การสังเคราะห์ภาพโดยใช้ คอมพิวเตอร์ กราฟฟิค 3 มิติ

Image Synthesis Using 3D-Computer Graphics

โดย

นาย วุฒิชัย มีโพธิ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาอุตสาหกรรมศาสตร์บัณฑิต

สาขาเทคโนโลยีคอมพิวเตอร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

ปีการศึกษา ๒๕๓๕

การสังเคราะห์ภาพ โดยใช้ คอมพิวเตอร์ กราฟฟิค 3 มิติ

Image Systhesis Using 3D-Computer Graphics

นาย วุฒิชัย มีโพธิ์ 34162228

อาจารย์ที่ปรึกษา

อาจารย์ เกษตร์ ศิริสันติสัมฤทธิ์

ปริญญานิพนธ์สำหรับปริญญาอุตสาหกรรมศาสตร์บัณฑิต

สาขา เทคโนโลยีคอมพิวเตอร์อุตสาหกรรม

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2535

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

สาขา เทคโนโลยีคอมพิวเตอร์อุตสาหกรรม

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

เรื่อง การสังเคราะห์ภาพ โดยใช้ คอมพิวเตอร์กราฟฟิค 3 มิติ

Image Synthesis Using 3D - Computer Graphics

ผู้จัดทำ

นาย วุฒิชัย มีโพธิ์ 34162228

....อาจารย์ที่ปรึกษา

(อ. เกษตร์ ศิริสันติสัมฤทธิ์)

บทคัดย่อ

ปริญญานิพนธ์นี้ กล่าวถึง การสังเคราะห์ภาพ ในลักษณะ ภาพกราฟฟิค 3 มิติด้วย หลักการของ คอมพิวเตอร์ กราฟฟิค กำหนดตำแหน่งของภาพ หรือ วัตถุ เมื่อมีการ เคลื่อนที่ ของวัตถุ จากแกน ของมิติ การคำนวณ เพื่อแสดงความ สัมพันธ์ ระหว่าง ทิศทาง จุดกำหนด แสง แสงสะท้อน ตำแหน่ง ของ วัตถุ และเงาที่ปรากฏ โดยการ เปลี่ยน ตำแหน่ง ของจุด มองแสง และ วัตถุได้ทุก ตำแหน่ง เน้นถึงการ สังเคราะห์ แสง และรูปร่าง ของวัตถุ ให้เห็นถึง ความตื้นลึก ของวัตถุ การประยุกต์ วิธีการที่เหมาะสม เพื่อให้เกิด ความ ต่อเนื่อง ของ ความโค้ง และ เว้า ของวัตถุ ได้เหมือนจริง

ABSTRACT

This thesis presents a procedure for producing realistic images, actually display the three-dimensional objects on a computer graphics screen. To specify the axis we show the relation of light source, reflectec light, shadow and object position. The goal is to find the shading and the development of algorithms, producing pictures as a continous function shading and a smooth appearance is obtained.

สารบัญ

เรื่อง			หน้า

บทนำ

คำพูดที่ว่า " ภาพหนึ่งภาพ อาจแทน คำพูด นับพันคำ " ยังคงเป็น สัจธรรม ที่ยังคงอยู่คู่กับ โลกใบ นี้อยู่ โดยเฉพาะอย่างยิ่งกับ คอมพิวเตอร์ และถ้าจะพูด ถึงภาพหนึ่งภาพในโลกของ คอม พิวเตอร์แล้ว นั่น หมายความถึง การสร้างภาพให้ปรากฏขึ้นมาบนจอ คอมพิวเตอร์ หรือ มอนิเตอร์ หรืออาจ ออกทาง hard copy เช่น เครื่องพิมพ์ (printter) เครื่อง พลอตภาพ (plotter) นั่นเอง แต่การที่จะทำให้ คอมพิวเตอร์ นำเสนอภาพ ออกมาแต่ละภาพนั้น ยังคงเป็นปัญหาอยู่ว่า จะ ไปเอาภาพมาจากไหน ซึ่งจะกล่าวไป มัน ก็ มี อยู่หลาย วิธี ซึ่งวิธีแรกนั้นก็โดยการใช้ อุปกรณ์ กวาดภาพ ยกตัวอย่างเช่น เครื่องกวาดภาพ (Scanner) กล้องถ่าย VDO ซึ่งต้องมี ส่วนฮาร์ด แวร์ช่วยอย่างมากแถม ราคาแพงหูดับตับไหม้ มาใช้ในการเก็บภาพเอาไว้ และวิธีต่อมาก็คือการใช้ Software ทางด้าน กราฟฟิกส์ ช่วยในการออกแบบ (CAD /CAM) สร้างโมเดลขึ้นมาก่อนแล้ว หลังจากนั้นก็ นำมาตกแต่งให้สีให้สัน ให้ภาพที่ได้ออกมามีความเหมือน จริงมากที่สุด และนั่นคือที่มา ของงานปริญญานิพนธ์ เล่มนี้

