไม่สามารถกำหนดได้อย่างตายตัวหรือไม่ทราบค่าโดยทั่วไปมักจะคำนวณภาพโดยกำหนดระยะห่างจากผู้สังเกตการณ์ถึงจอภาพเป็นระยะคงที่ ทำให้ระบบมีการเหลื่อมในแนวนอนแบบคงที่ (Fixed Horizontal Parallax) สำหรับจุดหนึ่งจากระนาบการฉาย ไม่ว่าผู้สังเกตจะอยู่ตำแหน่งใดก็ตาม



รูปที่ 2-12 ความเหลื่อมในแนวนอนที่เปลี่ยนไปเมื่อตำแหน่งของผู้สังเกตเปลี่ยน

ระบบที่มีการเหลื่อมในแนวนอนแบบคงที่ ก่อให้เกิดการบิดเบือนของภาพตามแกนการมองตั้งฉากกับระนาบของผู้มอง ดังแสดงไว้ในรูปที่ 2-13 เมื่อผู้สังเกตเคลื่อนออกไปไกลจากจอขึ้นภาพจะยาวเรียวขึ้นและเมื่อผู้สังเกตเคลื่อนที่เข้าใกล้จอมากขึ้นภาพก็จะหดลง เมื่อผู้สังเกตเคลื่อนจากอีกด้านหนึ่งไปอีกด้านหนึ่งภาพก็จะเลื่อนตามไปด้วย ในทางปฏิบัติแล้วการประยุกต์ใช้ต่อปัจจัยเหล่านี้จะทำการกำหนดตำแหน่งที่น้อยที่สุดในการมองภาพที่จะทำให้การปรับขนาดภาพ ในแนวจากด้านหน้าไปด้านหลังของภาพต่อการปรับสัดส่วนภาพในแนวขึ้นลงหรือซ้ายขวา



รูปที่ 2-13 ความบิดเบือนที่เกิดขึ้น เมื่อตำแหน่งของผู้สังเกตเปลี่ยนไป แต่ภาพที่แสดงมีขนาดเท่าเดิม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3. ระบบจินตทัศน์ข้อมูล

การจินตทัศน์ข้อมูล (Data Visualization) คือกระบวนการแปลงข้อมูลไปเป็นภาพ [1] โดยภาพที่แทนข้อมูลเหล่านี้ จะต้องสื่อถึงความหมายและความสัมพันธ์ของข้อมูลเหล่านั้น เพื่อให้ผู้ดูเล็งเห็นถึงพฤติกรรม ซึ่งอาจเป็นการเปลี่ยนแปลงของข้อมูล ข้อผิดปกติของบางกลุ่มข้อมูล หรือแนวโน้มของข้อมูล เป็นต้น การจินตทัศน์เป็นกระบวนการที่มีความจำเป็นอย่างยิ่ง ทั้งนี้เนื่องจากจำนวนข้อมูลที่สนใจนั้นโดยทั่วไปมักมีจำนวนมากเกินกว่าที่ผู้ศึกษาจะสามารถดูได้ทุกค่าของข้อมูล จึงจำต้องผ่านการจินตทัศน์เพื่อเห็นภาพรวมของกลุ่มข้อมูลที่ศึกษา ตัวอย่างเช่นการศึกษาสภาพการเปลี่ยนแปลงของข้อมูลการซื้อขายหุ้น ข้อมูลอุณหภูมิหรือความกดอากาศเพื่องานพยากรณ์อากาศ ข้อมูลแรงกระทบพื้นผิวยานพาหนะในอุโมงค์ลม ข้อมูลการอ้างอิงข้อมูลในหน่วยความจำของไมโครโปรเซสเซอร์ เป็นต้น เนื่องจากการจินตทัศน์เป็นเทคนิคการใช้ภาพเพื่อนำเสนอพฤติกรรมของข้อมูลนั้น การใช้ภาพเคลื่อนไหวในงานการจินตทัศน์นั้นจึงได้รับความนิยมมากโดยภาพเคลื่อนไหวที่นำเสนอนั้นจะแปรเปลี่ยนตามข้อมูลที่เปลี่ยนแปลงตามเวลา

สำหรับในโครงการสิ่งประดิษฐ์นี้ ผู้วิจัยมุ่งเน้นการใช้ระบบการจินตทัศน์ที่ได้รับความนิยมอยู่สองระบบคือ VTK (Visualization Toolkit) [1] ซึ่งเป็นระบบการสร้างการจินตทัศน์จากศูนย์วิจัยและพัฒนาของบริษัทเจนเนอรัล อิเล็กทริก ที่เป็นซอฟต์แวร์สาธารณะ และ 3D Studio MAX ซึ่งเป็นซอฟต์แวร์สร้างแบบจำลองทางด้านสามมิติของบริษัทออโตเดกส์ ที่ได้รับความนิยมเป็นอย่างสูงในงานการออกแบบ โดยได้พัฒนาส่วนเสริมให้กับระบบทั้งสองเพื่อให้สามารถแสดงภาพสามมิติแบบสเตอริโอสลับเชิงเวลา ทำให้การจินตทัศน์งานของระบบทั้งสองทางด้านสามมิติเป็นจริงมากขึ้น

ผู้วิจัยเลือกระบบทั้งสองมาศึกษาและเพิ่มเติมนั้น ด้วยเหตุผลดังต่อไปนี้

1. *ความนิยม* ระบบทั้งสองได้รับความนิยมจากผู้ใช้งานเป็นอย่างสูงตั้งแต่ผู้ใช้ระดับเริ่มต้นจนถึงระดับใช้เพื่องานวิจัย และงานอาชีพ VTK เป็นระบบงานการจินตทัศน์ข้อมูล ที่ได้รับความนิยมในวงการศึกษามาก ในขณะที่ 3D Studio MAX นั้นเป็นระบบเพื่อการจินตทัศน์ที่ได้รับความนิยมในงานออกแบบมาก ส่งผลให้มีกรณีศึกษาและตัวอย่างการใช้งานเพื่อการทดลองมากมาย
2. *เครื่องคอมพิวเตอร์ที่ใช้* ระบบทั้งสองสามารถทำงานได้กับเครื่องคอมพิวเตอร์ส่วนบุคคล ซึ่งในปัจจุบันมีประสิทธิภาพต่อราคาที่ดีมาก และเป็นเครื่องคอมพิวเตอร์ที่สามารถต่อพ่วงอุปกรณ์แว่นตาผลึกเหลวเพื่อการรับรู้ความลึกที่มีราคาถูกได้

---

[1] W. Schroeder, K. Martin, and B. Lorensen, "The Visualization ToolKit : An Object-Oriented Approach to 3D Graphics", Prentice Hall, 1996.

3. *การออกแบบเชิงวัตถุ* ระบบทั้งสองถูกพัฒนาขึ้นโดยใช้แนวทางการออกแบบเชิงวัตถุ (object-oriented design) ซึ่งสนับสนุนการเพิ่มเติมหน่วยทำงานใหม่ได้อย่างมีระบบ

สำหรับหัวข้อย่อยในบทนี้จะกล่าวถึงองค์ประกอบ คุณลักษณะ และการทำงานของระบบทั้งสอง

3.1 Visualization Toolkit

VTK (Visualization Toolkit) เป็นซอฟต์แวร์สำหรับงานจินตทัศน์และงานคอมพิวเตอร์กราฟิกส์สามมิติจากศูนย์วิจัยและพัฒนาของบริษัทเจนเนอรัล อิเล็กทริก มีจุดเด่นที่สำคัญดังต่อไปนี้

1. VTK ประกอบด้วยคลังคำสั่งที่เป็นคลาส (class) สำหรับภาษา C++ และภาษา Tcl/Tk [2] เพื่องานทางกราฟิกส์และการจินตทัศน์มากมาย การที่ VTK สนับสนุนทั้ง C++ และ Tcl/Tk การพัฒนางานจินตทัศน์จึงสามารถทำได้ด้วยการเขียนโปรแกรมภาษา C++ สำหรับงานที่มีการคำนวณและต้องการความเร็ว และด้วยภาษา Tcl ซึ่งเป็นภาษาที่มีลักษณะการทำงานเป็นแบบ interpreter เพื่อให้การพัฒนางานจินตทัศน์กระทำได้เสร็จรวดเร็ว
2. VTK ใช้แบบจำลองทางกราฟิกส์ในระดับสูงกว่าคลังคำสั่งทางกราฟิกส์ทั่วๆไปอาทิเช่น OpenGL หรือ PEX นั่นหมายความว่าการพัฒนางานทางคอมพิวเตอร์กราฟิกส์และงานจินตทัศน์จะกระทำได้ง่ายกว่ามาก
3. VTK ใช้ได้กับหลากหลายเครื่องคอมพิวเตอร์ อาทิเช่น Silicon Graphics, HP, Sun, Linux, PC (Win95 และ Windows NT)
4. VTK ถูกออกแบบโดยใช้แนวคิดเชิงวัตถุ จึงทำให้การเพิ่มเติมความสามารถเสริมต่างๆ กระทำได้อย่างมีระบบ
5. VTK เป็นซอฟต์แวร์สาธารณะที่ใช้แก้ไข เพิ่มเติม ได้โดยไม่เสียค่าใช้จ่ายใดๆ มีเอกสารสนับสนุน มากมาย พร้อมทั้งโปรแกรมต้นฉบับแสนกว่าบรรทัด

ต้นแบบทางกราฟิกส์ของ VTK มีลักษณะที่เลียนการสร้างภาพยนตร์ ซึ่งประกอบด้วยวัตถุพื้นฐาน 6 ประเภทดังนี้

1. Render Master มีหน้าที่ประสานวิธีทางกราฟิกส์ที่ไม่ขึ้นกับอุปกรณ์ และสร้างวินโดว์การแสดงภาพ (rendering window)
2. Render Window มีหน้าที่จัดการวินโดว์บนจอภาพ โดยมีตัวแสดงภาพ (renderer) หนึ่งหรือมากกว่าหนึ่งตัวคอยแสดงภาพในวินโดว์

---

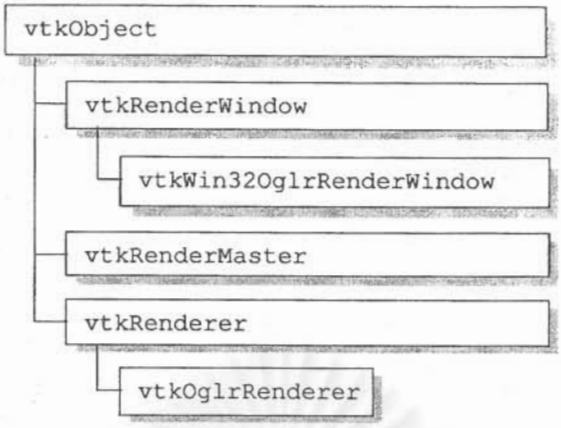
[2] J.K. Ousterhout, *Tcl and the Tk Toolkit*. Addison-Wesley Publishing Company, Reading, M.A. 1984.

3. Renderer มีหน้าที่ประสานการแสดงภาพของแสง (light) กล้อง (camera) และตัวแสดง (actor)
4. Light มีหน้าที่ให้แสงกับตัวแสดงในฉาก
5. Camera เป็นตัวตั้งตำแหน่งมุมมอง ระยะโฟกัส และลักษณะสมบัติอื่นๆของกล้อง
6. Actor หรือตัวแสดง เป็นวัตถุที่ถูกวาดโดยตัวแสดงภาพในฉาก ตัวแสดงนี้ถูกนิยามจาก
    - perperty ที่แทนลักษณะสมบัติของตัวแสดง เช่นสี ภาพพื้นผิววัตถุ ลักษณะการให้เงา ลักษณะการวาด (แบบ wireframe หรือ shaded)
    - mapper ที่แทนนิยามรูปทรงเรขาคณิตของตัวแสดง
    - transform ที่เป็นเมตริกซ์ขนาด 4x4 เพื่อใช้ในการเปลี่ยนแปลงลักษณะการวาดของตัวแสดง

หากจะพิจารณาเฉพาะส่วนการแสดงภาพเท่านั้น จะพบว่า VTK มีคลาสของวัตถุให้ใช้ได้ตามผังลำดับชั้นคลาสที่แสดงในรูปที่ 3-1 ดังมีรายละเอียดของแต่ละคลาสดังนี้

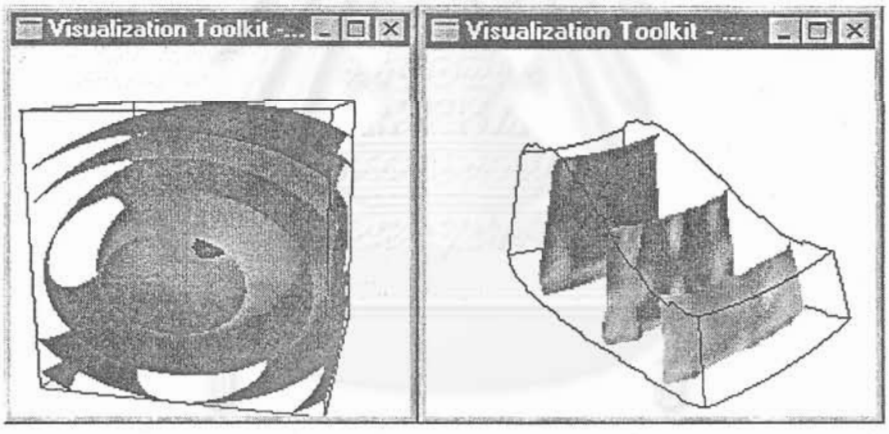
- `vtkObject` เป็นคลาสพื้นฐานใน VTK คลาสนี้มีวิธีการเพื่อการติดตามการเปลี่ยนแปลงเพื่อการหาข้อผิดพลาดของการทำงาน (debug) และการพิมพ์ ดังนั้นวัตถุส่วนใหญ่ใน VTK จะเป็นคลาสย่อยของ `vtkObject`
- `vtkRenderMaster` เป็นคลาสของวัตถุที่ถูกใช้เพื่อสร้างวินโดว์แสดงภาพ ซึ่งมีหน้าที่ในการเชื่อมโยงการทำงานกับระบบปฏิบัติการ เพื่อหาว่าจะใช้คลังคำสั่งชุดใดในการแสดงภาพ
- `vtkRenderWindow` เป็นคลาสสำหรับสร้างวัตถุเพื่อกำหนดพฤติกรรมของวินโดว์เพื่อการแสดงภาพ วิธีการต่างๆของคลาสนี้มีไว้ให้เพื่อประสานขบวนการการแสดงภาพ การกำหนดขนาดของวินโดว์ และการควบคุมการจัดการบัฟเฟอร์คู่
- `vtkRenderer` เป็นคลาสที่มีไว้เพื่อระบุข้อกำหนดต่างๆของตัวแสดงภาพ ซึ่งมีหน้าที่ในการแปลงรูปทรงเรขาคณิต แสง และมุมกล้องไปเป็นภาพในวินโดว์ นอกจากนี้ยังรับผิดชอบในการแปลงระบบพิกัดต่างๆระหว่างพิกัดโลก พิกัดมุมมอง และพิกัดของจอภาพอีกด้วย
- `vtkWin32OlgrRenderWindow` เป็นคลาสย่อยของ `vtkRenderWindow` ที่ทำหน้าที่วินโดว์แสดงภาพผ่านคลังคำสั่ง OpenGL ภายใต้ระบบปฏิบัติการ Windows แบบ 32 บิต (หาก VTK ถูกติดตั้งลงในระบบปฏิบัติอื่น ก็ต้องใช้คลังคำสั่งที่มีในระบบปฏิบัติการนั้น สำหรับในโครงการสิ่งประดิษฐ์นี้ ผู้วิจัยเลือกใช้ VTK ที่ทำงานในระบบปฏิบัติการ Windows 95 หรือ Windows NT ที่ในชุดคลังคำสั่งทางด้านกราฟิกส์ของ OpenGL)
- `vtkOglrRenderer` เป็นคลาสย่อยของ `vtkRenderer` ที่ทำหน้าที่ตัวแสดงภาพโดยใช้คลังคำสั่งของ OpenGL

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3-1 ลำดับชั้นของคลาสที่เกี่ยวกับการแสดงภาพ [3]

รูปที่ 3-2 แสดงตัวอย่างผลลัพธ์ที่ได้จาก VTK รูปทางขวาแสดงผลของการพล็อต isosurface ของฟังก์ชัน $x^2+2y^2+3z^2+yz=c$ และรูปทางขวาแสดงระนาบ x-y ที่บิดเบี้ยวตามค่า z ของฟังก์ชัน $F(x,y)=z$ โดยมีการให้สีตามค่าลักษณะสมบัติของจุดต่างๆบนระนาบ



รูปที่ 3-2 ตัวอย่างวินโดว์แสดงภาพของ VTK

รูปที่ 3-3 แสดงตัวอย่างโปรแกรมภาษา C++ และ Tcl ที่สร้างรูปลูกบาศก์หนึ่งลูก โปรแกรมทั้งสองเริ่มต้นด้วยการใช้วัตถุ Render Master (ตัวแปร `rm`) เพื่อสร้าง Render Window ขึ้นมาหนึ่งวินโดว์ (ตัวแปร `renWin`) ด้วยวิธี `MakeRenderWindow` สร้างตัวแสดงภาพ Renderer (ตัวแปร `ren1`) ให้กับวินโดว์ ด้วยวิธี `MakeRenderer` แล้วเพิ่มตัวแสดง (ตัวแปร `cubeAct`) ลงไปในตัวแสดงภาพ ด้วยวิธี `AddActor` แต่ก่อนจะเพิ่มนั้นต้องกำหนดลักษณะสมบัติให้กับตัวแสดง `cubeAct` ก่อนว่าเกิดจากรูปทรงเรขาคณิตแบบลูกบาศก์ด้วยวิธี `SetMapper` โดยการเรียกวิธี `GetOutput` จากตัวผลิตลูกบาศก์ ส่ง

---

[3] แผนผังแสดงต้นไม้การสืบทอดคลาสนี้ มีคลาสฐานอยู่ด้านบนสุด โดยการสืบทอดจะแสดงจากบนลงล่างและเยื้องไปด้านขวา

ผลไปให้ mapper ลูกบาศก์ด้วยวิธี SetInput เมื่อเพิ่มตัวแสดง CubeAct เสร็จแล้วก็ให้วินโดว์แสดงภาพแสดงผลลัพธ์

นอกจากนี้เราสามารถเพิ่มตัวโต้ตอบให้กับวินโดว์แสดงภาพได้ด้วย เพื่อให้ผู้ใช้สามารถขยาย เลื่อน หมุนกล้อง หรือปรับลักษณะสมบัติอื่นๆ ที่ส่องดูฉากผ่านวินโดว์นี้ได้ โดยการกดปุ่มเปลี่ยนแปลงต่างๆทางแป้นพิมพ์ ตัวโต้ตอบนี้เรียกว่า Interactor รูปที่ 3-4 แสดงตัวอย่างการเพิ่มตัวโต้ตอบเข้าในโปรแกรมในรูปที่ 3-3 (ขอแสดงเฉพาะโปรแกรมภาษา Tcl เท่านั้น) สำหรับปุ่มคำสั่งต่างๆที่ผู้ใช้กดได้แสดงไว้ในตารางที่ 3-1

```
// C++ code to draw a cube
#include  "RenderM.hh"
#include  "CubeSrc.hh"
#include  "PolyMap.hh"

main()
{
  vtkRenderMaster    rm;
  vtkCubeSource      cubeSrc;
  vtkPolyMapper      cubeMpr;
  vtkRenderWindow   *renWin;
  vtkRenderer       *ren1;
  vtkActor          *cubeAct;

  renWin = rm.MakeRenderWindow();
  ren1 = renWin->MakeRenderer();

  cubeMpr = SetInput(
            cubeSrc->GetOutput) );
  cubeAct = new vtkActor;
  cubeAct->SetMapper( cubeMpr );
  ren1->AddActors( cubeAct );

  renWin->Render();
}
```

```
# Tcl code to draw a cube

vtkRenderMaster  rm;
vtkCubeSource    cubeSrc;
vtkPolyMapper    cubeMpr;
vtkActor         cubeAct;

set renWin [rm MakeRenderWindow];
set ren1 [$renWin MakeRenderer];
cubeMpr SetInput \
        [cubeSrc GetOutput];
cubeAct SetMapper cubeMpr;
$ren1 AddActors cubeAct;

$renWin Render;
```

รูปที่ 3-3 ตัวอย่างโปรแกรม C++ และ Tcl ที่แสดงลูกบาศก์หนึ่งลูก

```
# Tcl code to draw a cube with interactor
source vtkInt.tcl

vtkRenderMaster  rm;
vtkCubeSource    cubeSrc;
vtkPolyMapper    cubeMpr;
vtkActor         cubeAct;

set renWin [rm MakeRenderWindow];
set ren1 [$renWin MakeRenderer];
set iren [$renWin MakeRenderWindowInteractor];

cubeMpr SetInput [cubeSrc GetOutput];
cubeAct SetMapper cubeMpr;
$ren1 AddActors cubeAct;
$iren SetUserMethod {wm deiconify .vtkInteract};
$iren Initialize;
$renWin Render;

wm withdraw .
```

รูปที่ 3-4 ตัวโปรแกรมภาษา Tcl ที่แสดงลูกบาศก์หนึ่งลูกพร้อมกับตัวโต้ตอบ

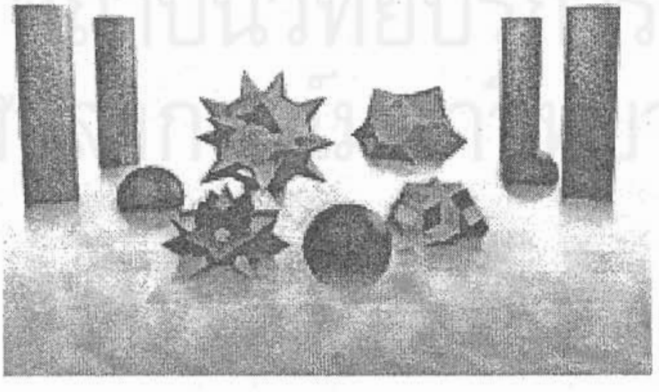
ตารางที่ 3-1 ปุ่มคำสั่งต่างๆของตัวโต้ตอบ