เราจะมาพูดถึงในส่วนที่ว่า ถ้าหากภาพที่เราต้องการนำเสนอ มันยัง คงเป็นอากาศ อยู่นั่นหมายความ ว่า มันเป็น จินตนาการ ของผู้ออกแบบเท่านั้น ยังมิเคยมีการสร้างตัว จริงออกมาเป็น ตัว เป็นตนขึ้นมาเลย ท่านก็คงจะหากล้องใด ๆ ในโลกมา ถ่ายภาพแห่ง จินตนาการ ได้อย่างแน่ นอน โดยเฉพาะ ความต้องการ ที่จะมองภาพในมุมต่าง ๆ ของวัตถุ ก็ทำไม่ได้ เรื่องนี้เรา จำเป็น ที่จะต้อง อาศัย ความสามารถของ คอมพิวเตอร์ มาช่วย งานแห่งจินตนาการของเรา ซึ่งการ ใช้ คอมพิวเตอร์ สร้างภาพนั้น ถ้าหากต้องการ ภาพที่เหมือนจริง เราหลีกเลี่ยงไม่ได้ ที่จะนำเอา ภาพ 3 มิติ มาพิจารณา นอกจากคุณจะมีตาเพียงตาเดียว ในทางปฏิบัติ นั้น การที่มีตาเพียงตาเดียว นั้นไม่สามารถที่จะมองเห็น ความลึกของวัตถุ หรือความเป็น 3 มิติ ได้ กล่าวคือ สิ่งที่มีชีวิตเกือบ จะทั้งหมดล้วนแต่มี ตา สองตา กันทั้งนั้น เมื่อมีตาสองข้างก็เกิด การมองแบบ Sterio ขึ้นมาซึ่ง การที่ตา สองข้างมองเห็นนั้นจะมีสมองส่วน หนึ่งทำหน้าที่ ในการถอดระหัส เพื่อส่งผลให้การมองเห็น ของสิ่งมีชีวิตนั้นมองเห็นวัตถุนั้นมีความลึกขึ้นมาในกรณี ที่ท่านมี ตาข้างเดียวตั้งแต่เกิด ท่านจะไม่มี ความรู้สึกถึงความลึก แต่ถ้าหากท่าน มีตาสอง ข้างแล้วลอง ปิดตา เอาไว้ข้างหนึ่งแล้วมองวัตถุ ท่านจะ ยังคงมีความรู้สึก ว่ามันยังคงมีความลึกอยู่ดี นั่นก็เป็นเพราะ สาเหตุ ที่ว่า จาก ประสบการณ์ ที่ผ่าน มาในชีวิต สมองของท่าน ได้ชดเชย ความรู้สึก ให้เอาเองว่าวัตถุนั้นมีความลึก

เมื่อ กล่าวถึง คอมพิวเตอร์ 3 มิติ แล้วเราคง จะต้องหา อะไร สักอย่างมา เพื่อใช้อ้างอิงถึง จุดต่าง ๆ ใน ระบบ3 มิติ โดยความเหมาะ สมและความง่ายในการ พิจารณาแล้วจะพบว่า

การใช้ คณิตศาสตร์ แขนง หนึ่ง ที่ เรียกว่า พีชคณิตเวกเตอร์ มาประยุกต์ใช้ในการอ้างถึงจุดต่าง ๆ ใน ระบบ 3 มิติ อาจกล่าวได้ด้วยว่า Vector ได้ถือกำเนิดมาเพื่อ 3 มิติ โดยเฉพาะก็ว่าได้

ในการกำหนดโมเดล ภาพให้ออกมาดูดีนั้น มันคงจะ ต้องไม่มีเพียงแค่การลากเส้นลากสาย ที่กำหนด ให้เท่านั้นหรอกนะครับ มันจะต้องมีการระบายสีกันบ้างหรือที่ ฝรั่งเขาเรียกว่า การทำ Shading นั่นแหละ จะเป็นการกำหนดสีของวัตถุซึ่งจะมีการกำหนด ค่าสัมประสิทธิ์การดูดกลืน (Transparent) ค่าสัมประสิทธิ์ การสะท้อนแสงแบบ เป็นระเบียบ (Specular) ค่าสัมประสิทธิ์ การสะท้อนแบบ ไม่เป็นระเบียบ (diffuse) และอื่น ๆ อีกมากมายซึ่งเราจะกล่าวกัน ต่อไป ในเรื่องของการทำ Sading อีกครั้งหนึ่งในรายละเอียด

เมื่อ พูดถึง การสร้างภาพ กราฟฟิกส์ 3 มิติ แล้วไม่ได้พูดถึง การติดตามรังสี (Ray Tracing) อันลือลั่น ก็คงจะเป็นการ ล้าสมัยพอสมควร เทคนิคการติดตามรังสีนั้นเป็นเทคโนโลยีล่า สุดของการ สังเคราะห์ภาพเลยทีเดียว ได้ถือกำเนิดขึ้นมาได้เวลาล่วงผ่านไปประมาณ 10 กว่าปีก็ว่า ได้ การติดตาม รังสี เป็นเทคโนโลยีที่ สามารถ สร้างภาพได้ใกล้ เคียง กับความจริง มาก ที่สุดเลย ก็ว่าได้ การติดตามรังสี เป็นวิธีการ ที่สลับ ซับซ้อนมาก ๆ เลยและถือได้ว่าสิ้นเปลืองเวลามากที่สุดใน การสร้างภาพ แต่สาเหตุที่ การติดตามรังสีไม่เสื่อมความนิยมลงไปตามกาลเวลานั้นก็เพราะความเหมือนจริง อย่างที่สุดของ การติดตามรังสี ซึ่งถ้าหากไม่บอกว่า ภาพ ๆ นี้คอมพิวเตอร์ สังเคราะห์ มาก็อาจทำให้ ผู้พบเห็นเดาไปว่าเป็นภาพถ่าย เลยเชียวแหละโดยมากแล้ว การติดตามรังสี จะนำไป ทำงานบนเครื่อง ที่มีความ สามารถสูง ๆ อาทิเช่นเครื่อง เวอร์ค สเตชั่น (Work Station) เครื่อง เมนเฟรม เป็นต้น แต่ในปัจจุบันนั้นความสามารถของ PC (Persnal Computer) หรือเครื่อง คอมพิวเตอร์ส่วน บุคคลนั้นมีความสามารถ มากขึ้นจึงพอที่จะนำเอา การติดตามรังสี มาใช้ใน เครื่อง ระดับนี้ได้บ้างแต่การทำงานก็คงไม่ทันการเท่ากับเครื่องระดับ สูง ๆ ได้

การทำปริญญานิพนธ์ครั้งนี้นั้นขอกล่าวนำ โดยย่อ และการนำมาประยุกต์ ใช้งานโดยจะเน้นถึงการการที่ เราสามารถนำเอาทฤษฎีชั้นสูงมาใช้งานได้ในเบื้องต้น เนื่องจาก ว่าถ้าหากเราจะเจาะลึกลง ไปให้มาก กว่านี้จำเป็น อย่างยิ่ง ที่จะต้องใช้เวลามากกว่านี้ เนื่องจาก การศึกษาในเรื่องของ ระบบ 3 มิตินั้น ยังมี ราย ละเอียด ปลีกย่อยอีกมากมาย และยังคงเป็นเทคโนโลยี ปิด อยู่ โดยเฉพาะอย่างยิ่ง ประเทศไทย ขอเน้น ประเทศไทย ยังไม่ค่อย ที่จะกล้าเล่นกับ การติดตามรังสี กันบ้างเลยสังเกตุ จากวารสาร การประชุมทาง วิชาการ ในแต่ละปีนั้น ยังไม่มี เรื่องราวที่เกี่ยวข้องกับ การติดตามรังสี มาปรากฏให้เห็นเลยแม้แต่น้อย จึงเป็นอุปสรรคอย่างหนึ่งในการหาข้อมูล ซึ่งเท่าที่จะหาได้ ก็เป็นหนังสือจาก ต่างประเทศ อีกอย่างหนึ่งนั้นก็ คือ คนไทยยังมอง ไม่เห็นว่า เทคโนโลยี ทางด้านนี้ นั้น สามารถ ทำอะไร ได้ บ้าง