ปุ่มคำสั่งของตัวโต้ตอบ	ความหมาย
กดปุ่มซ้ายของเมาส์	หมุน
กดปุ่มซ้ายของเมาส์ พร้อมกดปุ่ม SHIFT	เลื่อน
กดปุ่มขวาของเมาส์	ขยาย
ตัวอักษร 'e'	เลิกการทำงาน
ตัวอักษร 'r'	ตั้งกล้องกลับสู่สภาพแรกสุด
ตัวอักษร 'w'	แสดงภาพวัตถุแบบกรอบเส้น (wireframe)
ตัวอักษร 's'	แสดงภาพวัตถุแบบมีพื้นผิว
ตัวอักษร 'u'	สำหรับให้ผู้ใช้ตั้งเองได้
ตัวอักษร '3'	แสดงภาพแบบสามมิติแดงน้ำเงิน

สำหรับรายละเอียดของ VTK สามารถหาอ่านเพิ่มเติมได้ในหนังสือ The Visualization ToolKit : An Object-Oriented Approach to 3D Graphics ของ W. Schroeder, K. Martin, และ B. Lorensen จัดพิมพ์และจำหน่ายโดย Prentice Hall ISBN: 0-13-199837-4

3.2 3D-Studio MAX

3D-Studio Max เป็นซอฟต์แวร์สร้างแบบจำลองสามมิติของบริษัทออโตเดกส์ คุณสมบัติของซอฟต์แวร์นี้ประกอบด้วยเครื่องมือการสร้างต้นแบบจำลองสามมิติ รวมทั้งการสร้างภาพเหมือนจริง และภาพเคลื่อนไหวจากแบบจำลองที่สร้างขึ้น (ดังตัวอย่างที่แสดงในรูปที่ 3-5) 3D-Studio Max ใช้หลักการออกแบบเชิงวัตถุ ซึ่งอนุญาตให้เพิ่มเติมความสามารถของการสร้างภาพได้ด้วยการพัฒนาส่วนเสริม plug-in ให้กับ 3D-Studio MAX



รูปที่ 3-5 ตัวอย่างแบบจำลองสามมิติจาก 3D-Studio MAX

วัตถุพื้นฐานที่ใช้ใน 3D-Studio Max แบ่งออกได้เป็นหลายๆประเภทดังนี้

1. Procedurals เป็นวัตถุที่ผู้พัฒนานิยามขึ้นแบ่งออกเป็นประเภทย่อยๆ ดังนี้

   - Geometrics เป็นวัตถุรูปทรงเรขาคณิตต่างๆ ซึ่งเป็นวัตถุประเภทเดียวเท่านั้นที่ถูกแสดงได้แก่ แบบพื้นฐาน (primitive) เช่นกล่อง ทรงกลม กรวย ทรงกระบอก หลอด เป็นต้น แบบอนุภาค (particle) เพื่อการสร้างสภาพเสมือน เช่นฝน หิมะ ฝุ่น เป็นต้น แบบแผ่นตะแกรง (patch) แบบสร้างโครง (loft) และแบบประกอบ (compound) ซึ่งรวมวัตถุหลายๆประกอบเข้าด้วยกันด้วยวิธีบูลลีน หรือวิธีการกลายรูป (morph)

   - Shapes เป็นวัตถุสองมิติ เช่นวงกลม สี่เหลี่ยม เส้น เป็นต้น เส้นสองมิติเหล่านี้มีไว้ใช้ในการสร้างวัตถุแบบสร้างโครง (loft) โดยการกำหนดภาคตัดขวางและทางเดินของภาคตัดขวางเพื่อสร้างโครงของวัตถุ

   - Lights เป็นวัตถุที่สร้างแสงสว่างให้กับสภาพแวดล้อมของแบบจำลอง

   - Cameras เป็นวัตถุที่กำหนดมุมมองของสภาพแวดล้อม และคุณสมบัติต่างๆของกล้อง

   - Helpers เป็นวัตถุช่วยเสริมในการวางตำแหน่ง การวัด และการภาพเคลื่อนไหวระหว่างการสร้างแบบจำลอง

2. Space warps เป็นวัตถุประเภทที่ช่วยสร้างสภาพบิดเบี้ยวให้กับสภาพแวดล้อมแบบจำลอง เช่นการสร้างคลื่นให้กับพื้น การสร้างสภาพที่มีลมให้กับวัตถุแบบอนุภาค เป็นต้น

3. Modifiers เป็นวัตถุที่ถูกใช้เพื่อเปลี่ยนแปลงวัตถุอื่นๆ เช่นการงอ การทำให้เรียว หรือการบิด เป็นต้น

4. Controllers เป็นวัตถุที่ควบคุมการเคลื่อนไหว เช่นการกำหนดตำแหน่ง การหมุน การย่อขยาย การกำหนดภาพสำคัญ การแทรกภาพระหว่างการเคลื่อนไหว เป็นต้น

5. Systems เป็นวัตถุที่รวมวัตถุแบบ procedural เข้ากับวัตถุแบบ controller แบบ modifier หรือแบบ space warp เข้าทำงานร่วมกัน เพื่อสร้างระบบแบบจำลองที่มีความซับซ้อนสูง

นอกจากวัตถุพื้นฐานที่กล่าวถึงข้างบนนี้ ยังมีส่วนเสริม Plug-in อีกหลายประเภทดังนี้

1. Atmospheric เป็น plug-in เพื่อสร้างสภาพแวดล้อมด้วยเทคนิคพิเศษเพื่อให้เหมือนจริง เช่น การเติมหมอก ควัน เป็นต้น

2. Materials / Textures เป็น plug-in ที่สร้างความเหมือนจริงทางด้านวัสดุ และพื้นผิว ของวัตถุต่างๆ

3. File import / export เป็น plug-in เพื่อเสริมความสามารถในการอ่านและบันทึกแฟ้มข้อมูลในรูปแบบต่างๆ

4. Image processing เป็น plug-in เพื่อการประมวลผลภาพเพื่อตบแต่ง ปรับปรุง หรือเปลี่ยนแปลง หลังจากได้สร้างภาพเคลื่อนไหวจากแบบจำลองแล้ว (video post) เช่นการซ้อนภาพ การใส่หัวเรื่อง หรือคำบรรยาย การสร้างภาพเลือนหาย เป็นต้น โดยสามารถทำงานกับภาพจากแฟ้มหนึ่งแฟ้ม หรือเป็นการรวมภาพจากสองแฟ้มเข้าเป็นหนึ่งแฟ้มก็ได้

สำหรับผู้พัฒนา plug-in เพิ่มเติมให้กับ 3D-Studio Max นั้น สามารถใช้วัตถุในคลาส ต่างๆเพื่อการพัฒนาได้ดังนี้

1. Main plug-in class เป็นคลาสหลักเพื่อใช้สำหรับการสร้าง plug-in ประเภทต่างๆ ตัวอย่าง เช่นถ้าอยากจะสร้าง plug-in ที่ทำการประมวลผลภาพ video post ก็ต้องสร้างจาก `class ImageFilter` หรือถ้าต้องการสร้าง plug-in เพื่อแสดงสภาพบรรยายกาศแบบใหม่ ก็ต้องสร้างจาก `class Atmospheric` เป็นต้น

2. Geometric / Bitmap class เป็นคลาสที่มีไว้สร้างและจัดการรูปทรงเรขาคณิต และภาพบิตแมพต่างๆ

3. Interface class เป็นคลาสที่มีไว้ใช้เพื่อติดต่อกับเข้าไปในส่วนกลางของ 3D-Studio Max และยังสามารถใช้เพื่อติดต่อไปยัง plug-in มาตรฐานอื่นๆได้ด้วย

4. User Interface class เป็นคลาสที่มีไว้ประสานการติดต่อกับผู้ใช้ด้วยวิธีการแบบกราฟิกส์ทั่วไป เช่นการป้อนค่าพารามิเตอร์ผ่านทางแป้นพิมพ์ ผ่านทางการกดปุ่มควบคุมต่างๆ และยังสามารถใช้เพื่อเข้าถึงวินโดว์ต่างๆที่แสดงบนจอภาพได้อีกด้วย

5. Utility class เป็นคลาสของวัตถุที่มีไว้ใช้งานจิปาถะ เพื่อการประมวลผลที่สะดวกขึ้น เช่น การกำหนดหมายเลขคลาส การแสดงแถบเลือกสีและจานสี การจัดการสายอักษร เป็นต้น

รูปที่ 3-6 แสดงตัวอย่างคลาสสำคัญต่างๆของชุดพัฒนา 3D-Studio MAX ที่ใช้ในการพัฒนาส่วนเสริม (คลาสต่างๆที่แสดงนี้เป็นที่สืบทอดจาก class Animatable ที่เป็นคลาสของวัตถุที่ถูกสร้างเป็นส่วนหนึ่งของแบบจำลองที่สร้างภาพเคลื่อนไหวได้ [4])

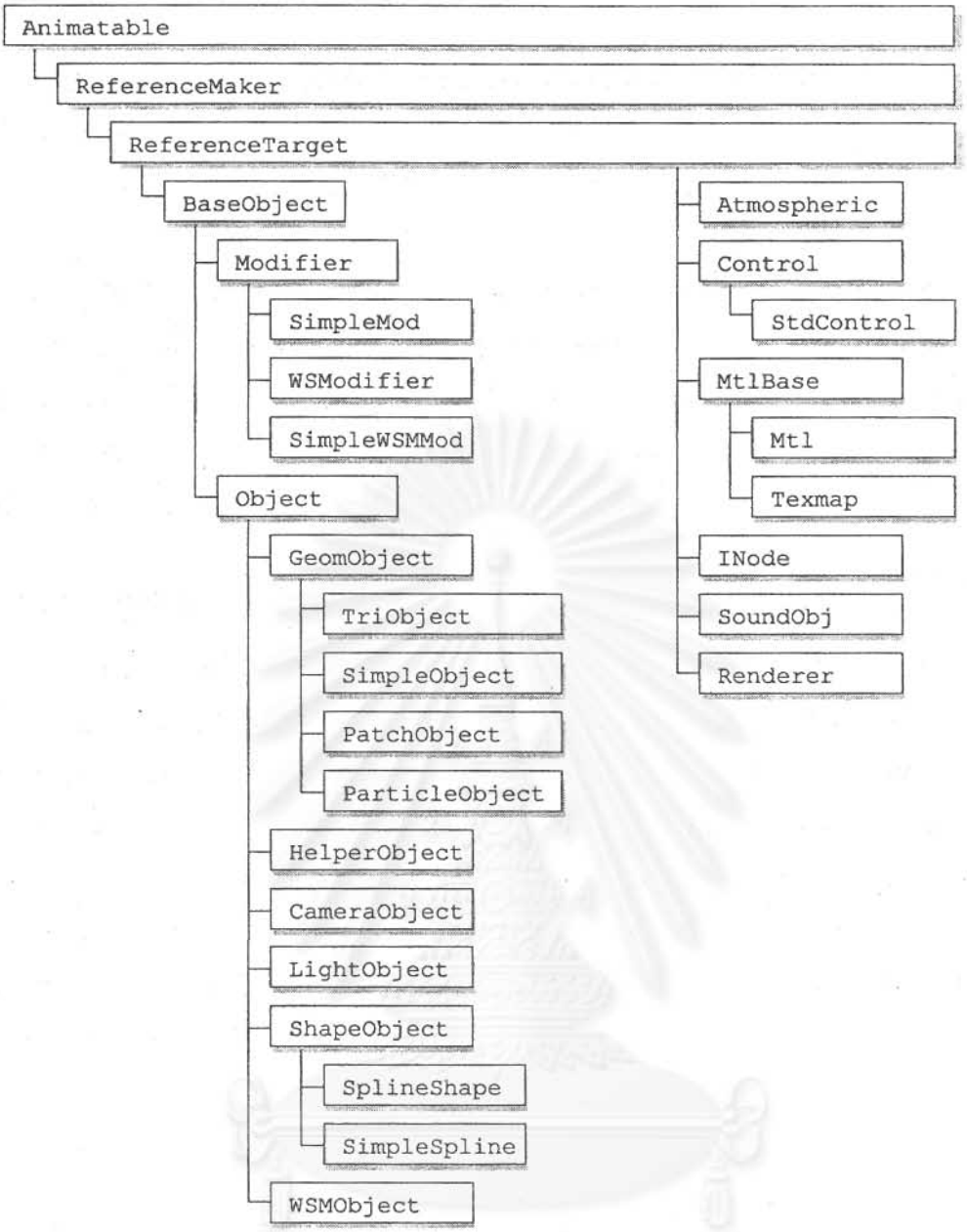
---

[4] รายละเอียดและคำอธิบายวิธีการใช้คลาสต่างๆของ 3D-Studio MAX จะขอไม่กล่าวในรายงานนี้ ผู้สนใจสามารถหารายละเอียดดังกล่าวได้ในชุดพัฒนาซอฟต์แวร์สำหรับ 3D-Studio MAX

ส่วนเสริม plug-in ของ 3D-Studio MAX ในทางปฏิบัติแล้วก็คือคลังคำสั่งประเภท DLL (Dynamic Linking Library) ที่มีปฏิบัติตามมาตรฐานที่ 3D-Studio MAX กำหนดไว้ โดยจะต้องมีฟังก์ชันหลัก 5 ฟังก์ชันดังต่อไปนี้

1. `DLLMain` เป็นฟังก์ชันที่ถูกเรียกเมื่อ plug-in นี้ถูกนำเข้าสู่ระบบ ภาระที่ต้องทำในฟังก์ชันนี้ก็คือการเตรียมตัวเช่นการจอง การตั้งค่าเริ่มต้น ของที่เก็บข้อมูลและตัวควบคุมต่างๆที่จำเป็นก่อนเริ่มทำงานจริง
2. `LibNumClasses` เป็นฟังก์ชันที่คืนค่าจำนวนคลาสของ plug-in
3. `LibClassDesc` เป็นฟังก์ชันที่คืนตำแหน่งของข้อมูลสำคัญของคลาสต่างๆของ plug-in
4. `LibDescription` เป็นฟังก์ชันที่คืนข้อความแสดงให้ผู้ใช้ทราบเกี่ยวกับ plug-in
5. `LibVersion` เป็นฟังก์ชันที่คืนค่าหมายเลขรุ่นของ 3D Studio MAX ที่ใช้ได้กับ plug-in

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3-6 แผนผังแสดงต้นไม้การสืบทอดคลาสสำคัญ (ที่เกี่ยวกับการสร้างภาพเคลื่อนไหว) ในชุดพัฒนาของ 3D-Studio MAX

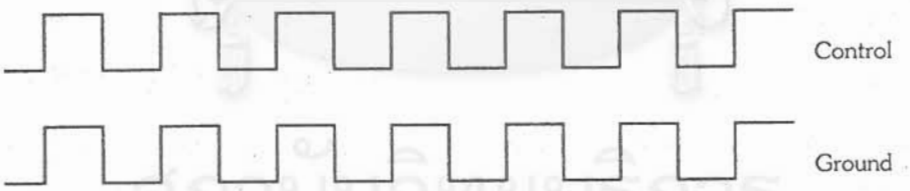
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4. ฮาร์ดแวร์สำหรับการแสดงภาพสามมิติสเตอริโอสลับเชิงเวลา

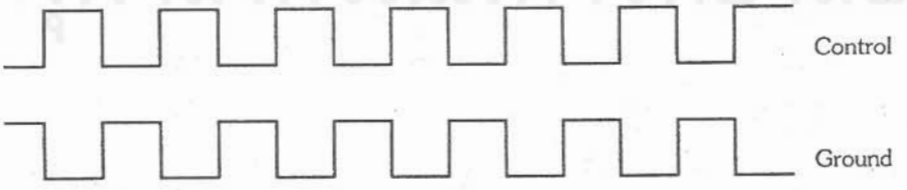
ในบทนี้จะได้กล่าวถึงการทำงานของอุปกรณ์และวงจรควบคุมต่างๆ ที่ใช้ประกอบการมองภาพสามมิติที่แสดงบนจอภาพ เพื่อรับรู้ความลึกได้ อุปกรณ์ที่ใช้คือแว่นตาผลึกเหลวที่มีแผ่นพลาสติกบังตาบรรจุสารผลึกเหลวไว้ซึ่งเราสามารถส่งสัญญาณไฟฟ้าเพื่อบังคับให้สารนี้มีสภาพทึบหรือโปร่งแสงได้ ส่วนวงจรการควบคุมแว่นผลึกเหลวนั้นจะมีสองประเภทคือ วงจรเพื่อการแสดงภาพสามมิติสเตอริโอแบบพลิกหน้า และวงจรเพื่อการแสดงภาพสามมิติสเตอริโอแบบสอดสลับ

4.1 แว่นตาผลึกเหลว

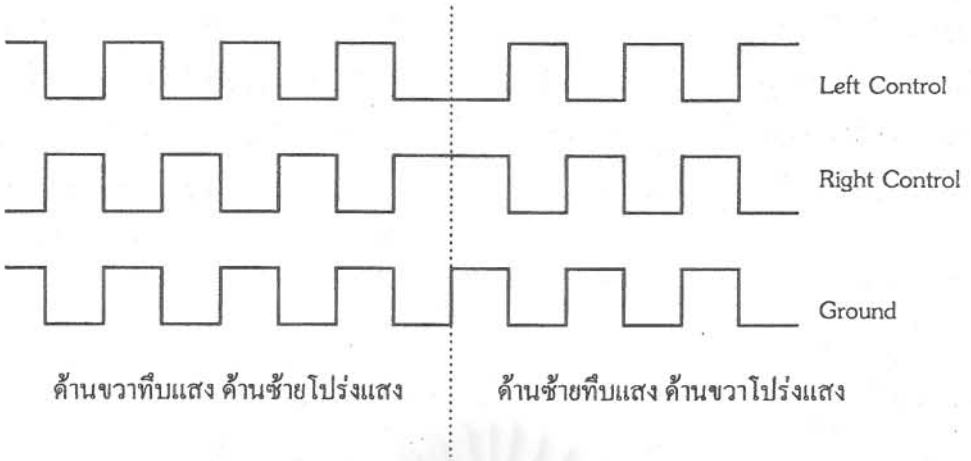
แว่นผลึกเหลวมีแผ่นบังตาแบบผลึกเหลวสองข้าง เมื่อแผ่นผลึกเหลวได้รับสัญญาณที่เหมาะสมจะทำให้ผลึกเหลวโปร่งแสงหรือทึบแสงตามแต่ลักษณะสัญญาณที่ป้อนให้ ตัวแว่นผลึกเหลวนี้จะรับสัญญาณ 3 เส้น คือสัญญาณร่วม Ground, สัญญาณควบคุมแผ่นผลึกเหลวข้างซ้าย และสัญญาณควบคุมแผ่นผลึกเหลวข้างขวา โดยทั้ง 3 สัญญาณจะเป็นคลื่นรูปสี่เหลี่ยมแรงดันประมาณ 10 โวลท์ ความถี่ประมาณ 1.6KHz แผ่นผลึกเหลวจะโปร่งแสงถ้าคลื่นสี่เหลี่ยมของสัญญาณร่วมและของสัญญาณควบคุมมีเฟสเหมือนกัน (รูปที่ 4-1) และจะทึบแสงเมื่อต่างเฟสกัน 180 องศา (รูปที่ 4-2) เมื่อปรับเฟสของคลื่นสี่เหลี่ยมที่ควบคุมแผ่นผลึกเหลวให้เหมาะสม ก็สามารถสลับการปิดเปิดแผ่นผลึกเหลวที่บังตาได้ตรงตามภาพที่แสดง ดังตัวอย่างในรูปที่ 4-3



รูปที่ 4-1 แผ่นผลึกเหลวโปร่งแสง เนื่องจากสัญญาณที่ป้อนให้มีเฟสเหมือนกัน



รูปที่ 4-2 แผ่นผลึกเหลวทึบแสง เนื่องจากสัญญาณที่ป้อนให้มีเฟสต่างกัน 180 องศา



รูปที่ 4-3 ตัวอย่างการเปลี่ยนเฟสของสัญญาณควบคุมผลึกเหลว

4.2 วงจรควบคุมแว่นเปิดปิดผลึกเหลว

วงจรควบคุมแว่นเปิดปิดผลึกเหลวมีหน้าที่ในการรับสัญญาณคำสั่งจากเครื่องคอมพิวเตอร์ เพื่อปิดและเปิดแผ่นบังตาของแว่นผลึกเหลวให้คล้องจองกับภาพที่แสดงทางจอภาพ ยังผลให้ผู้สวมแว่นเห็นภาพคู่ของตาแต่ละข้างตามที่สร้างไว้

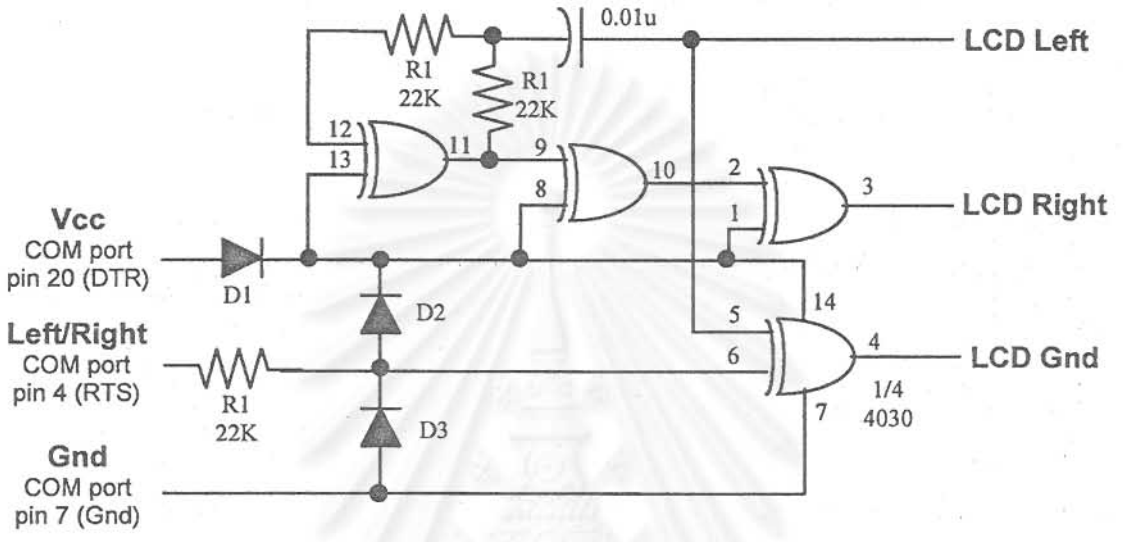
ในโครงการสิ่งประดิษฐ์นี้ใช้เทคนิคการแสดงภาพสามมิติสเตอริโอสองลักษณะด้วยกันคือ แบบ*พลิกหน้า* (page flipping) และแบบ*สอดสลับ* (interlacing) ในแบบพลิกหน้านั้นภาพสำหรับตาซ้ายและตาขวาจะถูกแสดงออกทางจอสลับกันไปเชิงเวลาโดยตรง โดยมีสัญญาณคำสั่งสลับภาพออกมาทางพอร์ตอนุกรมของระบบเพื่อการประสานจังหวะ ในขณะที่แบบสอดสลับนั้นภาพสำหรับตาซ้ายและตาขวาจะถูกผสานกันเป็นภาพเดี่ยว (ภาพซ้ายในเส้นคี่เรียกว่าภาพ*ฟิลด์ 1* และภาพขวาใช้เส้นคู่เรียกว่าภาพ*ฟิลด์ 2*) แล้วตั้งให้จอภาพทำงานแบบสอดสลับ เพื่อแสดงภาพฟิลด์ 1 สลับกับภาพฟิลด์ 2 ส่วนสัญญาณการควบคุมแว่นนั้นจะถอดรหัสมาจากสัญญาณ*รหัสเส้นขาว* (จะกล่าวรายละเอียดในหัวข้อถัดไป) เพื่อระบุจังหวะเวลาที่พร้อมสลับการเปิดปิดแผ่นผลึกเหลวบังตา

ด้วยลักษณะการทำงานของแว่นผลึกเหลวที่กล่าวในหัวข้อย่อยที่แล้ว และเทคนิคการแสดงภาพสามมิติสเตอริโอที่ใช้โครงการนี้ จึงมีลักษณะวงจรที่ใช้งานอยู่สองรูปแบบดังที่จะกล่าวรายละเอียดในหัวข้อย่อยต่อไปนี้

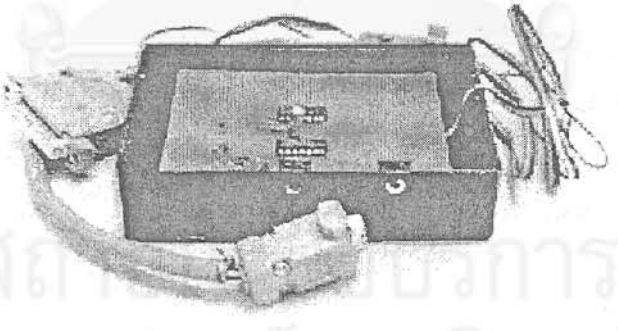
4.2.1 วงจรควบคุมแว่นเปิดปิดผลึกเหลวแบบพลิกหน้า

วงจรควบคุมแว่นเปิดปิดผลึกเหลวแบบพลิกหน้า เป็นวงจรที่ไม่ซับซ้อน (ดูรูปที่ 4-4 สำหรับต้นแบบวงจรที่สร้างขึ้นแสดงในรูปที่ 4-6) ซึ่งอาศัยพอร์ตอนุกรมเป็นทางออกของสัญญาณ จากเครื่องคอมพิวเตอร์ส่วนบุคคล ในการกำหนดการเปิดปิดแผ่นผลึกเหลว (ทั้งนี้เนื่องจากเครื่องคอมพิวเตอร์ส่วนบุคคลโดยทั่วไปมีพอร์ตอนุกรมสองพอร์ต ตัวเครื่องจึงมักมีพอร์ตเพียงพอเพื่อมาใช้ในการนี้) โดยใช้

สัญญาณขาที่ 4 (RTS) และ ขาที่ 20 (DTR) [1] เป็นสัญญาณควบคุม สัญญาณที่ขา DTR เป็นสัญญาณควบคุมการทำงานของแว่น เพื่อให้แว่นทำงานหรือไม่ ในกรณีที่ให้แว่นตาทำงาน จะมีแผ่นผลึกเหลวข้างใดข้างหนึ่งทึบแสง ทั้งนี้ขึ้นกับสัญญาณที่ขา RTS ตารางที่ 4-1 รูปที่ 4-5 และ รูปที่ 4-6 แสดงต้นแบบวงจรการควบคุมแว่นผลึกเหลวแบบพลิกหน้ารุ่นแรกและรุ่นที่สองที่ได้สร้างขึ้น โดยรุ่นที่สองนั้นมีขนาดของต้นแบบที่เล็กกะทัดรัดกว่ารุ่นแรกมาก



รูปที่ 4-4 วงจรควบคุมแว่นผลึกเหลวแบบพลิกหน้า [2]

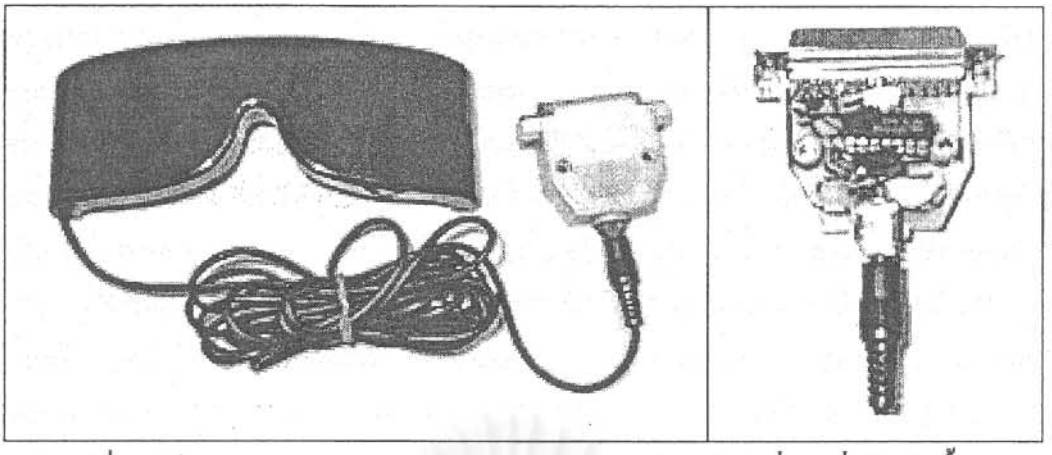


รูปที่ 4-5 ต้นแบบวงจรการควบคุมแว่นผลึกเหลวแบบพลิกหน้ารุ่นแรกที่ได้สร้างขึ้น

---

[1] RTS และ DTR เป็นชื่อของขาสัญญาณเมื่อใช้พอร์ตอนุกรมทางด้านการสื่อสารข้อมูล ในที่นี้เราใช้ชื่อดังกล่าวเพื่อการอ้างอิงเท่านั้น จึงไม่มีความสัมพันธ์ใดๆกับชื่อสัญญาณ นอกจากนี้ตำแหน่งขาสัญญาณจากพอร์ตอนุกรมที่ใช้นั้นเป็นของหัวต่อแบบ DB25

[2] D. Stampe, B Roehl, and J. Eagan, "Virtual Reality Creations,", The Waite Group, 1993. อนึ่งวงจรนี้ได้เผยแพร่ในเอกสาร วารสาร และหนังสือหลายเล่ม รวมทั้งในอินเตอร์เน็ตด้วย โดยไม่มีใครทราบที่มาว่าใครเป็นผู้ได้ออกแบบไว้



รูปที่ 4-6 ต้นแบบวงจรการควบคุมแว่นผลึกเหลวแบบพลิกหน้ารุ่นที่สองที่ได้สร้างขึ้น

ตารางที่ 4-1 สัญญาณควบคุมการทำงานของแว่นผลึกเหลว

DTR	RTS	แผ่นผลึกเหลวของตาซ้าย	แผ่นผลึกเหลวของตาขวา
-12V	ระดับใดๆ	โปร่งแสง	โปร่งแสง
+12V	+12V	ทึบแสง	โปร่งแสง
+12V	-12V	โปร่งแสง	ทึบแสง

วงจรในรูปที่ 4-4 นั้นประกอบด้วยวงจรสร้างสัญญาณคลื่นสี่เหลี่ยม (XOR สองตัวบนต่อกับวงจร RC) ความถี่ประมาณ 1.6KHz พร้อมกับสัญญาณความถี่เดียวกันที่กลับเฟสกัน (ใช้ XOR อีกหนึ่งตัวทำหน้าที่กลับเฟสสัญญาณ) เพื่อส่งไปเป็นสัญญาณควบคุมแผ่นผลึกเหลวของตาซ้ายและตาขวาตามลำดับ (แผ่นผลึกเหลวทั้งคู่จะรับสัญญาณที่กลับเฟสกัน เนื่องจากเมื่อข้างหนึ่งทึบ อีกข้างหนึ่งต้องโปร่ง) ส่วนสัญญาณที่ส่งให้กับขา ground ของแว่น เป็นสัญญาณคลื่นสี่เหลี่ยมความถี่เดียวกัน แต่เฟสของสัญญาณนี้จะตรงกับของสัญญาณควบคุมแผ่นผลึกเหลวข้างใดนั้น ขึ้นกับสัญญาณควบคุมเลือกจากพอร์ตอนุกรม (ขา 4) เพื่อให้ได้การทำงานตรงตามตารางที่ 4-1 ให้สังเกตว่าเนื่องจากเราเลือกใช้วงจรแบบ CMOS ซึ่งใช้พลังงานน้อยมากในการทำงาน ดังนั้นจึงสามารถใช้แรงดันไฟจากสัญญาณขาออกของพอร์ตอนุกรม ซึ่งคือสัญญาณ DTR (ขาที่ 20) ทำให้เมื่อสัญญาณ DTR = 12V วงจรจะทำงาน และเมื่อ DTR = -12V วงจรจะไม่ทำงาน ทำให้แผ่นผลึกเหลวโปร่งแสงทั้งสองข้าง

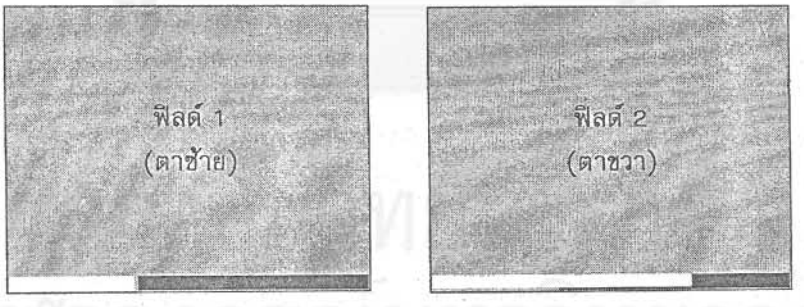
สำหรับโปรแกรมที่คอยส่งสัญญาณควบคุมนั้นจะมีลักษณะดังนี้ (สมมติว่าใช้พอร์ตอนุกรม COM2)

```
do {
   WaitForVerticalRetrace();
   outportb( COM2, DTR_ON_RTS_OFF );
   FlipPage( LEFT );

   WaitForVerticalRetrace();
   outportb( COM2, DTR_ON_RTS_ON  );
   FlipPage( RIGHT );

} while( NotYetStop() );
outportb( COM2, DTR_OFF );
```

ส่วนของโปรแกรมข้างบนนี้จะวนการทำงานโดยสลับการแสดงภาพสองภาพ โดยจังหวะที่สลับหน้าได้คือหลังจากได้รับสัญญาณ vertical retrace (รับทราบเหตุการณ์นี้ได้โดยการวน `inportb( 0x3DA )` เพื่อคอยตรวจสอบบิต b_3 เมื่อมีการเปลี่ยนจากหนึ่งเป็นศูนย์ ก็แสดงว่าเกิด vertical retrace) ก็จะส่งสัญญาณควบคุม DTR และ RTS ที่สั่งให้แผ่นผลึกเหลวซ้ายโปร่งขวาทึบ จากนั้นสั่งให้แผงวงจรแสดงภาพพลิกไปหน้าซ้าย แล้วรอสัญญาณ vertical retrace ลูกถัดไป เมื่อได้รับสัญญาณนี้ ก็จะส่งสัญญาณควบคุม DTR และ RTS ที่สั่งให้แผ่นผลึกเหลวซ้ายทึบขวาโปร่ง แล้วสั่งให้แผงวงจรแสดงภาพพลิกไปหน้าขวา กระทำเช่นนี้ไปเรื่อยๆจนกว่าจะหยุดการทำงาน ก็ส่งสัญญาณทาง DTR เพื่อให้แว่นหยุดการทำงาน (จากส่วนโปรแกรมข้างต้นนี้ DTR_ON_RTS_OFF คือค่า 3 , DTR_ON_RTS_ON คือค่า 1 และ DTR_OFF คือค่า 2 ส่วน COM2 คือพอร์ตหมายเลข 0x2FC)

4.2.2 วงจรควบคุมแว่นเปิดปิดผลึกเหลวแบบสอดสลับ

เนื่องจากการแสดงภาพสามมิติสเตอริโอแบบสอดสลับนั้น อาศัยการทำงานของจอภาพแบบสอดสลับ ซึ่งแสดงภาพฟิลด์ 1 และ 2 สลับกันไป และเพื่อให้แผ่นผลึกเหลวบังตาของแว่นที่ผู้มองสวมนั้นเปิดปิดให้ได้จังหวะที่ถูกต้องนั้น จะต้องสัญญาณควบคุมบางอย่างประกอบการสั่งการให้ถูกต้องด้วย ซึ่งสามารถทำได้โดยการใช้รหัสเส้นขาว (while line code) [3] ซึ่งเป็นเส้นสีขาวสองเส้นแสดงที่สองเส้นสุดท้ายของจอภาพ เส้นแรกเป็นเส้นสีขาวความยาวประมาณ 1/3 ของจอภาพ และอีกเส้นหนึ่งความยาวประมาณ 2/3 ของจอภาพ จะเห็นได้ว่าถ้าการแสดงผลเป็นแบบสอดสลับแล้ว เส้นขาวที่ยาว 1/3 ของจอภาพจะถูกแสดงเป็นเส้นสุดท้ายของฟิลด์ 1 และในทำนองเดียวกันเส้นขาวที่ยาว 2/3 ของจอภาพจะถูกแสดงเป็นเส้นสุดท้ายของฟิลด์ 2 ดังรูปที่ 4-7 สำหรับกรณีที่ไม่มีรหัสเส้นขาวปรากฏแผ่นผลึกเหลวของตาทั้งสองข้างจะโปร่งแสง ดังนั้นวงจรควบคุมแว่นเปิดปิดผลึกเหลวแบบสอดสลับจะต้องอาศัยรหัสเส้นขาวนี้ประกอบด้วย



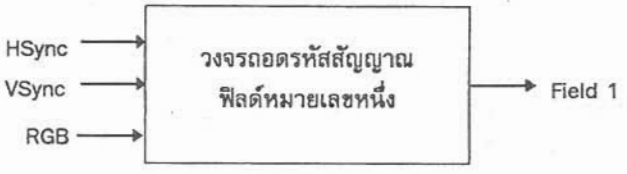
รูปที่ 4-7 รหัสเส้นขาวเป็นสัญญาณกำกับฟิลด์

จากรหัสเส้นขาวที่กล่าวมาแล้วนั้น จังหวะที่มีการเปลี่ยนแปลงการทึบหรือโปร่งแสงของแผ่นผลึกเหลวนั้น จะเกิดขึ้นหลังจากตรวจพบว่าเป็นเวลาที่จอภาพกำลังวาดเส้นขาวที่เส้นสุดท้ายของจอภาพ (นั่นหมายความว่ากำลังจะเกิดสัญญาณ vertical retrace แล้ว) ดังนั้น วงจรควบคุมแว่นผลึกเหลวแบบสอด

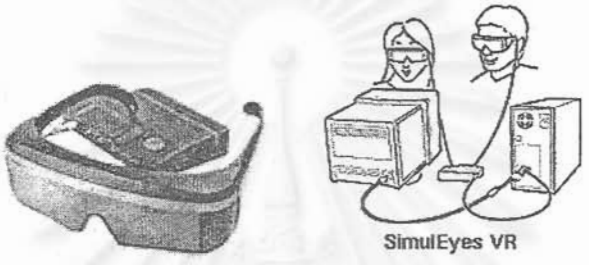
---

[3] รหัสเส้นขาวเป็นเทคนิคที่ใช้กับอุปกรณ์การแสดงและมองภาพสามมิติ SimulEyes® ของบริษัท StereoGraphics

สลับนี้ จะเป็นวงจรที่ถอดรหัสสัญญาณ VSync HSync และ RGB ที่เข้าสู่จอภาพ ออกเป็นสัญญาณระบุหมายเลขฟิลด์ของภาพที่แสดงออกจอภาพ เพื่อส่งไปควบคุมการเปิดปิดของแว่นผลึกเหลว ดังรูปที่ 4-8 สำหรับในโครงการสิ่งประดิษฐ์นี้ใช้ วงจรถอดรหัสสัญญาณฟิลด์ SimulEyes® ของบริษัท StereoGraphics[4] ซึ่งลักษณะการต่ออุปกรณ์และตัวอุปกรณ์ดังรูปที่ 4-9



รูปที่ 4-8 แผนผังแสดงสัญญาณขาเข้าและขาออกของวงจรถอดรหัสฟิลด์หมายเลขหนึ่ง



รูปที่ 4-9 แว่นผลึกเหลว อุปกรณ์ควบคุม และลักษณะการต่ออุปกรณ์ของ SimulEyes®

นอกจากวิธีการใช้เส้นรหัสสีขาวสองเส้นสุดท้ายเป็นตัวระบุหมายเลขฟิลด์ ยังสามารถใช้เทคนิคอื่นในการเข้าและถอดรหัสได้ ตัวอย่างเช่นมงคล ภิญโญสโมสร [5] ได้เสนอวิธีการใช้เส้นรหัสสีขาวเส้นสุดท้ายเพียงเส้นเดียวในฟิลด์คู่เป็นตัวกำหนดว่าฟิลด์ต่อไปจะเป็นฟิลด์คี่ โดยตัววงจรถอดรหัสจะนับจำนวนสัญญาณ HSync เพื่อตรวจจับเส้นสุดท้ายของฟิลด์ เพื่อให้วงจรถอดรหัสควบคุมแว่นได้ถูกต้อง โดยมีรายละเอียดของวงจรถอดรหัสเส้นขาวดังรูปที่ 4-10 [6] ในวงจรนี้จะมีการนับจำนวนสัญญาณ HSync (ตัวนับถูกตั้งค่าเป็นศูนย์เมื่อได้สัญญาณ VSync) เมื่อจำนวนลูกสัญญาณ HSync ระบุว่าเส้นที่จะวาดถัดไปคือเส้นสุดท้ายของภาพ (โดยการเปรียบเทียบจำนวนลูกสัญญาณ กับค่าคงที่ที่ตั้งไว้ล่วงหน้า ค่านี้แปรตามความละเอียดของจอภาพ) ทำให้สัญญาณ A=B ทำงาน ประจวบกับเมื่อสัญญาณของ RGB เป็นสีขาว (ซึ่งแรงดันขนาด 0.7V จะทำให้ทรานซิสเตอร์ทำงาน ยังผลให้สัญญาณขาออกของ AND เกต มีค่าเป็นหนึ่ง ซึ่งเป็นการตั้งค่าให้ฟลิบฟลอปมีค่าศูนย์ (ฟลิบฟลอปนี้ต่อเพื่อให้ทำงานเป็นแบบ toggle) เพื่อระบุว่าฟิลด์ถัดไปเป็นฟิลด์ 2 (เพื่อไปควบคุมแว่นผลึกเหลว) ให้สังเกตว่าถ้าถึงเส้นสุดท้ายของจอภาพ แต่ไม่มีเส้นสีขาว จะ