จะว่าไปแล้ว หนังสือเกี่ยวกับ คอมพิวเตอร์ 3 มิติ ภาษาไทย นั้น ไม่ต้องไปหาเลย จะมีก็ไม่เกินสองเล่มซึ่งก็ ให้อะไรได้ ไม่มากเท่าไหร่ ส่วนมาก แล้ว การศึกษา ระบบ3 มิตินั้น จะเป็น การพานิช เสียมากกว่าสังเกตุจาก Software ที่เกี่ยวกับ กราฟฟิกส์นั่นจะ มีราคา แพงมากเลย ในการทำ ปริญญานิพนธ์ ครั้งนี้ก็ทำได้ เพียงแต่ส่วนที่จะ สามารถทำได้เท่านั้น

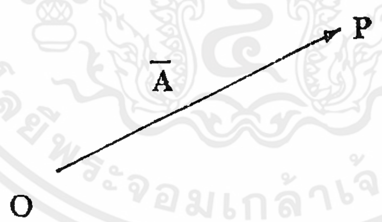
บทที่ 1
เวคเตอร์เบื้องต้น

แนวความคิด ใน เรื่องของ เวคเตอร์เริ่มขึ้น ในปี 1788 *โจเซฟ ลุยส์ ลากรองจ์* (1736 - 1813) ได้พิมพ์งานของเขาชื่อ กลศาสตร์วิเคราะห์ซึ่งแสดงถึง วิธีใช้การวิเคราะห์ใน การศึกษาวิชา กลศาสตร์ และ ต่อมา ในศตวรรษที่ 19 นักคณิตศาสตร์ ชาวไอริส ชื่อ *วิลเลี่ยม โอ แวน แฮมมิลตัน* (1805 - 1865) ได้เริ่ม ทฤษฎี ควอเทอร์เนี่ยน ซึ่งเป็นวิธีใหม่ที่จะช่วยให้เข้าใจทั้งพีชคณิต และฟิสิกส์ได้ดี และในเวลาต่อมา โจเซียห์ วิลเลียด กิบบส์ (1839 - 1903) และ โอลิเวอร์ เฮวีไซด์ (1850 - 1925) ได้รวมควอเทอร์เนียนวิเคราะห์ และ เรขาคณิตวิเคราะห์ เข้าด้วยกัน และ เรียกวิชาใหม่นี้ว่า เวคเตอร์พีชคณิต

ปริมาณ สเกล่าร์ และ ปริมาณเวคเตอร์

ปริมาณ สเกลล่าร์ (Soalar quantity) คือ ปริมาณที่มี แต่ขนาด แต่ไม่ มี ทิศทาง เช่น มวล ความร้อน เวลาอุณหภูมิ เป็นต้น

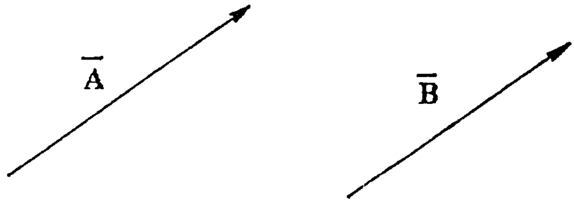
ปริมาณ เวคเตอร์ (Veotor quantity) คือ ปริมาณที่มี ทั้ง ขนาด และ ทิศทาง เช่น ความเร่ง แรง ความเร็ว เป็นต้น