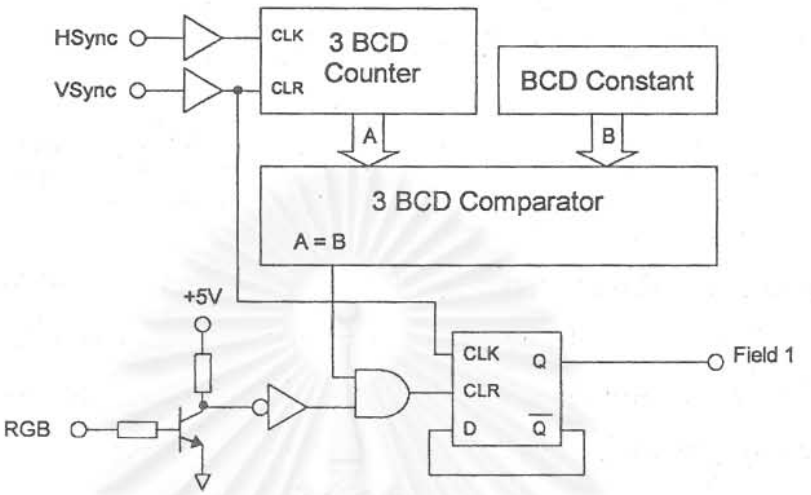
---

4 http://www.stereographics.com

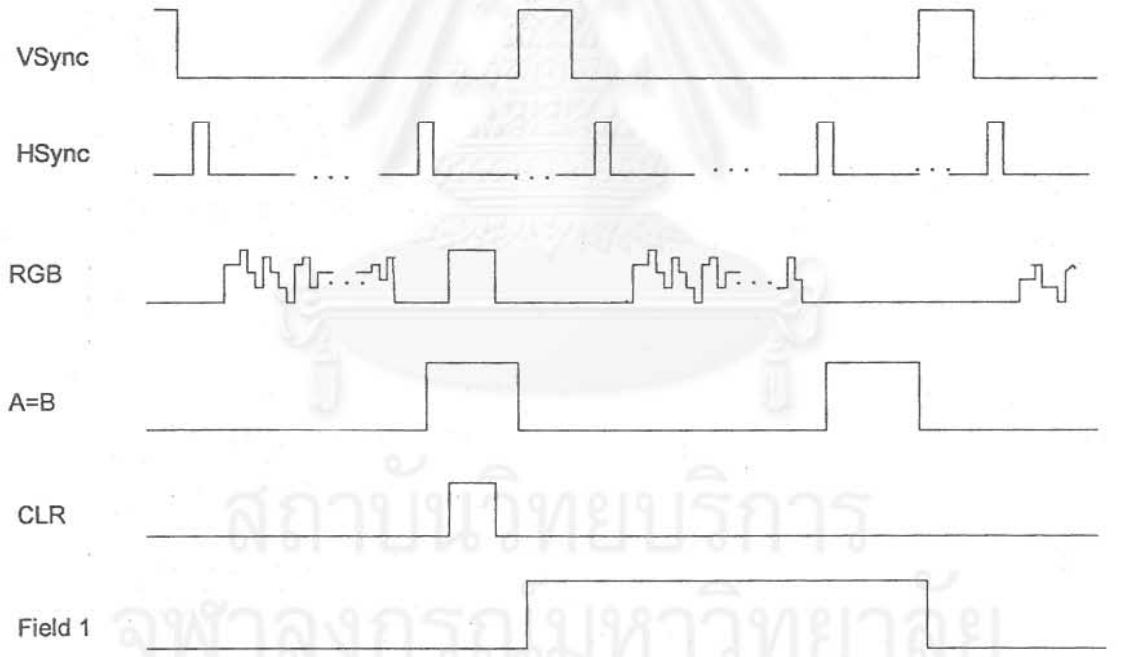
5 มงคล ภิญโญสโมสร, "การออกแบบและพัฒนาระบบการแสดงภาพสเตอริโอแบบแทรกสอด" วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต วิทยาศาสตร์คอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย 2539

6 รูปวงจรที่นำเสนอนี้ถูกดัดแปลงเล็กน้อยเพื่อให้ง่ายต่อการเข้าใจ

ไม่มีการสั่งงานที่สัญญาณ CLR ของฟลิปฟลอป ดังนั้นเมื่อถึง VSync ลูกถัดไป ฟลิปฟลอปจะเปลี่ยนค่ากลับกับค่าที่เป็นอยู่ รูปคลื่นสัญญาณต่างๆ ปรากฏในรูปที่ 4-11



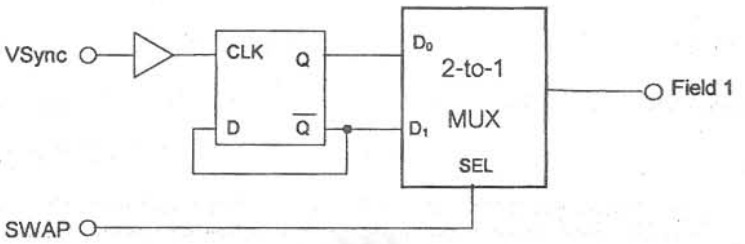
รูปที่ 4-10 แผนผังวงจรถอดรหัสฟิลด์ที่หนึ่ง



รูปที่ 4-11 รูปคลื่นของสัญญาณสำคัญต่างๆในวงจรรูปที่ 4-10

นอกจากวิธีการถอดรหัสฟิลด์ที่กล่าวมาสองวิธีข้างต้น เราสามารถลดความซับซ้อนของวงจรถอดรหัสฟิลด์ลงได้ให้เหลือเพียงเป็นวงจรหารความถี่ของสัญญาณ VSync ด้วยสอง เป็นสัญญาณระบุฟิลด์ แต่ด้วยวิธีนี้การสั่งการเปิดปิดแผ่นผลึกเหลวนั้นอาจทำงานสลับข้างได้ เมื่อเทียบกับฟิลด์ที่แสดงจริงบนจอภาพ (มีโอกาสผิดครึ่งต่อครึ่ง ซึ่งไม่เหมือนกับสองวิธีข้างต้นที่ใช้รหัสเส้นขาวเป็นสัญญาณช่วยกำหนด)

ดังนั้นจึงต้องให้ผู้มองเป็นผู้ช่วยกำหนด (การมองภาพจะผิดปกติมากหากตาทั้งสองมองภาพสลับกับที่ควรจะเป็น ซึ่งเป็นที่สังเกตได้ง่าย) นั่นคือมีการเพิ่มสวิตช์ให้ผู้มองปรับระดับสัญญาณการระบุหมายเลขฟิลด์ (สัญญาณ SWAP) ก็จะได้วงจรดังรูปที่ 4-12



รูปที่ 4-12 วงจรถอดรหัสฟิลด์ที่มีการชดเชยความผิดพลาด

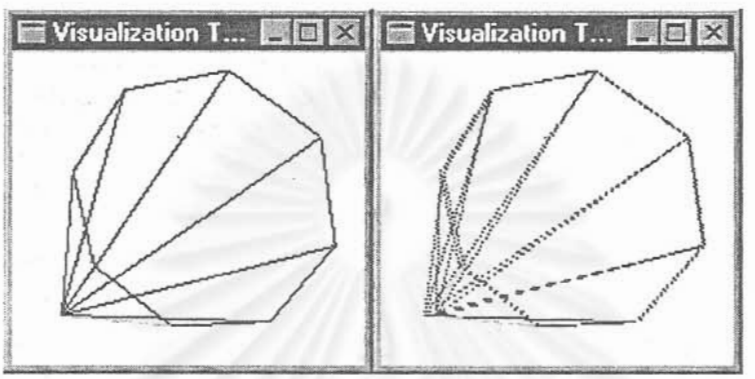
สำหรับกรณีที่เลือกใช้รหัสเส้นขาวเป็นตัวระบุฟิลด์ ผู้ใช้ต้องเรียกใช้โปรแกรมแสดงเส้นสีขาว โดยโปรแกรมนี้จะต้องแสดงเส้นสีขาวที่ต้องการบนจอภาพตลอดเวลาการมองภาพ โดยจะต้องเป็นโปรแกรมที่แสดงแบบ "always on top" นั่นคือไม่ว่าผู้ใช้จะย้ายวินโดว์ใดมาทับตำแหน่งเส้นสุดท้าย ก็ไม่สามารถทับได้ เนื่องจากเส้นรหัสนั้นจะ "อยู่บนตลอด" รูปที่ 4-13 แสดงโปรแกรมดังกล่าวที่เขียนด้วยภาษา Visual BASIC

```
Declare Function SetWindowPos Lib "user" (ByVal h%, ByVal hb%,
 ByVal x%, ByVal y%, ByVal cx%, ByVal cy%, ByVal f%) As Integer
' The above Declare statement must appear on one line.

Const SWP_NOMOVE = 2
Const SWP_NOSIZE = 1
Const FLAGS = SWP_NOMOVE Or SWP_NOSIZE
Const HWND_TOPMOST = -1
Const HWND_NOTOPMOST = -2

Sub Form_Load ()
  success% = SetWindowPos(WLC.hWnd, HWND_TOPMOST,0,0,0,0,FLAGS)

  Width = Screen.Width
  Height = 2 * Screen.TwipsPerPixelY
  Left = 0
  Top = Screen.Height - Height
  Show
  Form_Paint
End Sub

Sub Form_Paint ()
  ScaleMode = 3
  DrawWidth = 1
  Line (0, 0)-(ScaleWidth * .33, 0), RGB(255, 255, 255)
  Line (0, 1)-(ScaleWidth * .67, 1), RGB(255, 255, 255)
End Sub
```

รูปที่ 4-13 โปรแกรมแสดงเส้นรหัสขาวสองสุดท้ายบนจอภาพ

5. ส่วนเสริมแสดงภาพสามมิติแบบสอดสลับใน VTK

ในบทนี้จะได้กล่าวถึงหน้าที่ โครงสร้าง และการพัฒนาส่วนเสริมเพื่อการแสดงภาพสามมิติแบบสอดสลับใน Visualization Tool Kit รูปที่ 5-1 แสดงเส้นขอบของกรวยฐาน 8 เหลี่ยมรูปธรรมดา และรูปแบบสอดสลับภาพคู่แล้วแสดงโดย VTK ที่มีส่วนเสริม



รูปที่ 5-1 ภาพเปรียบเทียบรูปแบบปกติ และแบบสอดสลับ

5.1 หน้าที่ของส่วนเสริม

จุดประสงค์หลักของการพัฒนาส่วนเสริมใน VTK ก็เพื่อให้ระบบจินตทัศน์ VTK มีความสามารถในการแสดงภาพสามมิติแบบสอดสลับ โดยเพิ่มฟังก์ชันเลือก/ไม่เลือก คุณสมบัติการแสดงภาพนี้ เข้าใน VTK โดยจะต้องไม่กระทบถึงคุณสมบัติอื่นๆ ของ VTK หน้าที่ของส่วนเสริมที่พัฒนาขึ้นมีดังต่อไปนี้

1. เพิ่มประเภทการแสดงภาพสามมิติแบบสอดสลับ (VTK_STEREO_INTERLACE) [1] เพื่อให้ผู้พัฒนาสามารถระบุความต้องการแสดงภาพสามมิติแบบสอดสลับในโปรแกรมได้
2. เพิ่มคำสั่งให้กับตัวโต้ตอบ (interactor) โดยกำหนดให้เป็นแป้นพิมพ์หมายเลข 4 เป็นตัวสลับการเลือกไปมาระหว่างการแสดงภาพสามมิติแบบแดงน้ำเงิน กับแบบสอดสลับ [2]

รูปที่ 5-2 แสดงตัวอย่างโปรแกรม Tcl เพื่อแสดงรูปกรวยในรูปที่ 5-1 ที่เพิ่มคำสั่งกำหนดให้ตัวแสดงภาพแสดงแบบสามมิติสเตอริโอ และต้องการแบบสอดสลับด้วย (คำสั่งที่พิมพ์ด้วยตัวเอียง) และเนื่องจากโปรแกรมนี้มีการใช้ตัวโต้ตอบ ผู้ใช้สามารถกดแป้นพิมพ์หมายเลข 3 เพื่อสลับการเลือก/ไม่เลือก

---

[1] VTK เดิมนั้นมีประเภทการแสดงภาพสามมิติแบบแดงน้ำเงิน (VTK_STEREO_RED_BLUE) และแบบบนล่าง (VTK_STEREO_CRYSTAL_EYES) แต่เฉพาะแบบแดงน้ำเงินเท่านั้นที่ใช้งานได้

[2] นอกจากนี้การกดแป้นพิมพ์หมายเลข 3 จะเป็นการกำหนดว่าต้องการ หรือไม่ต้องการแสดงภาพแบบสามมิติสเตอริโออีกด้วย

I19947625

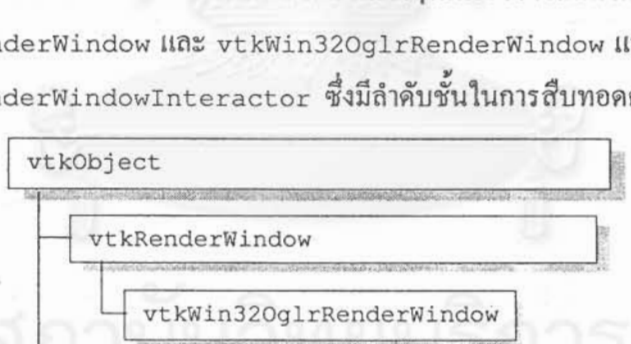
การแสดงภาพสามมิติสเตอริโอ และกดแป้นพิมพ์หมายเลข 4 เพื่อสลับการเลือกแบบสอดสลับหรือแบบแดงน้ำเงิน ในกรณีที่เลือกแสดงแบบสามมิติสเตอริโอ

```
# Tcl code to draw a cube with interactorvtk
source vtkInt.tcl

vtkRenderMaster  rm;
vtkConeSource    coneSrc;
vtkPolyMapper    coneMpr;
vtkActor         coneAct;

set renWin [rm MakeRenderWindow];
set ren1 [$renWin MakeRenderer];
set iren [$renWin MakeRenderWindowInteractor];
coneSrc SetResolution 8
coneMpr SetInput [coneSrc GetOutput];
coneAct SetMapper coneMpr;
$ren1 AddActors coneAct;
$ren1 SetBackground 1 1 1;
$iren SetUserMethod {wm deiconify .vtkInteract};

$iren Initialize;
$renWin StereoRenderOn;
$renWin SetStereoType VTK_STEREO_INTERLACE;
$renWin Render;
wm withdraw .
```

รูปที่ 5-2 โปรแกรมแสดงรูปลูกบาศก์หนึ่งลูกแบบสามมิติสอดสลับ

5.2 *โครงสร้างของส่วนเสริม*

ส่วนเสริมนี้จะประกอบด้วยคำสั่งและฟังก์ชันต่างๆที่เสริมเข้าในคลาสของตัวแสดงภาพ ซึ่งประกอบด้วย `vtkRenderWindow` และ `vtkWin32OglrRenderWindow` และคลาสของตัวโต้ตอบ ซึ่งคือ `vtkWin32RenderWindowInteractor` ซึ่งมีลำดับชั้นในการสืบทอดคลาสดังรูปที่ 5-3



รูปที่ 5-3 ลำดับชั้นของคลาสใน VTK ที่เกี่ยวข้องกับส่วนเสริมสามมิติแบบสอดสลับ

ฟังก์ชันสำคัญที่เกี่ยวข้องในคลาสเหล่านี้ที่ต้องได้รับการปรับปรุงและเพิ่มเติมมีดังแสดงในตารางที่ 5-1เมื่อมีการกำหนดให้แสดงภาพแบบสามมิติสเตอริโอ VTK จะเรียกแสดงภาพสองครั้ง เพื่อสร้างภาพสำหรับตาทั้งสอง โดยหลังจากสร้างภาพแรกเสร็จ จะมีการเรียกฟังก์ชัน `StereoMidpoint` หนึ่งครั้ง และเมื่อสร้างภาพที่สองเสร็จจะเรียกฟังก์ชัน `StereoRenderComplete` หนึ่งครั้ง และเมื่อต้องการ

แสดงภาพออกจอก็จะเรียกฟังก์ชัน CopyResultFrame สิ่งที่จะต้องจัดการคือ ภายในฟังก์ชัน StereoMidpoint จะบันทึกเฉพาะเส้นคี่ของภาพแรกที่สร้างได้ (ใช้ฟังก์ชัน GetOddRowPixel) ไว้ในเส้นคี่ของ StereoBuffer (เป็นที่เก็บรูปชั่วคราว) และในฟังก์ชัน StereoRenderComplete จะบันทึกเฉพาะเส้นคู่ของภาพที่สอง (ใช้ฟังก์ชัน GetEvenRowPixel) ไว้ในส่วนเส้นคู่ของ StereoBuffer จะได้เป็นภาพสองภาพซ้อนทับกัน ในลักษณะสามมิติแบบสอดสลับ และในฟังก์ชัน CopyResultFrame จะเรียกใช้ฟังก์ชัน DrawRGBAFrame เพื่อแสดงภาพที่เก็บใน ResultFrame ออกสู่จอภาพ สำหรับตัวโต้ตอบนั้น ก็เพียงแต่แทรกคำสั่งตั้งสภาวะของการแสดงให้ตรงกับที่ผู้ใช้กดแป้นพิมพ์ ซึ่งเป็นหน้าที่รับผิดชอบของฟังก์ชัน vtkHandleMessage ซึ่งจัดการแป้นพิมพ์ภายในคลาส vtkWin32RenderWindowInteractor

ตารางที่ 5-1 คลาสที่ต้องปรับปรุงหรือเพิ่มเติมฟังก์ชัน

คลาส	ฟังก์ชันที่เปลี่ยนแปลง	ฟังก์ชันที่เพิ่มเติม
vtkRenderWindow	StereoUpdate StereoMidpoint StereoRenderComplete CopyResultFrame	
vtkWinOglrRenderWindow		GetOddRowPixel GetEvenRowPixel DrawRGBAFrame
vtkWin32RenderWindowInteractor	vtkHandleMessage	

5.3 การพัฒนาส่วนเสริม

โปรแกรมต้นฉบับภาษา C++ ของ VTK ประกอบด้วยแฟ้มข้อมูลชนิด *.cxx จำนวน 269 แฟ้ม และ ชนิด *.hh และ *.h จำนวน 295 แฟ้ม ในจำนวนนี้มีแฟ้มที่ต้องถูกเพิ่มเติมคำสั่งหรือฟังก์ชันใหม่ดังนี้ คือ แฟ้ม vtkRenderWindow.hh และ vtkRenderWindow.cxx สำหรับคลาส vtkRenderWindow, แฟ้ม vtkWin32OglrRenderWindow.hh และ vtkWin32OglrRenderWindow.cxx สำหรับคลาส vtkWin32OglrRenderWindow, และแฟ้ม vtkWin32RenderWindowInteractor.cxx สำหรับคลาส vtkWin32RenderWindowInteractor รูปที่ 5-4 ถึง รูปที่ 5-11 แสดงคำสั่งและฟังก์ชันที่เพิ่มเติมในแฟ้มต่างๆเหล่านี้ เพื่อสร้างส่วนเสริมการแสดงภาพสามมิติแบบสอดสลับให้กับ VTK

ขั้นตอนการพัฒนาส่วนเสริมมีดังนี้

1. แก้ไขแฟ้ม 5 แฟ้มที่กล่าวไว้ข้างต้น
2. สร้าง vtklib.lib ซึ่งเป็นคลังคำสั่งของ VTK สำหรับการเชื่อมต่อกับโปรแกรมภาษา C++ เพื่อสร้างงานจินตทัศน์ โดยการแปลและเชื่อมแฟ้มโปรแกรมที่เกี่ยวข้องทั้งหมดใน project vtklib.mdp
3. สร้าง vtktcllib.lib ซึ่งเป็นคลังคำสั่งของ VTK สำหรับการเชื่อมต่อกับโปรแกรม Tcl โดยการแปลและเชื่อมแฟ้มโปรแกรมที่เกี่ยวข้องทั้งหมดใน project vtktcllib.mdp

4. สร้าง vtk.exe จาก vtktcllib.lib และคลังคำสั่งอื่นๆ (เช่น tcl75.lib, tk41.lib, opengl32.lib, glaux.lib) เป็น Tcl/Tk application สำหรับ VTK โดยการแปลและเชื่อมแฟ้มโปรแกรมที่เกี่ยวข้องทั้งหมดใน project makefile.mak

ให้สังเกตว่าการสร้างภาพสามมิติแบบสอดสลับตามวิธีที่นำเสนอนี้ ใช้เวลาที่นอกจากจะแปรตามความซับซ้อนของรูปแบบวัตถุต่างๆในการจินตทัศน์แล้ว ยังแปรตามขนาดของวินโดวส์ที่แสดงด้วย เนื่องจากต้องใช้เวลาในการย้ายภาพเส้นคี่และเส้นคู่ของภาพที่สร้างได้สองครั้งสำหรับตาทั้งสอง เวลาในการย้ายนี้แปรผันโดยตรงกับขนาดของวินโดว์ ทั้งนี้เนื่องจากฟังก์ชัน `GetOddRowPixel` `GetEvenRowPixel` และ `DrawRGBAFrame` ทำงานแปรตามขนาดของวินโดวส์ที่ทำสำเนาภาพทั้งสิ้น ดูรูปที่ 5-12 ประกอบ

```
#define   VTK_STEREO_INTERLACE     3

class vtkRenderWindow : public vtkObject
{
protected:
  ...

public:
  ...

  virtual void GetEvenRowPixel
            (unsigned char *data,int x1, int y1, int x2, int y2, int front);
  virtual void GetOddRowPixel
            (unsigned char *data,int x1, int y1, int x2, int y2, int front);
  virtual void DrawRGBAFrame
            (int x1, int y1, int x2, int y2,unsigned char *data, int front);
};
```

รูปที่ 5-4 คำสั่งเพิ่มเติมในแฟ้ม vtkRenderWindow.hh

```
void vtkRenderWindow::StereoUpdate(void)
{
  if (this->StereoRender && (!this->StereoStatus))
    {
    switch (this->StereoType)
      {
      case VTK_STEREO_RED_BLUE:
      case VTK_STEREO_INTERLACE:
         this->StereoStatus = 1;
      }
  else if ((!this->StereoRender) && this->StereoStatus)
    {
    switch (this->StereoType)
      {
      case VTK_STEREO_RED_BLUE:
      case VTK_STEREO_INTERLACE:
         this->StereoStatus = 0;
      }
    }
}
```

รูปที่ 5-5 คำสั่งเพิ่มเติม (แสดงด้วยตัวเอียง)ในแฟ้ม vtkRenderWindow.cxx (ส่วนที่ 1)

```
void vtkRenderWindow::StereoMidpoint(void)
{
  int *size;
  switch (this->StereoType)
    {
    case VTK_STEREO_RED_BLUE:
      {
         ...
      }
      break;

    case VTK_STEREO_INTERLACE:
      {
      int *size;

      size = this->GetSize();
      this->StereoBuffer = new unsigned char [size[0]*size[1]*4]; //RGBA
      this->GetOddRowPixel(this->StereoBuffer, 0,0,size[0]-1,size[1]-1,0);
      }
      break;
    }
}

void vtkRenderWindow::StereoRenderComplete(void)
{
  switch (this->StereoType)
    {
    case VTK_STEREO_RED_BLUE:
      {
        ...
      }
      break;
    case VTK_STEREO_INTERLACE:
      {
      int *size;

      size = this->GetSize();
      this->GetEvenRowPixel(this->StereoBuffer, 0,0,size[0]-1,size[1]-1,0);
      this->ResultFrame = this->StereoBuffer;
      this->StereoBuffer = NULL;
      }
      break;
    }
}

void vtkRenderWindow::CopyResultFrame(void)
{
  if (this->ResultFrame)
    {
    int    *size;
    size = this->GetSize();
    if (this->StereoType == VTK_STEREO_INTERLACE)
       {
       this->DrawRGBAFrame(0,0,size[0]-1,size[1]-1,this->ResultFrame,0);
       }
    else
       {
       this->SetPixelData(0,0,size[0]-1,size[1]-1,this->ResultFrame,0);
       }
    }
  this->Frame();
}
```

รูปที่ 5-6 คำสั่งเพิ่มเติม (แสดงด้วยตัวเอียง)ในแฟ้ม vtkRenderWindow.cxx (ส่วนที่ 2)

```
class vtkWin32OglrRenderWindow : public vtkRenderWindow
{
protected:
  ...

public:
  ...

  virtual void GetEvenRowPixel
            (unsigned char *data,int x1, int y1, int x2, int y2, int front);
  virtual void GetOddRowPixel
            (unsigned char *data,int x1, int y1, int x2, int y2, int front);
  virtual void DrawRGBAFrame
            (int x1, int y1, int x2, int y2,unsigned char *data, int front);
};
```

รูปที่ 5-7 คำสั่งเพิ่มเติม (แสดงด้วยตัวเอียง)ในแฟ้ม vtkWin32OglrRenderWindow.hh

```
void vtkWin32OglrRenderWindow::GetEvenRowPixel
     (unsigned char *data,int x1, int y1, int x2, int y2, int front)
{
  long     yloop;
  int      y_low, y_hi, x_low, x_hi;
  unsigned long   *buffer;

  wglMakeCurrent(this->DeviceContext, this->ContextId);

  if (y1 < y2) { y_low = y1; y_hi  = y2; }
  else         { y_low = y2; y_hi  = y1; }
  if (x1 < x2) { x_low = x1; x_hi  = x2; }
  else         { x_low = x2; x_hi  = x1; }

  if (front)
    glReadBuffer(GL_FRONT);
  else
    glReadBuffer(GL_BACK);

  buffer = (unsigned long *) data;
  for (yloop = y_hi; yloop >= y_low; yloop-=2)
    {
    // read in a row of pixels
    glReadPixels(x_low,yloop,(x_hi-x_low+1),1,
                 GL_RGBA,GL_UNSIGNED_BYTE, buffer+yloop*(x_hi-x_low+1));
    }
}
```

รูปที่ 5-8 ฟังก์ชัน GetEvenRowPixel ที่เพิ่มใหม่ใน vtkWin32OglrRenderWindow.cxx

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
void vtkWin32OglrRenderWindow::GetOddRowPixel
     (unsigned char *data,int x1, int y1, int x2, int y2, int front)
{
  long    yloop;
  int     y_low, y_hi, x_low, x_hi;
  unsigned long   *buffer;

  wglMakeCurrent(this->DeviceContext, this->ContextId);

  if (y1 < y2) { y_low = y1; y_hi  = y2; }
  else         { y_low = y2; y_hi  = y1; }
  if (x1 < x2) { x_low = x1; x_hi  = x2; }
  else         { x_low = x2; x_hi  = x1; }

  if (front)
    glReadBuffer(GL_FRONT);
  else
    glReadBuffer(GL_BACK);

  buffer = (unsigned long *) data;
  for (yloop = y_hi-1; yloop >= y_low; yloop-=2)
    {
    // read in a row of pixels
    glReadPixels(x_low,yloop,(x_hi-x_low+1),1,
                 GL_RGBA, GL_UNSIGNED_BYTE, buffer+yloop*(x_hi-x_low+1));
    }
}
```

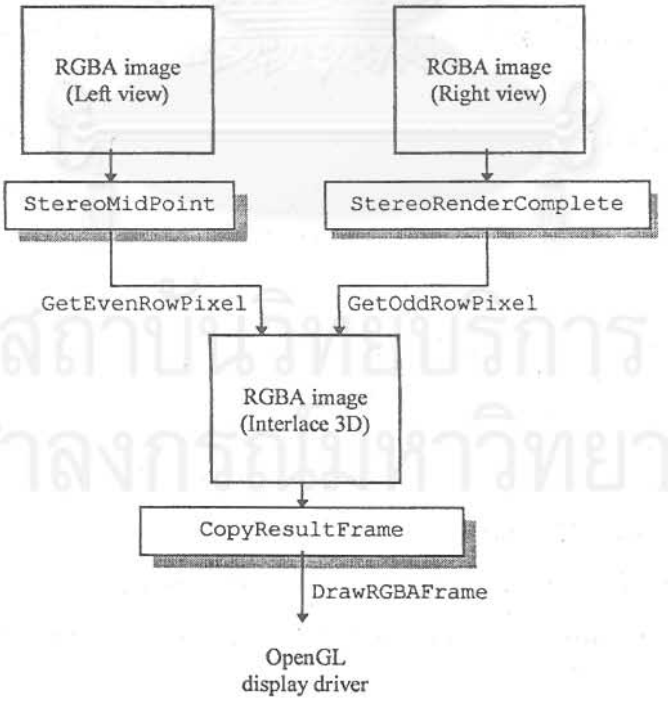
รูปที่ 5-9 ฟังก์ชัน GetOddRowPixel ที่เพิ่มใหม่ใน vtkWin32OglrRenderWindow.cxx

```
void vtkWin32OglrRenderWindow::DrawRGBAFrame
     (int x1, int y1, int x2, int y2, unsigned char *data, int front)
{
  int       y_low, y_hi, x_low, x_hi, yloop;
  unsigned long   *buffer;

  // set the current window
  wglMakeCurrent(this->DeviceContext, this->ContextId);

  if (y1 < y2) { y_low = y1; y_hi  = y2; }
  else         { y_low = y2; y_hi  = y1; }
  if (x1 < x2) { x_low = x1; x_hi  = x2; }
  else         { x_low = x2; x_hi  = x1; }

  if (front)
    glReadBuffer(GL_FRONT);
  else
    glReadBuffer(GL_BACK);

  // now write the binary info one row at a time
  buffer = (unsigned long *) data;
  for (yloop = y_hi; yloop>=y_low; yloop--)
    {
    glMatrixMode( GL_MODELVIEW );
    glPushMatrix();
    glLoadIdentity();
    glMatrixMode( GL_PROJECTION );
    glPushMatrix();
    glLoadIdentity();
    glRasterPos2f( 2.0 * (GLfloat)(x_low) / this->Size[0] - 1,
                   2.0 * (GLfloat)(yloop) / this->Size[1] - 1);
    glMatrixMode( GL_MODELVIEW );
    glPopMatrix();
    glMatrixMode( GL_PROJECTION );
    glPopMatrix();

    glDrawPixels((x_hi-x_low+1),1,GL_RGBA, GL_UNSIGNED_BYTE,
                 buffer+yloop*(x_hi-x_low+1));
    }
}
```

รูปที่ 5-10 ฟังก์ชัน DrawRGBAFrame ที่เพิ่มใหม่ใน vtkWin32OglrRenderWindow.cxx

```
LRESULT CALLBACK vtkHandleMessage
                 (HWND hWnd,UINT uMsg, WPARAM wParam, LPARAM lParam)
{
  ...

  switch (uMsg)
    {

    ...

    case WM_CHAR:
       switch (wParam)
          {
            ...

            case '4' :
              {
              if (me->RenderWindow->GetStereoType()==VTK_STEREO_INTERLACE)
                 me->RenderWindow->SetStereoType( VTK_STEREO_RED_BLUE );
              else
                 me->RenderWindow->SetStereoType( VTK_STEREO_INTERLACE );
              }
              break;

            ...
          }
          break;
    ...

    }
  return 0;
}
```

รูปที่ 5-11 คำสั่งเพิ่มเติม (แสดงด้วยตัวเอียง)ในแฟ้ม vtkWin32RenderWindowInteractor.cxx



รูปที่ 5-12 แผนผังการรวมภาพคู่

6. ส่วนเสริมแสดงภาพสามมิติแบบสอดสลับใน 3D-Studio MAX

จากที่ได้กล่าวมาในบทก่อนหน้านี้ถึงคุณลักษณะของซอฟต์แวร์ 3D-Studio MAX ซึ่งเป็นซอฟต์แวร์เพื่อการจินตทัศน์แบบจำลองสามมิติ ในบทนี้จะได้กล่าวถึงคุณสมบัติ โครงสร้างและรายละเอียดการพัฒนาส่วนเสริมแสดงภาพสามมิติแบบสอดสลับใน 3D-Studio MAX ซึ่งประกอบด้วยส่วนเสริมสองส่วนคือ ส่วนเสริมเพื่อการดูภาพสามมิติแบบสอดสลับระหว่างการออกแบบ และส่วนเสริมเพื่อการสร้างแฟ้มภาพยนตร์สามมิติแบบสอดสลับ

6.1 ส่วนเสริมเพื่อการดูภาพสามมิติแบบสอดสลับระหว่างการออกแบบ

ส่วนเสริมนี้เป็น plug-in ประเภท procedural object (ซึ่งแฟ้มรหัสโปรแกรมของส่วนเสริมจะเป็นประเภท *.dlo) มีไว้เพื่อให้ผู้ออกแบบแบบจำลองสามมิติด้วย 3D-Studio MAX สามารถมีวินโดว์เพื่อการดูภาพแบบจำลองสามมิติแบบสอดสลับระหว่างการออกแบบ เมื่อเปลี่ยนตำแหน่งกล้อง เปลี่ยนรูปลักษณะวัตถุต่างๆ หรือเปลี่ยนคุณลักณะใดๆของสภาพแวดล้อมของแบบจำลอง ก็จะปรากฏผลการเปลี่ยนแปลงดังกล่าวในวินโดว์สามมิติแบบสอดสลับดังกล่าวทันทีทันใด ส่วนเสริมที่พัฒนาขึ้นนี้ให้ชื่อคลาสว่า `StereObject`

ในมุมมองของผู้ออกแบบนั้น สภาพแวดล้อมของแบบจำลองจะต้องมีการติดตั้งกล้องสองกล้องที่มีคุณสมบัติเหมือนกันทุกประการ ทำหน้าที่เปรียบเสมือนตาสองข้างของผู้ออกแบบที่มองเข้าไปในแบบจำลอง กล้องทั้งสองนี้จะถูกวางให้มีระยะห่างกันตามแนวนอน (กำหนดโดยค่า e ที่ได้กล่าวถึงในหัวข้อ 2.4.2) กล้องทั้งสองนี้จะถูกเชื่อมเข้ากับ `StereObject` ซึ่งเป็นกรอบสี่เหลี่ยมล้อมรอบกล้องทั้งสองนี้ การเชื่อมนี้เองทำให้เมื่อมีการเคลื่อนย้าย `StereObject` จะทำให้กล้องทั้งสองเคลื่อนตามไปด้วย และ `StereObject` นี้เองจะรับภาพที่ได้จากกล้องทั้งสองเพื่อรวมภาพคู่ที่ได้เป็นภาพสามมิติแบบสอดสลับเพื่อแสดงออกวินโดว์ต่อไป

ในการพัฒนาส่วนเสริม `StereObject` นี้มีรายละเอียดที่ต้องคำนึงถึงนอกเหนือจากการพัฒนา plug-in ทั่วไปดังนี้

1. `StereObject` นี้จะเป็นคลาสที่สืบทอดมาจากคลาส `SimpleSpline` และคลาส `IParamArray` ทั้งนี้เนื่องจาก `StereObject` มีรูปลักษณะเป็นกรอบสี่เหลี่ยม (`SimpleSpline` เป็นคลาสที่ใช้ในการสร้างรูปทรงสองมิติ) และ `StereObject` จะต้องมีการรับพารามิเตอร์จากผู้ออกแบบด้วย (`IParamArray` เป็นคลาสที่ที่ควบคุมการทำงานการติดต่อกับผู้ใช้ - user interface control)

2. พารามิเตอร์ของ StereObject ที่ผู้ออกแบบต้องป้อนให้คือความกว้างความยาวของกรอบสี่เหลี่ยมวินโดว์ที่แสดงภาพจากกล้องทางซ้าย และวินโดว์ที่แสดงภาพจากกล้องทางขวา โดยลักษณะการป้อนพารามิเตอร์ความกว้างความยาวนั้นจะเป็น spinner control ส่วนการกำหนดวินโดว์แสดงภาพของกล้องทั้งสองนั้นจะใช้แบบเลือกวินโดว์แล้วกดปุ่ม

3. หน้าที่หลักของ StereObject ก็คือการรวมภาพที่ได้จากวินโดว์ (เรียกว่า GraphicsWindows) ของกล้องซ้ายและกล้องขวาเข้าด้วยกันแบบสอดสลับ แต่ละ GraphicsWindows นั้นจะมีเนื้อที่เก็บภาพที่แสดงอยู่ขณะนั้น ซึ่งเก็บในรูปแบบ DIB (Device Independent Bitmap)

4. การรวมภาพคู่ของ StereObject นั้นจะเกิดขึ้นก็ต่อเมื่อเกิดการเปลี่ยนแปลงตำแหน่งกล้อง นั่นคือเมื่อ StereObject เปลี่ยนแปลง เหตุการณ์การเปลี่ยนแปลงนี้สามารถรับรู้ได้ในฟังก์ชัน ValidForDisplay() เนื่องจากก่อนที่วัตถุใดๆจะถูกแสดง จะมีการเรียกฟังก์ชันนี้เพื่อตรวจสอบความถูกต้องของวัตถุนั้นๆ ดังนั้นโปรแกรมการรวมภาพจึงถูกใส่ไว้ในฟังก์ชันนี้

5. เนื่องจาก StereObject เป็นคลาสที่สืบทอดจาก SimpleSpline ที่มีพารามิตอร์เพิ่มเติม ดังนั้นในคลาสนี้ต้องมีโปรแกรมเพื่อการบันทึกและอ่านพารามิเตอร์เหล่านี้จากแฟ้มข้อมูล

ในส่วนเสริมที่กล่าวถึงนี้ มีฟังก์ชันต่างๆที่ทำหน้าที่พอแบ่งออกได้เป็นสี่กลุ่มด้วยกันคือ กลุ่มที่ต้องพัฒนาตามมาตรฐานของ 3D-Studio MAX plug-in เพื่อให้ข้อมูลแก่ 3D-Studio MAX กลุ่มที่สองเป็นฟังก์ชันในการรับค่าพารามิเตอร์ของส่วนเสริม กลุ่มที่สามเป็นฟังก์ชันการบันทึกและอ่านจากแฟ้ม และกลุ่มสุดท้ายเป็นกลุ่มที่จัดการเพื่อการรวมภาพคู่ จะขออธิบายรายละเอียดของฟังก์ชันกลุ่มสุดท้าย เนื่องจากเป็นหน้าที่หลักของส่วนเสริมนี้ (ส่วนกลุ่มอื่นๆนั้นมีรายละเอียดที่หาได้จากคู่มือการพัฒนา 3D-Studio MAX plug-in)

ดังที่กล่าวมาแล้วมาการรวมภาพคู่และแสดงนั้นเป็นชุดคำสั่งที่ทำงานในฟังก์ชัน ValidForDisplay ซึ่งมีขั้นตอนการทำงานหลักๆคือการอ่านพารามิเตอร์ของ StereObject จากนั้นดึงภาพที่ได้จากกล้องตาซ้าย (ฟังก์ชัน CreateLMap) ดึงภาพที่ได้จากกล้องตาขวา (ฟังก์ชัน CreateRMap) และรวมภาพทั้งสองเข้าด้วยกัน (ฟังก์ชัน ShowStere) ดังแสดงในรูปที่ 6-1 ให้สังเกตว่าในการทำงานของฟังก์ชันนี้จะไม่แสดงภาพทุกครั้งที่ถูกเรียก เนื่องจากการดึงและรวมภาพนั้นใช้เวลาการทำงานพอสมควร หากให้มีการรวมภาพทุกครั้งจะเสียเวลานานมาก จึงตั้งให้เกิดการรวมภาพทุกๆ k ครั้ง โดยค่า k ได้จากการทดลอง ซึ่งมีค่า 12 นั้นคือจะมีการรวมภาพและแสดงเมื่อฟังก์ชันนี้ถูกเรียกทุกๆ 12 ครั้ง

ภาพที่ได้จากกล้องซ้ายและกล้องขวานั้น ได้มาจากการเรียกๆฟังก์ชัน CreateLMap และ CreateRMap ตามลำดับ รายละเอียดของฟังก์ชันนี้แสดงในรูปที่ 6-2 (ในรูปนี้แสดงเฉพาะฟังก์ชัน CreateLMap สำหรับฟังก์ชัน CreateRMap นั้น ทำงานในลักษณะเดียวกัน) โดยการดึงภาพในรูปแบบ DIB (Deivce Independent Bitmap) จาก GraphWindows ของวินโดว์ที่แสดงภาพของกล้อง ด้วยฟังก์ชัน

GetDIB แล้วแปลงภาพที่ได้ให้อยู่ในรูปแบบ bitmap ด้วยฟังก์ชัน FromDIB เก็บในตัวแปร lmap เพื่อการรวมภาพต่อไป

เมื่อได้ภาพจากกล้องซ้ายและกล้องขวา การรวมภาพจะเกิดขึ้นในฟังก์ชัน ShowStere (รูปที่ 6-3) ซึ่งอาศัยฟังก์ชัน GetLinearPixels ในการอ่านเส้นคู่จากภาพของกล้องขวา ไปทับเส้นคู่ของภาพจากกล้องซ้าย ด้วยฟังก์ชัน PutPixels แล้วใช้ฟังก์ชัน Display ในการแสดงภาพออกอีกวินโดว์หนึ่ง ในกรณีที่เป็นการแสดงวินโดว์ครั้งแรก หากเคยมีวินโดว์สามมิตินี้อยู่แล้ว ก็จะใช้ฟังก์ชัน RefreshWindow ในการแสดงภาพใหม่บนวินโดว์เดิมที่มีอยู่

```
BOOL StereObject::ValidForDisplay(TimeValue t)
{
    float length, width;
    static i = 0 ;

    pblock->GetValue(PB_LENGTH, t, length, ivalid);
    pblock->GetValue(PB_WIDTH, t, width, ivalid);
    if (i >= 12)
        {
        // update stereoscopic image.
        CreateLMap();
        CreateRMap();
        ShowStere();
        i= -1;
        }
    i++;

    return (length == 0 || width == 0) ? FALSE : TRUE;
}
```

รูปที่ 6-1 ฟังก์ชันการแสดงภาพสามมิติแบบสอดสลับของส่วนเสริม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
void StereObject::CreateLMap()
{
    BITMAPINFO        *bmi = NULL;
    BitmapInfo         biFile;
    BITMAPINFOHEADER *bmih;
    int                size;

    if (leftGw == NULL || !IsWindow(leftGw->getHWnd()))
        return;

    leftGw->getDIB(NULL, &size);
    bmi  = (BITMAPINFO *)malloc(size);
    bmih = (BITMAPINFOHEADER *)bmi;
    leftGw->getDIB(bmi, &size);

    // set information to BitmapInfo variable.
    biFile.SetWidth((WORD)bmih->biWidth);
    biFile.SetHeight((WORD)bmih->biHeight);
    biFile.SetType(BMM_TRUE_32);

    if (!IsWindow(hlmap) && lmap != NULL)
        {
        delete lmap;
        lmap = NULL;
        hlmap = NULL;
        }

    if (lmap == NULL)
        {
        lmap = TheManager->Create(&biFile);
        lmap->FromDib(bmi);     // convert DIB to bitmap
        }
    else
        {
        Bitmap *map;

        map = TheManager->Create(&biFile);
        map->FromDib(bmi);
        lmap->CopyImage (map,COPY_IMAGE_RESIZE_LO_QUALITY,0);
        }

    if (bmi)
        free(bmi);
}
```