รูป 1.1

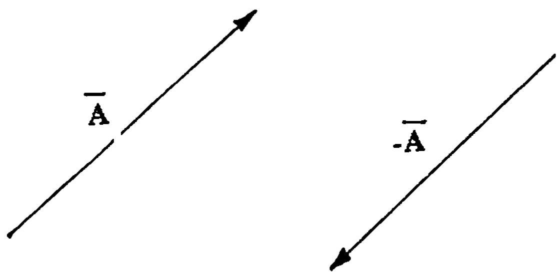
1.1 คุณสมบัติของ เวคเตอร์

1. เวคเตอร์ 2 เวคเตอร์ จะเท่ากัน ก็ต่อเมื่อ มีขนาด เท่ากัน และมีทิศทางไปทางเดียว กัน โดยไม่ต้องคำนึงถึงจุดกำเนิด



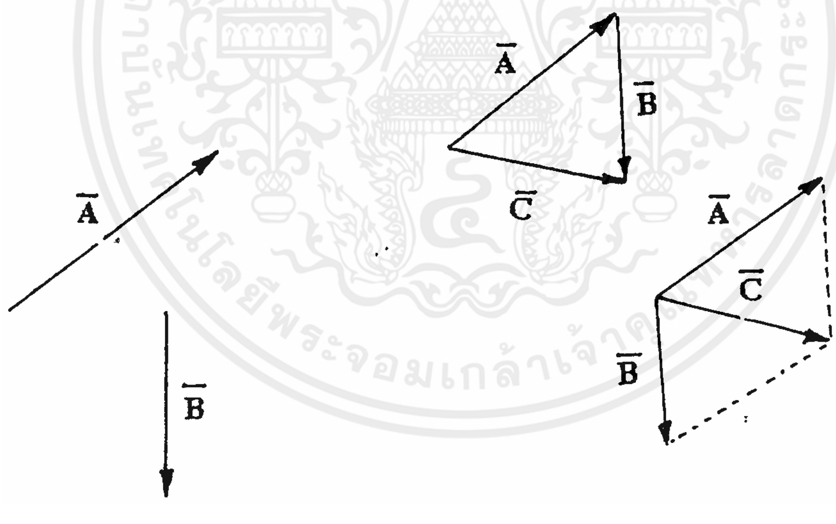
รูป 1.2

2. เวคเตอร์ที่ มี ขนาดเท่ากับ $\bar{A}$ แต่มีทิศทาง ตรงกันข้าม กับ $\bar{A}$ จะแทนด้วย $-\bar{A}$ เรียกว่า เวคเตอร์ลบ $\bar{A}$



รูป 1.3

3. ผลบวกของ เวคเตอร์ $\bar{A}$ และ $\bar{B}$ คือ $\bar{C}$ ซึ่งจะได้จาก วางจุดกำเนิดของ เวคเตอร์ $\bar{B}$ ลงบนจุด สิ้นสุด ของ เวคเตอร์ $\bar{A}$ แล้วเวคเตอร์ที่เชื่อมจากจุดกำเนิด ของ $\bar{A}$ กับจุดสิ้นสุดของ $\bar{B}$ เป็น เวคเตอร์ $\bar{C}$ เขียนว่า $\bar{C} = \bar{A} + \bar{B}$



รูป 1.4

4. ผลต่างของ เวคเตอร์ $\bar{A}$ และ $\bar{B}$ จะแทนด้วย $\bar{A} - \bar{B} = \bar{C}$ ซึ่งเมื่อ บวกกัน แล้วจะมีขนาดเท่ากับ $\bar{A}$ เสมอ หรือ $\bar{A} - \bar{B} = \bar{A} + (-\bar{B}) = \bar{C}$

ดังนั้น ถ้า $\bar{A} = \bar{B}$ แล้ว $\bar{A} - \bar{B}$ จะเป็น เวคเตอร์ที่มีขนาดและทิศทางไม่แน่นอน ซึ่งเวคเตอร์นี้เราเรียกว่า " เวคเตอร์ศูนย์ "

5. ผลคูณของ เวกเตอร์ $\bar{A}$ กับ Scalar m จะเป็นเวกเตอร์ $m\bar{A}$ ซึ่งมีขนาด เป็น m เท่าของเวกเตอร์ $\bar{A}$ และมีทิศทางไปทางเดียวกับ $\bar{A}$ ถ้า m เป็น บวก จะมีทิศทางตรงข้ามกับ $\bar{A}$ ถ้า m เป็น ลบ และจะเป็นเวกเตอร์ศูนย์ ถ้า m เป็น 0

1.2 กฎทางพีชคณิตของเวกเตอร์

ถ้า $\bar{A}, \bar{B}, \bar{C}$ เป็น เวกเตอร์ และ m , n เป็นสกาล่าร์แล้ว

1. $\bar{A} + \bar{B} = \bar{B} + \bar{A}$
2. $(\bar{A} + \bar{B}) + \bar{C} = \bar{A} + (\bar{B} + \bar{C})$
3. $m\bar{A} = \bar{A}m$
4. $m(n\bar{A}) = (mn)\bar{A}$
5. $(m + n)\bar{A} = m\bar{A} + n\bar{A}$
6. $m(\bar{A} + \bar{B}) = m\bar{A} + m\bar{B}$

1.3 เวกเตอร์หนึ่งหน่วย (Unit vector)

เวกเตอร์หนึ่งหน่วย คือ เวกเตอร์ใด ๆ ที่มี ขนาดหนึ่งหน่วย ดังนั้น ถ้า $\bar{A}$ เป็นเวกเตอร์ที่มีขนาด ไม่เป็น 0 หรือ $|\bar{A}| \neq 0$ แล้ว จะเป็น เวกเตอร์หนึ่งหน่วยทิศทางเดียว กับ $\bar{A}$ ให้ a เป็นเวกเตอร์ หนึ่งหน่วยที่มีทิศทางเดียวกับ $\bar{A}$ a และ $\bar{A} / |\bar{A}|$ จึงเท่ากัน และมี ทิศทางเดียวกัน ดังนั้น เวกเตอร์ทั้งสอง จะเป็นเวกเตอร์เดียวกัน นั่นคือ

$$\frac{\bar{A}}{|\bar{A}|} = a \text{ หรือ } \bar{A} = |\bar{A}|\,a$$

คุณสมบัติของ เวกเตอร์

1. เวกเตอร์ 2 เวกเตอร์ จะเท่ากัน ก็ต่อเมื่อ มีขนาด เท่ากัน และมีทิศทางไปทางเดียวกัน โดยไม่ต้องคำนึงถึงจุดกำเนิด

2. เวกเตอร์ที่ มี ขนาดเท่ากับ $\bar{A}$ แต่มีทิศทาง ตรงกันข้าม กับ $\bar{A}$ จะแทนด้วย $-\bar{A}$ เรียก ว่า เวกเตอร์ลบ A

3. ผลบวกของ เวกเตอร์ $\overline{A}$ และ $\overline{B}$ คือ $\overline{C}$ ซึ่งจะได้จาก วางจุดกำเนิดของ เวกเตอร์ $\overline{B}$ ลงบนจุดสิ้นสุดของ เวกเตอร์ $\overline{A}$ แล้วเวกเตอร์ที่เชื่อมจากจุดกำเนิด ของ $\overline{A}$ กับจุดสิ้นสุดของ $\overline{B}$ เป็นเวกเตอร์ $\overline{C}$ เขียนว่า $\overline{C} = \overline{A} + \overline{B}$

1.4 การคูณกันของเวกเตอร์

การคูณระหว่าง เวกเตอร์ กับ เวกเตอร์ ได้ผล 2 แบบคือ ผลคูณสเกลล่าร์ และ ผลคูณเวกเตอร์ ซึ่งจะกล่าวโดยละเอียดดังนี้

1.4.1 ผลคูณของ สกาล่าร์ กับ เวกเตอร์

ผลคูณสเกลล่าร์ระหว่าง $\overline{A}$ และ $\overline{B}$ ซึ่งแทนด้วยสัญญลักษณ์ $\overline{A} \cdot \overline{B}$ (อ่านว่า เวกเตอร์ A ดอต (Dot) เวกเตอร์ $\overline{B}$) กำหนดว่าจะเท่ากับขนาดของ $\overline{A}$ ขนาดของ $\overline{B}$ กับโคซายน์ของมุมระ หว่างเวกเตอร์ $\overline{A}$ และ เวกเตอร์ $\overline{B}$ ถ้าให้ แทนมุมระหว่างเวกเตอร์ทั้งสองจะเขียนแทน เป็น สมการได้ว่า