รูปที่ 6-2 ฟังก์ชันการดึงรูปจากกล้องซ้าย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
void StereObject::ShowStere()
{
    if (lmap == NULL || rmap == NULL)
        return;

    if (rmap != NULL)
        {
        BMM_Color_64 *line64 = NULL;

        if ((line64=(BMM_Color_64 *)
              calloc(rmap->Width(),sizeof(BMM_Color_64)))==NULL)
            return;

         for (int iy = 1; iy < rmap->Height(); iy+=2)
            {
            rmap->GetLinearPixels(0,iy,rmap->Width(),line64);
            lmap->PutPixels(0,iy,lmap->Width(),line64);
            }
        }

    if (!IsWindow(hlmap))
        {
        TCHAR title[MAX_PATH] = "Stereoscopic";

        lmap->Display(title, BMM_CN, TRUE, FALSE);
        hlmap = lmap->GetWindow();
       }
    else
        {
        lmap->RefreshWindow();        // redraw image
        }
}
```

รูปที่ 6-3 ฟังก์ชันการรวมภาพคู่แบบสอดสลับ

6.2 *ส่วนเสริมเพื่อการสร้างแฟ้มภาพยนตร์สามมิติแบบสอดสลับ*

ส่วนเสริมนี้เป็น plug-in ประเภท compositor ซึ่งเป็น image filter (ซึ่งแฟ้มรหัสโปรแกรมของส่วนเสริมจะเป็นประเภท *.flt) แบบหนึ่งที่รวมภาพสองภาพเข้าด้วยกัน ภาพๆหนึ่งเป็นภาพหลัก อีกภาพหนึ่งเป็นภาพรอง ส่วนเสริมนี้มีไว้สร้างแฟ้มภาพหรือภาพยนตร์ของแบบจำลองสามมิติ ที่สร้างโดย 3D-Studio MAX โดยจะเป็นภาพสามมิติแบบสอดสลับ (รูปแบบของแฟ้มนี้สามารถกำหนดได้ตามความสามารถของ 3D-Studio MAX เช่นเป็นแฟ้มแบบ *.AVI, *.FLC เป็นต้น) ส่วนเสริมที่พัฒนาขึ้นนี้ให้ชื่อคลาสว่า `ImageFilter_Stere`

`ImageFilter_Stere` เป็นคลาสที่สืบทอดมาจากคลาส `ImageFilter` โดยเราเพียงแต่เพิ่มเติมฟังก์ชันที่จำเป็นในการรวมภาพเท่านั้น (คือฟังก์ชัน `Render`) ส่วนวิธีการประสานติดต่อกับผู้ใช้นั้นจะใช้ขบวนการวิธีของการใช้งาน image filter ทั่วไป ซึ่งมีฟังก์ชันต่างๆรองรับอยู่แล้วในคลาส `ImageFilter` สำหรับฟังก์ชัน `Render` (ดูรูปที่ 6-4) นั้นทำการรวมภาพคู่แบบสอดสลับในลักษณะเดียวกับฟังก์ชัน `ShowStere` ของคลาส `StereObject` ที่กล่าวในหัวข้อที่แล้ว คือเพียงแต่ยึดภาพจากกล้องซ้ายเป็นหลัก (`lmap`) แล้วนำเส้นคี่ของภาพจากกล้องขวา (`rmap`) มาทับเส้นคี่ใน `lmap`

```
BOOL ImageFilter_Stere::Render(HWND hWnd)
{
   Bitmap        *lmap;
   Bitmap        *rmap;
   BMM_Color_64 *line64 = NULL;

   if ( !(lmap=srcmap) ) return (FALSE);
   if ( !(rmap=frgmap) ) return (FALSE);
   if ( lmap->width() != rmap->width() ) return (FALSE);

   if ((line64=(BMM_Color_64 *)
        calloc(rmap->Width(),sizeof(BMM_Color_64)))==NULL)
      return (FALSE);

   BOOL result = TRUE;
   BOOL abort  = FALSE;

   for (int iy = 1; iy < rmap->Height(); iy+=2)
      {
      //-- Progress Report
      SendMessage(hWnd,FLT_PROGRESS,iy,lmap->Height()-1);
      //-- Check for Abort
      SendMessage(hWnd,FLT_CHECKABORT,0,(LPARAM)(BOOL *)&abort);
      if (abort)
         {
         result = FALSE;
         break;
         }

      // get an odd line from the right image
      if (rmap->GetLinearPixels(0,iy,rmap->Width(),line64)!=1)
         {
         result = FALSE;
         break;
         }

      //-- put the line to the odd line of the left image
      if (lmap->PutPixels(0,iy,lmap->Width(),line64)!=1)
         {
         result = FALSE;
         break;
         }
      }

   if (line64) free(line64);

   return(result);

}
```

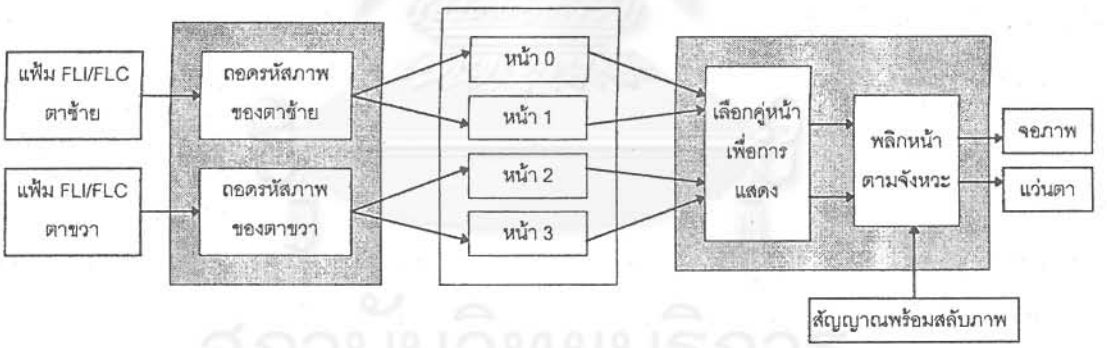
รูปที่ 6-4 ฟังก์ชันการรวมภาพคู่แบบสอดสลับของ ImageFilter_Stere

7. โปรแกรมแสดงภาพยนตร์ FLI/FLC สามมิติแบบพลิกหน้า

จากที่ได้กล่าวมาในบทที่แล้ว ถึงการเพิ่มองค์ประกอบในโปรแกรม 3D Studio MAX เพื่อทำ video post ของการสร้างภาพสามมิติสเตอริโอแบบแทรกสอดนั้น เป็นการรวมแฟ้มภาพยนตร์ของตาทั้งสอง ให้เป็นแฟ้มเดี่ยวที่สามารถใช้แสดงและรับรู้ภาพสามมิติ ในบทนี้จะขอนำเสนอโปรแกรมการแสดงแฟ้มภาพยนตร์จากแฟ้ม FLI/FLC ของตาทั้งสอง โดยใช้เทคนิคการแสดงภาพแบบพลิกหน้า (page flipping) ร่วมกับวงจรควบคุมแว่นผลึกเหลวที่กล่าวในบทที่ 4 ซึ่งสามารถแสดงภาพที่มีความละเอียดได้เต็มความสามารถของอุปกรณ์แสดงภาพ โดยโปรแกรมนี้ทำงานภายใต้สภาพปฏิบัติการ DOS

7.1 องค์ประกอบของโปรแกรม

โปรแกรมแสดงภาพยนตร์ FLI/FLC สามมิติสเตอริโอแบบพลิกหน้า (Page-flipping 3-D stereoscopic FLI/FLC movie player) ประกอบด้วยสองส่วนประมวลผล คือส่วนถอดรหัสภาพคู่จากแฟ้ม FLI/FLC เพื่อเก็บในหน้าหน่วยความจำแสดงภาพ และส่วนพลิกหน้าหน่วยความจำแสดงภาพเพื่อแสดงภาพสเตอริโอ รูปที่ 7-1 แสดงแผนผังการทำงานของโปรแกรม



รูปที่ 7-1 แผนผังการทำงานของโปรแกรม

ส่วนประมวลผลทั้งสองนี้ทำงานไปพร้อมกันแบบอสมวาร โดยมีหน่วยความจำแสดงภาพจำนวนสี่หน้าเป็นข้อมูลกลาง หน่วยความจำแสดงภาพสี่หน้านี้แบ่งออกเป็นสองชุด ชุดละสองหน้า คือชุดสำหรับตาซ้ายและตาขวาที่กำลังแสดงออกจอภาพ และชุดสำหรับตาซ้ายและตาขวาสำหรับการถอดรหัสภาพ แต่ละชุดจะใช้หน้าหน่วยความจำแสดงภาพที่สลับเปลี่ยนไปมา ระหว่างคู่หน้า 0,2 และคู่หน้า 1,3 การเปลี่ยนแปลงหมายเลขหน้าของชุดนั้น จะถูกกำหนดโดยส่วนถอดรหัสภาพคู่ ในแฟ้ม FLI/FLC จะมี

ตัวแปรระบุความเร็วในการแสดงภาพยนตร์ (เขตข้อมูล speed ของรูปแบบ FLI/FLC [1]) การถอดรหัสภาพนั้นจะกระทำตามจังหวะที่กำหนดโดยความเร็วนี้ เมื่อถึงเวลาการแสดงภาพคู่ใหม่ ก็จะถอดรหัสภาพหนึ่งคู่ใหม่และกำหนดให้หน้าหน่วยความจำคู่ที่เพิ่งได้รับภาพใหม่นั้น เป็นชุดสำหรับการแสดงออกจอภาพ และกำหนดให้หน้าหน่วยความจำอีกคู่เป็นชุดสำหรับการถอดรหัสในคราวต่อไป

สำหรับส่วนพลิกหน้านั้น จะเลือกคู่หน้าที่ถูกกำหนดให้เป็นชุดสำหรับการแสดงผล จากนั้นเตรียมพร้อมรอสัญญาณพร้อมสลับภาพ เมื่อได้รับสัญญาณนี้ส่วนพลิกหน้าจะกำหนดหน้าหน่วยความจำให้กับวงจรแสดงภาพ พร้อมทั้งส่งสัญญาณให้กับวงจรควบคุมแว่นผลึกเหลวที่เปิดปิดผลึกเหลวที่แว่นทั้งสองข้างให้เหมาะสม โดยจะพลิกหน้าสลับไปมาระหว่างหน้าหน่วยความจำสำหรับตาซ้ายและตาขวา ตามจังหวะของสัญญาณพร้อมสลับภาพที่ได้รับ สัญญาณพร้อมสลับภาพนี้เป็นสัญญาณขัดจังหวะ (interrupt) ที่สร้างขึ้นโดยตัวจับเวลา (timer) ของระบบ ที่ได้ถูกตั้งเวลาไว้ให้แปรตามสัญญาณสะบัดปืนอิเล็กตรอนตามแนวตั้งของวงจรแสดงภาพ

ส่วนประมวลผลทั้งสองที่กล่าวมานี้ทำงานไปพร้อมๆกัน โดยที่ส่วนถอดรหัสนั้นจะทำงานเป็นโปรแกรมหลัก อยู่ในวงวนการอ่านแฟ้มภาพยนตร์ ถอดรหัส รอเวลาแสดงภาพ และกำหนดชุดหน้าหน่วยความจำแสดงภาพ ในขณะที่ส่วนพลิกหน้าเพื่อแสดงภาพนั้นเป็นฟังก์ชันที่ถูกเรียกโดยระบบตามสัญญาณการขัดจังหวะของตัวจับเวลา

7.2 ส่วนถอดรหัสภาพคู่

หน้าที่หลักของโปรแกรมส่วนถอดรหัสภาพคู่ คือการทำงานแบบวงวนของการอ่านแฟ้มข้อมูลแฟ้ม FLI/FLC ถอดรหัสภาพภาพยนตร์ของสองแฟ้มนี้ ให้ได้ภาพทีละคู่ เพื่อนำไปเก็บไว้ในหน้าหน่วยความจำแสดงผลหนึ่งคู่ พร้อมทั้งระบุให้เป็นหน้าคู่ที่แสดงออกจอภาพ โดยวงวนนี้จะเริ่มรอบการทำงานใหม่ตามเวลาที่กำหนดในแฟ้มภาพยนตร์

เวลาในการถอดรหัสภาพเป็นประเด็นที่ต้องคำนึงมาก เพราะในที่นี้เราต้องถอดรหัสภาพจากแฟ้มสองแฟ้มพร้อมๆกัน เวลาส่วนใหญ่ที่เสียไปถูกใช้ในช่วงการอ่านข้อมูลจากแฟ้มข้อมูล ซึ่งโดยทั่วไปในเวลามากเกินกว่าจังหวะเวลาที่ระบุไว้ในแฟ้มภาพยนตร์ ดังนั้นวิธีการแก้ปัญหาก็คือการอ่านข้อมูลเข้าสู่หน่วยความจำแบบล่วงหน้า แล้วค่อยถอดรหัสในภายหลัง วิธีนี้จำต้องใช้บริการจากระบบจัดการหน่วยความจำส่วนขยายที่เรียกว่า extended memory เพื่อจองหน่วยความจำส่วนที่เกินจาก 1MB ที่ DOS รองรับได้ ตามมาตรฐาน XMS [2] ผ่านบริการขัดจังหวะหมายเลข 2Fh ของส่วนควบคุม HIMEM.SYS ซึ่งมีบริการให้เรียกใช้ได้หลายอย่างพอสรุปได้ในตารางที่ 7-1

---

1 รูปแบบของแฟ้ม FLI/FLC แสดงไว้ในแผ่น CD-ROM ประกอบรายงานนี้

2 มาตรฐาน XMS ถูกกำหนดโดยความร่วมมือของบริษัท Microsoft, Intel, Lotus และ AST รายละเอียดของการเรียกใช้ฟังก์ชันต่างๆใน XMS นั้น มีแสดงใน CD-ROM ประกอบรายงานนี้

ตารางที่ 7-1 บริการของส่วนจัดการหน่วยความจำขยาย

Function	Task
00h	Determine XMS version number
01h	Allocate High Memory Area (HMA)
02h	Free HMA
03h	Globally enable address line A20
04h	Globally disable address line A20
05h	Locally enable address line A20
06h	Locally disable address line A20
07h	Query status of address line A20
08h	Query free extended memory
09h	Allocate Extended Memory Block (EMB)
0Ah	Free allocated EMB
0Bh	Move EMB
0Ch	Lock EMB
0Dh	Unlock EMB
0Eh	Get EMB handle information
0Fh	Resize EMB
10h	Allocate Upper Memory Block (UMB)
11h	Free allocated UMB

ส่วนถอดรหัสภาพคู่มีโครงร่างการทำงานดังนี้

```
01: open two FLI/FLC files
02: validate file consistency
03: allocate XMS memory for FLI/FLC files
04: allocate virtual screen memory
05: do
06:    read header/data frame from the 2 FLI/FLC files
07:    if ( this is even numbered loop )
08:       uncompress the stereo pair to pages #0 and #2
09:       set pages #0 and #2 to be the visual pages
10:    else
11:       uncompress the stereo pair to pages #1 and #3
12:       set pages #1 and #3 to be the visual pages
13:    wait until next update time
14: } until ( receiving a quit command )
15: deallocate memory
```

เริ่มด้วยการเปิดแฟ้มภาพยนตร์สองแฟ้ม ตามด้วยการตรวจสอบความถูกต้องของแฟ้มทั้งสองว่ามีคุณสมบัติเหมือนกันหรือไม่ (เช่นจำนวนภาพ ความละเอียด เป็นต้น ซึ่งต้องเหมือนกัน จึงแสดงร่วมกันได้) จากนั้นมีการจองหน่วยความจำเพื่อรองรับแฟ้มทั้งสอง และหน่วยความจำที่เก็บหน้าการแสดงผล จากนั้นเข้าสู่วงวน (บรรทัดที่ 5 ถึง 14) ที่จะวนทำไปจนกระทำผู้ใช้สั่งให้เลิก โดยจะเป็นการอ่านภาพ ถอดรหัสภาพคู่ แล้วเก็บลงในหน้าหน่วยความจำแสดงผล ซึ่งจะเป็นหน้า 0 กับ 2 หรือ หน้า 1 กับ 3 ขึ้นกับว่าขณะนั้นเป็นการอ่านรอบที่เป็นเลขคู่ หรือเลขคี่ แล้วกำหนดหน้าคู่นั้นเป็นหน้าที่แสดงออกจอภาพ (ข้อมูลนี้ถูกใช้โดยส่วนพลิกหน้าแสดงภาพที่จะกล่าวต่อไป) จากนั้นรอให้ถึงเวลาแสดงภาพถัดไป เมื่อสิ้นสุดการทำงาน ก็จะคืนหน่วยความจำที่ได้จองมากลับสู่ระบบ

7.3 ส่วนพลิกหน้าแสดงภาพ

หน้าที่หลักของส่วนพลิกหน้าแสดงภาพคือ การเลือกหน้าหน่วยความจำแสดงผลที่เหมาะสมแสดงออกสู่จอภาพ เมื่อได้รับสัญญาณพร้อมสลับภาพจากแผงวงจรแสดงผล การเลือกหน้าแสดงนี้จะสลับกันไประหว่างหน้าที่เก็บภาพสำหรับตาซ้าย และหน้าที่เก็บภาพสำหรับตาขวา พร้อมกับตั้งค่าของจานสีสำหรับหน้านี้ และส่งสัญญาณควบคุมแผ่นผลึกเหลวออกผ่านทางพอร์ตอนุกรม การสลับหน้านี้จะใช้คำสั่งพลิกหน้าของแผงวงจรแสดงภาพ ซึ่งทำการเปลี่ยนหน้าหน่วยความจำที่ใช้แสดงภาพได้อย่างรวดเร็ว โดยการกำหนดที่อยู่เริ่มต้นของหน้นแสดงให้กับแผงวงจรแสดงผล ส่วนพลิกหน้าแสดงภาพนี้มีโครงร่างของทำงานดังนี้

```
01: if ( previously showed the left page )
02:    SetDisplayStartAddr( CurrentRightPageNumber )
03:    SetColorPalette( CurrentRightPageNumber )
04:    SetLCSGlass( RIGHT );
05: else
06:    SetDisplayStartAddr( CurrentLeftPageNumber )
07:    SetColorPalette( CurrentLeftPageNumber )
08:    SetLCSGlass( LEFT );
09: interrupt return
```

ลักษณะการทำงานของส่วนพลิกหน้านี้จะไม่เหมือนกับที่ได้กล่าวไว้ในหัวข้อ 4.2.1 เนื่องจากในส่วนพลิกหน้าภาพต้องทำงานไปพร้อมๆกับส่วนถอดรหัสภาพ ดังนั้นส่วนพลิกหน้าภาพนี้จะเป็นส่วนที่ถูกติดตั้งในระบบแบบที่ถูกเรียกจากเหตุการณ์ขัดจังหวะ ของนาฬิกาตั้งเวลา (timer interrupt) ซึ่งตั้งไว้ให้เกิดการขัดจังหวะพอดีกับสัญญาณพร้อมสลับภาพซึ่งถูกกำหนดโดยสัญญาณ vertical retrace ของแผงวงจรแสดงภาพที่ส่งไปจอภาพ [3] โดยก่อนทำการติดตั้งส่วนโปรแกรมนี้ โปรแกรมหลักจะวัดช่วงเวลาระหว่างสัญญาณ vertical retrace ที่เกิดขึ้น ด้วยส่วนโปรแกรมที่มีโครงร่างดังนี้

```
01: WaitforVerticalRetrace
02: ResetAndStartCounter
03: WaitforVerticalRetrace
04: StopCounter
05: VSyncDuration = GetCounterValue
06: SetAndStartTimer( VSyncDuration )
```

เริ่มต้นใช้ฟังก์ชัน `WaitforVerticalRetrace` เพื่อวนรอจนกว่าจะได้รับสัญญาณ vertical retrace (ซึ่งกระทำได้โดยการวนอ่านค่าจากพอร์ตหมายเลข `0x3DA` เพื่อตรวจสอบบิต b_3 เมื่อมีการเปลี่ยนจากระดับหนึ่งเป็นศูนย์ ก็แสดงว่าเกิด vertical retrace) แล้วเริ่มจับเวลาด้วยนาฬิการจับเวลาของระบบ (ซึ่งคือ 8254 counter ที่มีหมายเลขพอร์ตการติดต่อคือ 0x40 และ 0x43 ทำงานที่ mode 2) แล้ววนรอสัญญาณ vertical retrace อีกครั้งหนึ่ง เมื่อเกิดสัญญาณ vertical retrace ก็ให้บันทึกค่าตัวเลขของนาฬิกาจับเวลาไว้ โดยจะใช้ค่าที่บันทึกนี้เองป้อนให้นาฬิกาตั้งเวลา (ซึ่งคือ 8254 Timer ทำงานในลักษณะที่เหมือนกันกับ

---

[3] มีบางแผงวงจรที่สามารถสร้างเหตุการณ์ขัดจังหวะจากสัญญาณ vertical retrace ได้โดยตรง แต่เนื่องจากโครงการนี้ต้องการให้สามารถใช้ได้กับแผงวงจรแสดงภาพจำนวนมาก ที่ส่วนใหญ่ไม่มีคุณสมบัติดังกล่าว จึงจำต้องใช้เหตุการณ์ขัดจังหวะจากนาฬิกาตั้งเวลาแทน

8254 counter ที่กล่าวไว้ข้างต้น) เพื่อผลิตเหตุการณ์ขัดจังหวะ จะได้สลับภาพตามช่วงเวลาดังกล่าว (ซึ่งคือช่วงเวลาที่จะเกิดสัญญาณ vertical retrace)

ดังนั้นส่วนพลิกหน้าภาพจะถูกเรียกให้ทำงานทุกๆครั้งที่เกิดเหตุการณ์ขัดจังหวะจากนาฬิกาตั้งเวลา ซึ่งมีความถี่การขัดจังหวะเท่ากับความถี่ของสัญญาณ vertical retrace แต่ในทางปฏิบัติแล้ว การขัดจังหวะจากนาฬิกากับสัญญาณ vertical retrace ทั้งสองนี้จะเกิดไม่ตรงกันนัก ทำให้ยิ่งเวลาผ่านไปนานเท่าใดยิ่งคลาดเคลื่อนมากขึ้นเท่านั้น Dave Stampe และคณะ[4] แนะนำว่าควรมีการปรับการเริ่มตั้งเวลาใหม่หลังจากผ่านการขัดจังหวะไปทุกๆ 15 ครั้ง โดยเมื่อเกิดเหตุการณ์ครบทุก 15 ครั้ง จะมีวนรอสัญญาณ vertical retrace เมื่อดักพบ ก็เริ่มให้นาฬิกาตั้งเวลาทำงานใหม่ ก็จะทำให้การทำงานถูกต้อง ดังนั้นจะได้โครงร่างการทำงานของส่วนพลิกหน้าแสดงภาพดังนี้