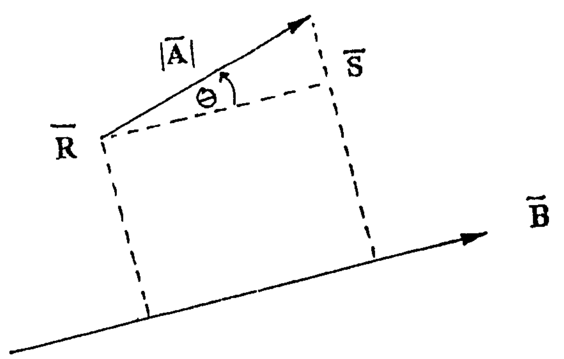
$$\overline{A} \cdot \overline{B} = |\overline{A}||\overline{B}| \cos\theta \quad ; \quad 0 < \theta < \pi$$

ดังนั้นจะเห็นว่าผลคูณสเกลล่าร์ระหว่าง 2 เวกเตอร์ ใด ๆ จะมีค่าเป็นสเกลล่าร์มิใช่เวกเตอร์

คุณสมบัติของผลคูณสเกลล่าร์

1. ผลคูณสเกลล่าร์จะมีคุณสมบัติ การสลับที่ $\overline{A} \cdot \overline{B} = \overline{B} \cdot \overline{A}$

2. ภาพฉายของ $\overline{A}$ บน $\overline{B}$ จะมีค่าเท่ากับผลคูณสเกลล่าร์ของ $\overline{A}$ กับ เวกเตอร์หนึ่งหน่วยที่มีทิศทาง เดียวกันกับ $\overline{B}$



รูป 1.5

ตามรูปถ้า θ คือมุมระหว่าง $\bar{A}$ และ $\bar{B}$ ภาพฉายของ $\bar{A}$ บน $\bar{B}$ = ระยะทาง PQ

$$PQ = RS$$
$$= |\bar{A}| \cos\theta$$
$$= |\bar{A}| \cdot |b| \cos\theta$$
$$= \bar{A} \cdot b$$

เมื่อ b คือเวกเตอร์หนึ่งหน่วยที่มีทิศทางเดียวกับ $\bar{B}$

3. ถ้า m เป็นปริมาณสเกลล่าร์ $m(\bar{A}\cdot\bar{B}) = m\bar{A}\cdot\bar{B} = \bar{A}\cdot m\bar{B}$

4. ผลคูณปริมาณสเกลล่าร์จะเป็นไปตามกฎการกระจาย

$$\bar{A}(\bar{B}+\bar{C}) = \bar{A}\cdot\bar{B} + \bar{A}\cdot\bar{C}$$

โดยจะพิสูจได้ถ้า ให้ a เวกเตอร์หนึ่งหน่วยที่มีทิศทางเดียวกันกับ A

PQ = ภาพฉายของ $\bar{B}$ บน $\bar{A}$

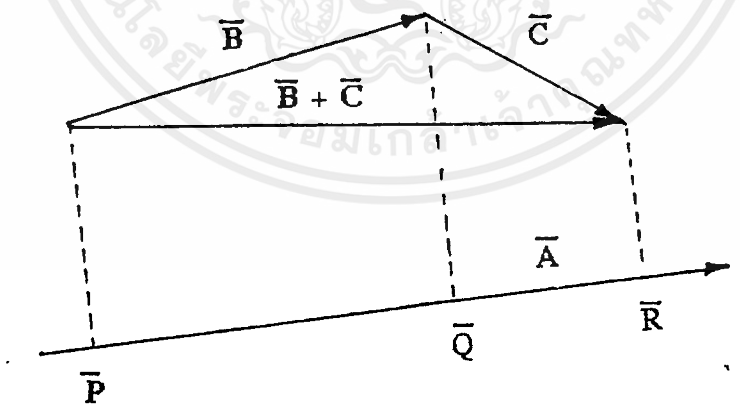
$= \bar{B}$ a

QR = ภาพฉายของ $\bar{C}$ บน $\bar{A}$

$= \bar{C}$ a

PR = ภาพฉายของ $(\bar{B}+\bar{C})$ บน $\bar{A}$

$= (\bar{B}+\bar{C})$ a



รูป 1.6 การพิสูจน์ กฎการกระจาย

จากรูปจะเห็นว่า PR = PQ + QR

$$(\bar{B}+\bar{C})\, a = \bar{B}\, a + \bar{C}\, a$$

คูณตลอดสมการด้วย $|\bar{A}|$ ซึ่งเป็นปริมาณสเกลล่าร์ จะได้จากคุณสมบัติข้อ 3 ว่า

$$(\bar{B}+\bar{C})\, |\bar{A}|\, a = \bar{B}\, |\bar{A}|\, a + \bar{C}\, |\bar{A}|\, a$$

และจากคุณสมบัติของเวคเตอร์หนึ่งหน่วย จะได้ว่า $|\bar{A}|\, a = \bar{A}$ ดังนั้น

$$(\bar{B} + \bar{C})\cdot\bar{A} = \bar{B}\cdot\bar{A} + \bar{C}\cdot\bar{A}$$

และเนื่องจากกฎการสลับที่ จะได้ว่า $\bar{A}\cdot(\bar{B}+\bar{C}) = \bar{A}\cdot\bar{B} + \bar{A}\cdot\bar{C}$

5. $i\cdot i = |\bar{I}|\cdot|\bar{I}| \cos 0 = 1$

ในทำนองเดียวกัน $j\cdot j = k\cdot k = 1$

แต่ $i\cdot j = |\bar{I}|\cdot|\bar{J}| \cos 90 = 0$

ในทำนองเดียวกัน $j\cdot k = k\cdot i = 0$

6. ถ้ากำหนดองค์ประกอบของเวคเตอร์ให้ จะหาผลคูณปริมาณสเกลล่าร์ของเวคเตอร์ได้โดยไม่ต้องทราบมุมระหว่างเวคเตอร์ทั้งสองได้ เช่น

$$\bar{A} = \bar{A}_1 i + \bar{A}_2 j + \bar{A}_3 k$$
$$\bar{B} = \bar{B}_1 i + \bar{B}_2 j + \bar{B}_3 k$$
$$\bar{A}\cdot\bar{B} = (\bar{A}_1 i + \bar{A}_2 j + \bar{A}_3 k)\cdot(\bar{B}_1 i + \bar{B}_2 j + \bar{B}_3 k)$$

โดยกฎการกระจายจะได้

$$\bar{A}\cdot\bar{B} = \bar{A}_1 i\,(\bar{B}_1 i + \bar{B}_2 j + \bar{B}_3 k) + \bar{A}_2 j\,(\bar{B}_1 i + \bar{B}_2 j + \bar{B}_3 k) + \bar{A}_3 k\,(\bar{B}_1 i + \bar{B}_2 j + \bar{B}_3 k)$$
$$= \bar{A}_1\bar{B}_1 + \bar{A}_2\bar{B}_2 + \bar{A}_3\bar{B}_3$$

จากนี้เราทราบว่า เมื่อ $\bar{A} = \bar{B}$ จะได้ $\bar{A}\cdot\bar{A} = A^2$

1.4.2 ผลคูณของ เวคเตอร์ กับ เวคเตอร์

ผลคูณเวคเตอร์ของ $\bar{A}$ และ $\bar{B}$ แทนด้วยสัญญลักษณ์ $\bar{A} \times \bar{B}$ จะเป็นเวคเตอร์ซึ่งจะเรียกว่า $\bar{C}$ ซึ่ง $\bar{C}$ จะเป็นเวคเตอร์ที่มีขนาดเท่ากับผลคูณของขนาด $\bar{A}$ ขนาด $\bar{B}$ และ ซายน์(sin) ของ มุมระหว่าง $\bar{A}$ กับ $\bar{B}$ ซึ่ง C จะมีทิศทางตั้งฉากกับ ทั้ง $\bar{A}$ และ $\bar{B}$ ที่จะทำให้ $\bar{A}$, $\bar{B}$,$\bar{C}$ เป็นระ บบมือขวา หรือจะเขียนเป็นสมการว่า

$C = \frac{1}{\sqrt{1+\omega^2 C^2}}$

$$\bar{A} \times \bar{B} = |\bar{A}|\,|\bar{B}| \sin\theta\, u \;;\; 0 < \theta < \pi$$

$= \frac{1}{1+\omega}$

เมื่อ θ คือมุมระหว่าง $\bar{A}$ และ