```
01: if ( have not adjusted for 15 interrupts )
02:    WaitforVerticalRetrace
03:    SetAndStartTimer( VSyncDuration )
04:
05: if ( previously showed the left page )
06:    SetDisplayStartAddr( CurrentRightPageNumber )
07:    SetColorPalette( CurrentRightPageNumber )
08:    SetLCSGlass( RIGHT );
09: else
10:    SetDisplayStartAddr( CurrentLeftPageNumber )
11:    SetColorPalette( CurrentLeftPageNumber )
12:    SetLCSGlass( LEFT );
13:
14: interrupt return
```

7.4 การใช้มาตรฐาน VESA

จากที่ได้กล่าวในหัวข้อที่แล้วมา จะเห็นได้ว่าการทำงานของโปรแกรมนี้ยุ่งเกี่ยวกับการควบคุมแผงวงจรแสดงผล แต่เนื่องจากผู้ผลิตแผงวงจรแสดงผลมีการเพิ่มความสามารถพิเศษต่างๆเช่น การจัดการการอ้างอิงหน้าหน่วยความจำแสดงผล การพลิกหน้า การเลือกความละเอียดและจำนวนสี การใส่ค่าในจานสี เป็นต้น โดยมีวิธีโปรแกรมเพื่อใช้ขีดความสามารถนี้ต่างๆกันไป จึงเกิดปัญหาในการพัฒนาซอฟต์แวร์ที่ใช้ได้กับแผงวงจรแสดงผลหลากหลายยี่ห้อ ทำให้ผู้ผลิตหลายรายร่วมกันจัดตั้งสมาคมมาตรฐานทางการแสดงภาพอิเล็กทรอนิกส์ (VESA - Video Electronic Standard Association) ขึ้นมาเพื่อจัดมาตรฐานการใช้ความสามารถของแผงวงจรแสดงผลผ่านทางชุดคำสั่ง API (application programming interface) ที่เป็นมาตรฐานเดียวในลักษณะเป็นส่วนขยายจากชุด API ใน BIOS ของระบบ (มีชื่อเรียกว่า VBE - VESA Bios Extension) ในปัจจุบันแผงวงจรแสดงผลโดยทั่วไปจะมีชุด API ตามมาตรฐาน VESA แต่ก็มีอีกหลายยี่ห้อที่ไม่ทำตามมาตรฐาน หรือทำตามแต่ไม่ครบถ้วน สำหรับในกรณีหลังนี้สามารถแก้ปัญหาได้โดยการเพิ่ม

---

4 D. Stampe, B. Roehl, and J. Eagan, "*Virtual Reality Creations*," The Waite Group, 1993 pp. 483-484.

ซอฟต์แวร์ขับแผงวงจรแสดงผลในทำตามมาตรฐาน VESA (VESA software driver) เช่น Display Doctor ของบริษัท SciTech Software [5]

โปรแกรมแสดงภาพยนตร์ FLI/FLC ในงานสิ่งประดิษฐ์นี้ใช้มาตรฐาน VESA ในการติดต่อควบคุมการทำงานของแผงวงจรแสดงผล (ในกรณีที่ใช้แผงวงจรที่ไม่มีมาตรฐาน VESA ก็ใช้ซอฟต์ขับแผงวงจรให้ได้มาตรฐาน VESA เสริมด้วย) โดยการเรียกใช้ฟังก์ชันต่างๆในคลังคำสั่ง VBE ผ่านทางคำสั่ง int 10h [6] แต่เนื่องจากการใช้งาน VBE โดยตรงนั้นมีความซับซ้อนและต้องใช้ภาษา assembly ในการเรียกใช้บริการ จึงใช้บริการ VBE เฉพาะฟังก์ชันการพลิกหน้าเท่านั้น เนื่องจากเป็นส่วนที่ต้องการความเร็วในการทำงาน (การพลิกหน้านี้ใช้ ฟังก์ชันหมายเลข 07h ของ int 10h เพื่อตั้งจุดเริ่มต้นของหน้าหน่วยความจำในการแสดงภาพ) ส่วนบริการอื่นๆของ VBE ที่ประเด็นเรื่องเวลาการทำงานไม่ใช่ปัจจัยหลัก จะใช้ชุดคลังคำสั่ง SuperVGA Kit version 5.2 ของบริษัท SciTech Software [7] ซึ่งมีฟังก์ชัน ภาษา C จำนวนหนึ่งเพื่อการสั่งงานแผงวงจรแสดงผลตามมาตรฐาน VESA

---

5 http://www.scitechsoft.com

6 รายละเอียดของฟังก์ชันต่างๆของ VBE อยู่ในแผ่น CD-ROM ประกอบรายงานโครงการสิ่งประดิษฐ์นี้ หรือสามารถหาได้จาก ftp://ftp.scitechsoft.com/devel/docs/vbe20-11.exe

7 SuperVGA Kit เป็น freeware ของ SciTech Software Inc. ตัวคลังคำสั่งและเอกสารวิธีใช้อยู่ในแผ่น CD-ROM ประกอบรายงานโครงการสิ่งประดิษฐ์นี้ หรือสามารถหาได้จาก http://www.scitechsoft.com

8. บทสรุปและข้อเสนอแนะ

โครงการสิ่งประดิษฐ์ได้นำแนวคิดของการรับรู้ความลึกจากความแตกต่างทางแนวนอนของภาพที่ตามนุษย์เห็น การแสดงภาพสเตอริโอสามมิติแบบสลับเชิงเวลา ประกอบกับแว่นเปิดปิดผลึกเหลว มาใช้เสริมให้กับระบบจินตทัศน์ เพื่อให้การนำเสนอข้อมูลต่างๆ มีมุมมองและการตีความหมายข้อมูลที่ดีขึ้น โดยการเพิ่มมิติการนำเสนอ ระบบจินตทัศน์ที่ได้รับการเสริมความสามารถการแสดงภาพสามมิติในโครงการนี้ คือ Visualization ToolKit (VTK) ของบริษัทเจนเนอรัลอิเล็กทริก และ 3D-Studio MAX ของบริษัทออโตเดกส์ เหตุผลที่เลือกการเสริมสามารถซอฟต์แวร์ทั้งสองนี้เนื่องจากเป็นซอฟต์แวร์ที่ได้รับความนิยม ใช้ได้กับเครื่องคอมพิวเตอร์ส่วนบุคคลราคาไม่แพง และการใช้แนวคิดเชิงวัตถุในการออกแบบ ตารางที่ 8-1 สรุปความสามารถเพิ่มเติมที่เสริมในซอฟต์แวร์ทั้งสอง ตารางที่ 8-2 สรุปวิธีการพัฒนาส่วนเสริม ตารางที่ 8-3 สรุปข้อจำกัดของส่วนเสริม และ ตารางที่ 8-4 สรุปข้อเสนอแนะของส่วนเสริม

ตารางที่ 8-1 ความสามารถในการแสดงภาพสามมิติที่เสริมเข้ายัง VTK และ 3D-Studio MAX

ระบบจินตทัศน์	ความสามารถของส่วนเสริม
Visualization ToolKit	1. เพิ่มความสามารถในการแสดงภาพสามมิติสเตอริโอแบบสอดสลับ 2. เพิ่มคำสั่งให้กับตัวโต้ตอบกับผู้ใช้ เพื่อเลือกวิธีการนำเสนอแบบสามมิติ 3. เพิ่มประเภทของการแสดงภาพสามมิติสเตอริโอแบบสอดสลับ ให้กับฟังก์ชันการแสดงผลที่มีอยู่เดิม 4. ใช้ความสามารถเสริมนี้ได้ในการพัฒนาระบบด้วยทั้งภาษา C++ และ Tcl
3D-Studio MAX	1. เพิ่มความสามารถในการดูภาพสามมิติแบบสอดสลับระหว่างการออกแบบแบบจำลอง โดยระบบติดต่อกับผู้ใช้ในลักษณะกราฟิกส์แบบเมนู ปุ่ม และช่องป้อนข้อมูลให้กับส่วนเสริม เพื่อเลือกวินโดว์แสดงภาพจากกล้องสองกล้อง ที่แทนตาทั้งสอง เพื่อการรวมภาพ 2. เพิ่มความสามารถในการสร้างภาพยนตร์สามมิติแบบสอดสลับของการเคลื่อนไหวของวัตถุในแบบจำลอง 3. มีโปรแกรมแยกต่างหากที่ทำงานบน DOS เพื่อการแสดงภาพยนตร์แบบ FLI/FLC ที่ได้จาก 3D-Studio MAX โดยแสดงเป็นสามมิติสเตอริโอแบบพลิกหน้า ทั้งนี้เพื่อให้ได้ความละเอียดของภาพสูงสุด

ตารางที่ 8-2 วิธีการพัฒนาส่วนเสริม

ส่วนเสริม	วิธีการพัฒนา
ส่วนเสริมของ VTK	1. ปรับปรุงฟังก์ชันที่เกี่ยวกับการแสดงภาพสเตอริโอ เพื่อเพิ่มการแสดงภาพสเตอริโอแบบสอดสลับ 2. เพิ่มเติมฟังก์ชันการอ่านและเขียน pixel แบบ RGBA เพื่อความเร็วในการรวมภาพคู่แบบสอดสลับ 3. ปรับปรุงฟังก์ชันของตัวโต้ตอบให้รับแป้นพิมพ์ เพื่อการแสดงภาพสเตอริโอแบบสอดสลับ
ส่วนเสริมของ 3D-Studio MAX	1. พัฒนา plug-in แบบ procedural object เป็นตัวเชื่อมกล้องสำหรับตาซ้ายและตาขวา เป็นรับรับภาพคู่จากกล้องทั้งสอง แล้วรวมเป็นภาพเดียวแบบสอดสลับ เพื่อให้ผู้ใช้ดูระหว่างการออกแบบแบบจำลอง 2. พัฒนา plug-in แบบ image filter compositor เพื่อรวมภาพที่ได้จาก animation sequence ของแบบจำลอง เป็นแฟ้มภาพยนตร์ ที่เก็บการรวมภาพคู่แบบสอดสลับ
โปรแกรมแสดงภาพยนตร์ FLI/FLC	1. พัฒนาโปรแกรมอ่านแฟ้มภาพยนตร์ FLI/FLC สองแฟ้ม สองตาสองข้าง เพื่อแสดงออกจอภาพแบบพลิกหน้า โดยใช้มาตรฐาน XMS ในการจัดการหน่วยความจำแบบ ใช้มาตรฐาน VESA ในการควบคุมแผงวงจรแสดงผลแบบ VESA และใช้ตัวจับ/ตั้งเวลาของเครื่องคอมพิวเตอร์ในการประสานจังหวะการสลับภาพ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 8-3 ข้อจำกัดของส่วนเสริม

ส่วนเสริม	ข้อจำกัด
ส่วนเสริมของ VTK	1. ใช้ได้เฉพาะกับ VTK รุ่นที่ 1.2 ที่ทำงานในระบบปฏิบัติการ Windows95 หรือ Windows NT ในขณะที่เขียนรายงานฉบับสมบูรณ์นั้น (25 ตุลาคม 2540) VTK ได้รับการพัฒนาจนเป็นรุ่นที่ 2.0 ที่มีคุณสมบัติเพิ่มเติมทางด้านการจินตทัศน์อีกมากมาย (แต่ยังไม่มีทางด้านแสดงภาพสามมิติสเตอริโอแบบสอดสลับ) 2. ผู้ใช้ต้องตั้งให้จอภาพแสดงผลแบบสอดสลับเอง ในกรณีที่ใช้วงจรควบคุมแบบใช้รหัสเส้นขาว ก็ต้องเรียกโปรแกรมที่แสดงรหัสเส้นขาวเอง
ส่วนเสริมของ 3D-Studio MAX	1. ใช้ได้เฉพาะกับ 3D-Studio MAX รุ่นที่ 1.2 ขึ้นไป เนื่องจากใช้ชุดพัฒนาซอฟต์แวร์ส่วนเสริมของรุ่นนี้ 2. ผู้ออกแบบต้องตั้งกล้องซ้ายและกล้องขวาให้มีคุณสมบัติเหมือนกันทุกประการ ยกเว้นตำแหน่งกล้อง 3. การแสดงภาพสามมิติสเตอริโอแบบสอดสลับในระหว่างการออกแบบนั้นอาจใช้เวลานานกว่าที่ผู้ออกแบบเปลี่ยนแปลงแบบจำลอง ทั้งนี้ขึ้นกับความซับซ้อนของแบบจำลอง และความเร็วของหน่วยประมาลผล ส่งผลให้เกิดการแสดงที่ผิดพลาดได้ อย่างไรก็ตามเมื่อแสดงผิดพลาด ผู้ออกแบบสามารถกดปุ่ม Redraw เพื่อให้แสดงภาพที่ถูกต้องได้ 4. ผู้ใช้ต้องตั้งให้จอภาพแสดงผลแบบสอดสลับเอง ในกรณีที่ใช้วงจรควบคุมแบบใช้รหัสเส้นขาว ก็ต้องเรียกโปรแกรมที่แสดงรหัสเส้นขาวเอง
โปรแกรมแสดงภาพยนตร์ FLI/FLC	1. ใช้ได้กับวงจรควบคุมแว่นผลึกเหลวในรูปที่ 4-4 ที่ใช้เทคนิคการแสดงแบบพลิกหน้าเท่านั้น 2. เครื่องคอมพิวเตอร์ที่ใช้ต้องมีและไม่ใช้พอร์ตอนุกรมหมายเลขสอง เนื่องจากโปรแกรมนี้ใช้สัญญาณของพอร์ตนี้ 3. ใช้ได้กับแผงวงจรแสดงผล Matrox MGA Millenium ซึ่งเป็นแผงวงจรที่สามารถสนองตอบต่อความเร็วในการปรับจานสีได้ทันกับความถี่ในการพลิกหน้า

ตารางที่ 8-4 ข้อเสนอแนะเพื่อการปรับปรุงส่วนเสริม

ส่วนเสริม	ข้อเสนอแนะ
ส่วนเสริมของ VTK	1. นำแนวคิดในการปรับปรุงที่ได้พัฒนามา ไปใช้กับ VTK ในรุ่นใหม่ๆ 2. ปรับปรุงการเลือกใช้ฟังก์ชันของ OpenGL เพื่อการรวมภาพแบบสอดสลับที่มีประสิทธิภาพมากขึ้น
ส่วนเสริมของ 3D-Studio MAX	1. เขียนคลาส StereObject ใหม่โดยสืบทอดจากคลาส helper จะเหมาะสมกว่า เนื่องจากเป็นเพียงตัวเสริมให้กล้องเกาะติดกันเท่านั้น หรือพัฒนาให้เป็นคลาสย่อยของ CameraObject ที่มีลักษณะเทียบได้กับ stereo camera ที่มีสองเลนส์รับภาพในตัว
โปรแกรมแสดงภาพยนตร์ FLI/FLC	1. พัฒนาให้ใช้ได้กับการแสดงภาพแบบพลิกหน้าในระบบปฏิบัติการ Windows ผ่านคลังคำสั่ง DirectX ของบริษัทไมโครซอฟต์ หรือคลังคำสั่ง WinDirect ของบริษัทไซเทคซอฟต์แวร์ 2. สำหรับวงจรควบคุมแว่นผลึกเหลวของ SimulEyes นั้นสามารถใช้กับการแสดงแบบพลิกหน้าได้ ถ้ามีการแสดงเส้นรหัส ดังนั้นควรเพิ่มคุณสมบัติในการวาดเส้นรหัสขาวที่เส้นเส้นล่างสุดจองจอภาพ จะได้ใช้โปรแกรมนี้ได้กับวงจรควบคุมแว่นของ SimulEyes
อื่นๆ	1. เพิ่มคุณสมบัติในการตั้งการทำงานจอภาพให้เป็นแบบสอดสลับ พร้อมทั้งวาดเส้นรหัสขาว (ถ้าวงจรควบคุมต้องการ) โดยอัตโนมัติจากส่วนเสริม 2. นำแนวคิดการแสดงภาพสเตอริโอไปสร้างส่วนเสริมให้กับระบบอื่นๆ เช่น Mathematica และ MathLab ซึ่งเป็นระบบการประมวลทางคณิตศาสตร์ที่มีความสามารถในการแสดงกราฟของฟังก์ชันต่างๆ 3. ปรับปรุงส่วนเสริมให้รองรับการแสดงภาพสามมิติแบบบนล่าง ที่ให้ความรู้สึกสบายตากว่าแบบสอดสลับ 4. พัฒนาคลังคำสั่งทางคอมพิวเตอร์กราฟิกส์เพื่อสร้างภาพสามมิติสเตอริโอแบบสอดสลับให้กับภาษาจาวา เพื่อการแสดงภาพสามมิติในระบบอินเตอร์เน็ต

เอกสารอ้างอิง

1. P. R. Keller and M. M. Keller, Visual Cues: Practical Data Visualization, IEEE Computer Society Press, June 1993.
2. S. Prasitjutrakul and W. Watcharawittayakul, "Algorithm Visualization System: An Overview of Internal Structure", Proc. of Thrid ASEAN Regional Seminar on Microelectronics and Information Technology, Bangkok, August 1994.
3. Stereographics, Stereographics Resource Guide, Stereographics Corporation, 1994
4. David F. Mcallister, "3-D Displays", BYTES, May 1992
5. http://pcvr.kasan.co.kr/3d_max.html
6. http://www.vmaxx.com/cmaxhome.html
7. http://www.stereographics.com
8. Larry F. Hodges, "Tutorial: Time-Multiplexed Stereoscopic Computer Graphics", IEEE Computer Graphics & Applications, March 1996, pp.20-30.
9. W. Schroeder, K. Martin, and B. Lorensen, "The Visualization ToolKit : An Object-Oriented Approach to 3D Graphics", Prentice Hall, 1996.
10. J.K. Ousterhout, Tcl and the Tk Toolkit. Addison-Wesley Publishing Company, Reading, M.A. 1984.
11. D. Stampe, B. Roehl, and J. Eagan, "Virtual Reality Creations," The Waite Group, 1993
12. http://www.scitechsoft.com
13. ftp://ftp.scitechsoft.com/devel/docs/vbe20-11.exe

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก การใช้งานส่วนเสริม

การใช้งานส่วนเสริมแสดงภาพสามมิติแบบสอดสลับใน VTK

ส่วนเสริมที่เพิ่มเติมใน VTK นั้นจะทำให้ผู้ใช้สามารถเลือกสลับการแสดงภาพสามมิติแบบแดงน้ำเงิน (ที่มีอยู่เดิม) กับแบบสอดสลับ (ที่เพิ่มใหม่) ด้วยการกดแป้นพิมพ์ '4' หากโปรแกรมจินตทัศน์ที่ทำงานใน VTK นั้นใช้ตัวโต้ตอบ (Interactor) ดังสรุปแป้นพิมพ์ที่เกี่ยวกับการแสดงแบบสามมิติของตัวโต้ตอบดังนี้

แป้นพิมพ์คำสั่งของตัวโต้ตอบ	ความหมาย
ตัวเลข '3'	สลับเลือก/ไม่เลือก การแสดงภาพสามมิติ
ตัวเลข '4'	สลับการแสดงภาพแบบสามมิติแดงน้ำเงิน / แบบสอดสลับ

สำหรับผู้พัฒนาโปรแกรมจินตทัศน์ด้วยภาษา Tcl/Tk นั้นสามารถเลือกการแสดงสามมิติแบบสอดสลับได้โดยการเติมคำสั่งข้างล่างนี้

```
$renWin StereoRenderOn;
$renWin SetStereoType 3;
$renWin Render;
```

ในกรณีที่พัฒนาด้วยภาษา C++ สามารถเลือกการแสดงสามมิติแบบสอดสลับได้ดังนี้

```
renWin->StereoRenderOn();
renWin->SetStereoType( 3 );
renWin->Render();
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

การใช้งานส่วนเสริมแสดงภาพสามมิติแบบสอดสลับใน 3D-Studio MAX

ส่วนเสริมของ 3D-Studio MAX ในการแสดงภาพสามมิติแบบสอดสลับ มีสองส่วนคือ ส่วนเสริมเพื่อการดูภาพสามมิติแบบสอดสลับระหว่างการออกแบบ และส่วนเสริมเพื่อการสร้างแฟ้มภาพยนตร์สามมิติแบบสอดสลับ ซึ่งมีวิธีใช้งานดังนี้

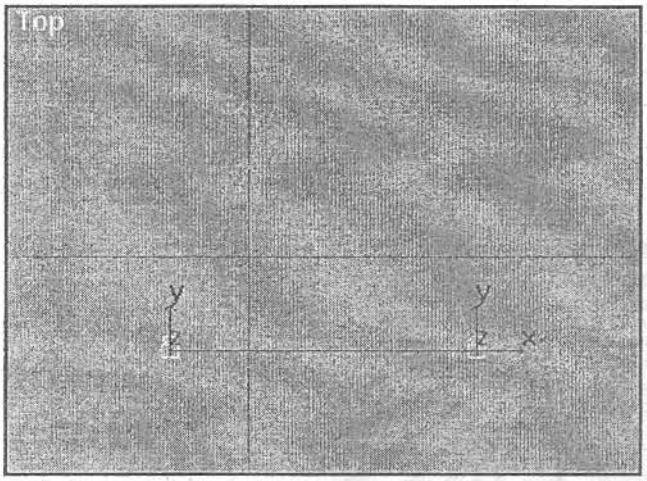
วิธีใช้ส่วนเสริมเพื่อการดูภาพสามมิติแบบสอดสลับระหว่างการออกแบบ

1. สร้าง Free Camera 2 ตัวโดยให้มีแกน y และ แกน z ขนานกัน แกน y ของกล้องทั้ง 2 ต้องมีขนาดเท่ากัน และ ใช้แกน x ร่วมกันเสมอ ดังรูปที่ 1
2. แสดงภาพที่มองจากกล้องทั้ง 2 ที่ view 2 viewport บนหน้าจอ
3. กดปุ่ม Create/ Shapes/ 3D Stereoscopic/ Dummy ดังรูปที่ 2
4. Click ขวาที่ view ของกล้องตาซ้ายแล้วกดที่ปุ่ม Pick Left view
5. Click ขวาที่ View ของกล้องตาขวา แล้วกดที่ปุ่ม Pick Right view จากขั้นตอนนี้ จะปรากฏหน้าต่างสำหรับดูภาพ Preview บนหน้าจอ
6. ให้แน่ใจว่าปุ่ม Dummy ถูกกดอยู่ แล้วจึงวาด Dummy บริเวณที่มีกล้องทั้ง 2 ตัว ควรวาดคร่อมกล้องทั้ง 2 โดยให้มีขอบเหลือพอสมควร เพื่อง่ายต่อการเลือก Dummy (รูปที่ 4 แสดงขั้นตอนของข้อ 4-6)
7. กดปุ่ม Select and Link บน toolbar
8. เลื่อน mouse มาที่กล้องตัวใดตัวหนึ่งกด Click ค้างไว้ แล้วลาก mouse มาบริเวณที่เป็น Dummy ให้แน่ใจว่า cursor เปลี่ยนเป็นสีเขียวแล้วจึงปล่อยปุ่ม mouse ซึ่งเป็นการกำหนดให้กล้องเป็น child ของ Dummy ดังรูปที่ 4
9. ทำแบบเดียวกันกับกล้องอีก 1 ตัว
10. หลังจากนี้ห้ามเคลื่อนย้ายกล้องทั้ง 2 ตัว โดยตรงแต่ให้เลื่อนจาก Dummy แทน เนื่องจากอาจทำให้ตำแหน่งและทิศทางต่างๆ ไม่ตรงตามข้อ 1
11. หากภาพบนหน้าต่าง Preview ไม่ถูกต้อง ให้กดเลือก Dummy แล้วกดปุ่ม Redraw จากหน้าต่าง Modify

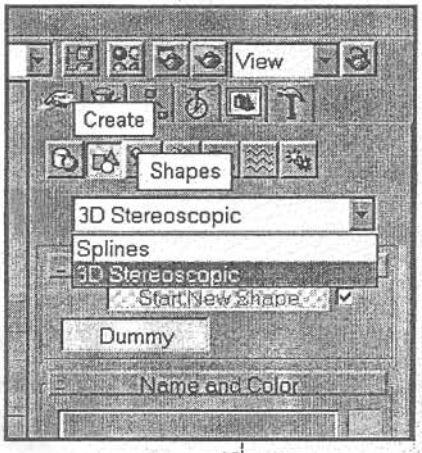
ข้อควรระวัง

- ขณะทำตามขั้นตอนที่ 3-5 หากกดปุ่มพลาดจะทำให้หน้าต่างของ 3D-Stereoscopic หาย ไป ให้ทำข้อ 6 ก่อนและให้แน่ใจว่า Dummy ถูกเลือกอยู่ แล้วจึงทำข้อ 4-5 จากหน้าจอ Modify ในภายหลัง

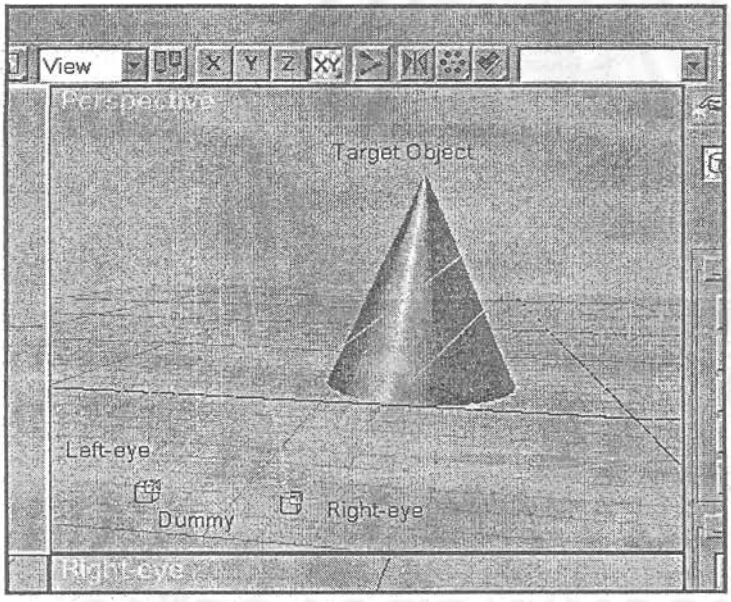
- ควรใส่แว่นตรวจดูภาพด้วยว่าควรปรับแก้ไขระยะระหว่างกล้องใหม่หรือไม่ เมื่อมีการเลื่อน Dummy ในลักษณะต่างๆ
- Parameter ต่างๆ ของกล้องทั้ง 2 ต้องเท่ากันทั้งหมดเสมอ

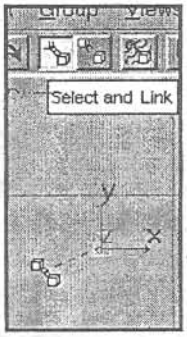


รูปที่ 8-1

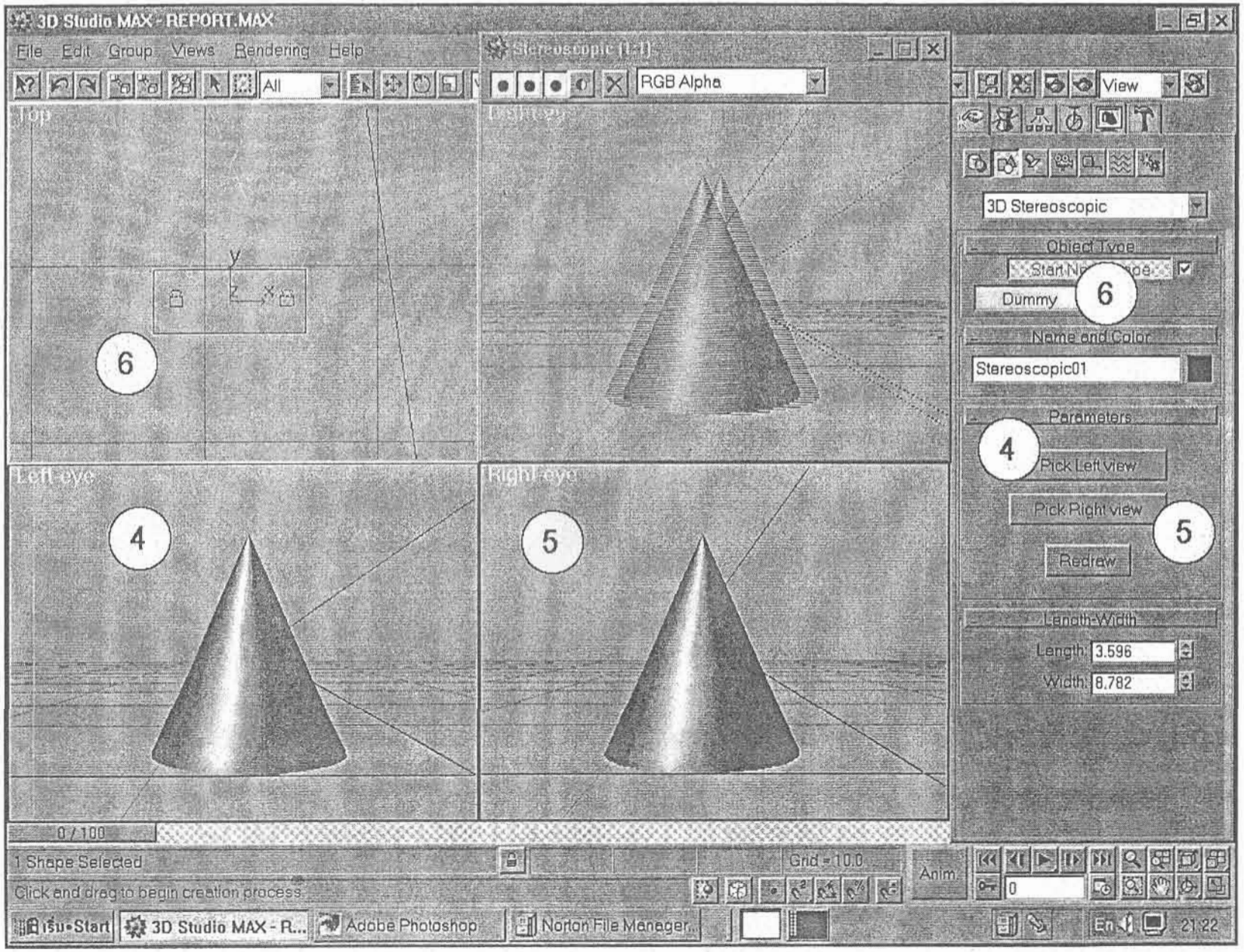


รูปที่ 8-2





รูปที่ 4


3D Studio MAX - REPORT.MAX
File Edit Group Views Rendering Help
All
Top
Left eye
Right eye
Stereoscopic (L+R)
RGB Alpha
View
3D Stereoscopic
Object Type
Dummy
6
Name and Color
Stereoscopic01
Parameters
4
Pick Left view
Pick Right view
5
Redraw
Length-Width
Length: 3.596
Width: 8.782
0 / 100
1 Shape Selected
Click and drag to begin creation process
Grid = 10.0
Anim
เริ่ม Start
3D Studio MAX - R...
Adobe Photoshop
Norton File Manager
21:22

รูปที่ 3

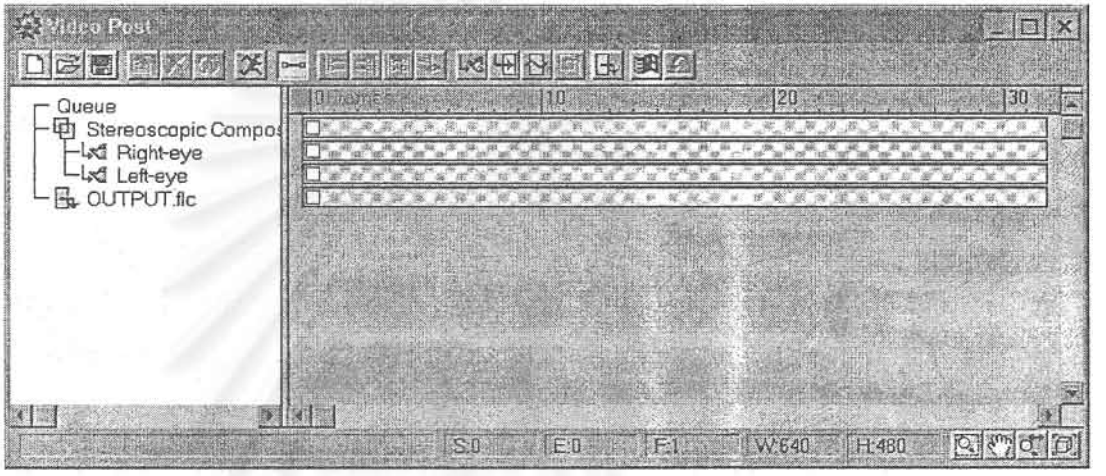
วิธีใช้ส่วนเสริมเพื่อการสร้างแฟ้มภาพยนตร์สามมิติแบบสอดสลับ

Add Scene Event

1. เลือก Rendering/Videopost จาก Menu Bar
2. เลือกภาพที่จะนำมา Render ให้เป็นภาพแบบ 3D Stereoscopic จากไฟล์ที่เคย Render มาแล้ว หรือจะ Render ใหม่จาก view ที่ตั้งไว้ตอนออกแบบโดยใช้กล้อง 2 ตัวก็ได้ ถ้าจะ Render จากไฟล์เก่าให้เลือกโดยกดที่ปุ่ม Add Scene Event แต่ถ้าต้องการ Render จาก view ให้กดปุ่ม Add Image Input Event

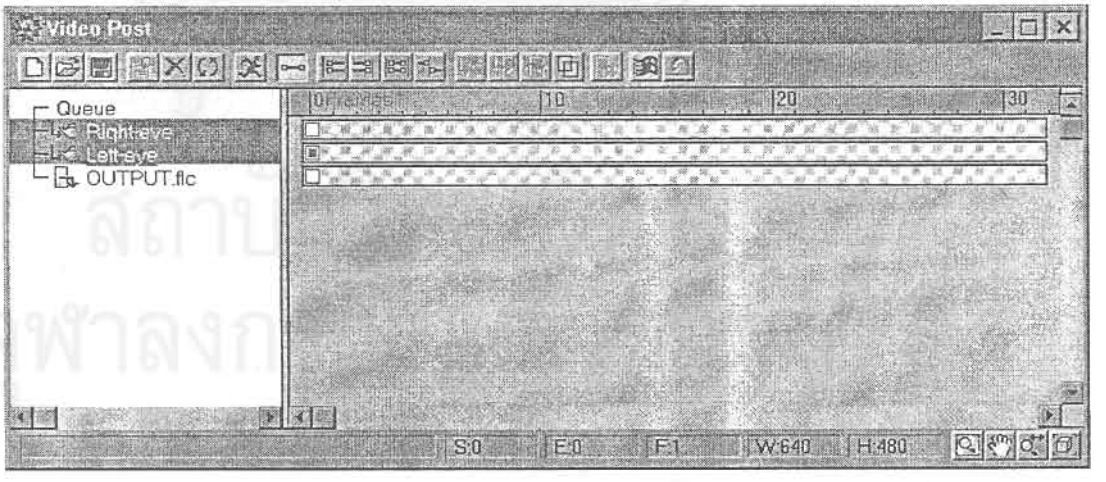
Add Image Input Event

3. ให้เลือกภาพที่จะทำมาเป็น Input ด้านตาขวา (หรือ ภาพที่ใช้เฉพาะสแกนไลด์เลขคี่) ก่อน
4. เลือกภาพที่จะให้เป็นตาซ้าย (สแกนไลด์เลขคู่)



Add Image Output Event

5. กดปุ่ม Add Image Output Event แล้วตั้งชื่อไฟล์ และ File Format ตามต้องการ
6. Click ที่ชื่อของภาพใดภาพหนึ่ง กด Ctrl ค้างไว้แล้ว Click ที่ภาพที่เหลืออีก 1 ภาพเพื่อให้เกิดแถบเลือกทั้ง 2 ภาพ จากขั้นนี้ปุ่ม Add Image Layer Event จะใช้งานได้แล้ว



Add Image Layer Event

Execute Sequence

7. กดปุ่ม Add Image Layer Event
8. เลือก Stereoscopic Compositor
9. กดปุ่ม Execute Sequence

การใช้งานโปรแกรมแสดงภาพยนตร์ FLI/FLC สามมิติแบบพลิกหน้า

โปรแกรมนี้ถูกแบ่งออกเป็นสองโปรแกรมย่อย คือ `flclr` และ `flilr` ซึ่งเป็นโปรแกรมแสดงภายนตร์ในรูปแบบ `flc` และ `fli` ตามลำดับ วิธีเรียกใช้ทั้งสองโปรแกรมนี้เป็นดังนี้

```
flclr left.flc right.flc speed
```

```
flilr left.fli right.fli speed
```

โดยที่ `left.flc` และ `left.fli` คือแฟ้มสำหรับตาซ้าย และ `right.flc` และ `right.fli` เป็นแฟ้มสำหรับตาขวา ในขณะที่ `speed` คือความเร็วในการแสดงภาพยนตร์ หากมีค่า 0 แสดงให้แสดงตามความเร็วที่เก็บในแฟ้มข้อมูล

ในขณะที่มีการแสดงภาพบนจอสามารถกดปุ่มลูกศรซ้ายและลูกศรขวา เพื่อปรับระยะห่างของภาพของตาซ้ายและภาพของตาขวา ให้ห่างตามแนวนอนน้อยกว่า (ลูกศรซ้าย) หรือมากกว่า (ลูกศรขวา) ที่กำหนดในแฟ้มข้อมูลได้ ทั้งนี้เนื่องจากในบางกรณีการเลื่อนภาพตามแนวนอนดังกล่าวจะทำให้รับรู้ความลึกได้เหมือนจริงขึ้น หากกดปุ่มลูกศรขึ้นหรือลง จะเป็นการตั้งระยะห่างกลับสู่สภาพเดิม และถ้าเป็นปุ่มอื่นๆ นอกเหนือจากที่กล่าวมาจะถือว่าเป็นการเลิกโปรแกรม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข แฟ้มข้อมูลใน CD-ROM

ตารางต่างๆข้างล่างนี้แสดง สารบบ พร้อมทั้งความหมาย ของข้อมูลในแผ่น CD-ROM ที่แนบมากับรายงานฉบับสมบูรณ์นี้

ชื่อสารบบ	ความหมาย
\3DSMAX	แฟ้มต่างๆที่เกี่ยวกับส่วนเสริมของ 3D-Studio MAX
\FLI-FLC	แฟ้มต่างๆที่เกี่ยวกับโปรแกรมแสดงแฟ้ม FLI-FLC สามมิติสเตอริโอแบบพลิกหน้า
\REPORT	รายงานฉบับสมบูรณ์ในรูปแบบของ Microsoft Word 95 แฟ้มหลักคือ MASTER.DOC ซึ่งจะเชื่อมกับแฟ้มรองอื่นๆโดยอัตโนมัติ
\UTIL	โปรแกรมอรรถประโยชน์ต่างๆ
\VTK	แฟ้มต่างๆที่เกี่ยวกับส่วนเสริมของ VTK

ชื่อสารบบ	ความหมาย
\3DSMAX\BINARY	STERE.DLO ส่วนเสริมเพื่อการดูภาพสามมิติแบบสอดสลับระหว่างการออกแบบ STEREFILTER.FLT ส่วนเสริมเพื่อการสร้างแฟ้มภาพยนตร์สามมิติแบบสอดสลับ
\3DSMAX\EXAMPLE	แฟ้มตัวอย่างแบบจำลอง และผลที่ได้ในรูปแบบของภาพยนตร์สามมิติสเตอริโอแบบสอดสลับ
\3DSMAX\MAXSDK	แฟ้มต่างๆของชุดพัฒนา plug-in ของ 3D-Studio MAX
\3DSMAX\PROGRAM	แฟ้มโปรแกรมต่างๆของส่วนเสริม
\3DSMAX\PROGRAM\VPOSTX	แฟ้มโปรแกรมต่างๆของส่วนเสริมเพื่อการสร้างแฟ้มภาพยนตร์สามมิติแบบสอดสลับ
\3DSMAX\PROGRAM\STEREOBJECT	แฟ้มโปรแกรมต่างๆของส่วนเสริมเพื่อการดูภาพสามมิติแบบสอดสลับระหว่างการออกแบบ

ชื่อสารบบ	ความหมาย
\FLI-FLC\BINARY	FLILR.EXE และ FLCLR.EXE โปรแกรมแสดงแฟ้ม FLI-FLC สามมิติสเตอริโอแบบพลิกหน้า
\FLI-FLC\EXAMPLE	แฟ้มภาพยนตร์คู่ในรูปแบบ FLI และ FLC สำหรับตาซ้ายและตาขวา สามตัวอย่าง
\FLI-FLC\MGA-DRV	ชุดโปรแกรมการควบคุมแผงวงจร Matrox MGA Millenium
\FLI-FLC\SDK-DOC	แฟ้มเอกสารที่เกี่ยวกับมาตรฐาน VESA การเรียกใช่คลังคำสั่ง SuperVGA kit รูปแบบแฟ้มข้อมูล FLI/FLC และมาตรฐานการเรียกใช้บริการ XMS
\FLI-FLC\PROGRAM	แฟ้มต่างๆของโปรแกรมแสดงแฟ้ม FLI-FLC สามมิติสเตอริโอแบบพลิกหน้า

ชื่อสารบบ	ความหมาย
\VTK\BINARY	ระบบ VTK ที่เติมส่วนเสริมแล้ว พร้อมตัวอย่าง
\VTK\EXAMPLE	แฟ้มตัวอย่างภาษา Tcl เพื่อการจินตทัศน์ภายใต้ระบบ VTK
\VTK\PROGRAM	แฟ้มโปรแกรม VTK ทั้งระบบที่ได้รับการเพิ่มส่วนเสริมแล้ว

ชื่อสารบบย่อยใต้สารบบ \VTK\PROGRAM\MARTINK\VTK	ความหมาย
...\TK4.1	แฟ้มโปรแกรมระบบ Tk รุ่นที่ 4.1 ที่ใช้งานร่วมกับ VTK ที่ได้รับการปรับปรุงส่วนเสริมแล้ว
...\TCL7.5	แฟ้มโปรแกรมระบบ Tcl รุ่นที่ 7.5 ที่ใช้งานร่วมกับ VTK ที่ได้รับการปรับปรุงส่วนเสริมแล้ว
...\VTK12	แฟ้มต้นฉบับของระบบ VTK, TK4.1 และ TCL7.5
...\VTK12SRC	แฟ้มโปรแกรมระบบ VTK รุ่น 1.2 ที่ได้รับการปรับปรุงส่วนเสริมแล้ว

ชื่อสารบบ	ความหมาย
\UTIL\LVIEWPRO	แฟ้มโปรแกรมการแสดงภาพรูปแบบต่างๆ
\UTIL\SIMULEYES	แฟ้มโปรแกรมวาดเส้นรหัสขาวและ การตั้งจอภาพให้ทำงานแบบสอดสลับของบริษัท StereoGraphics เพื่อใช้กับระบบแว่น SimulEyes พร้อมภาพสามมิติตัวอย่างแบบสอดสลับ
\UTIL\WLC	แฟ้มตัวโปรแกรมการวาดเส้นรหัสขาว ด้วยภาษา Visual BASIC

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค การเผยแพร่ผลงาน

ผลงานในโครงการสิ่งประดิษฐ์ได้รับการเผยแพร่ต่อสาธารณชนในงานนิทรรศการดังต่อไปนี้

1. งานจุฬาไฮเทค ณ ห้องนิทรรศการ ชั้น 6 ศูนย์การค้ามาบุญครอง พ.ศ. 2539
2. งานนิทรรศการจุฬาวิชาการ นิทรรศการทางวิชาการทางวิศวกรรมครั้งที่ 11 คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย พ.ศ. 2539
3. งานนิทรรศการของบริษัทออโตเดกส์ ณ โรงแรมแชงกรีลา พ.ศ. 2540

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย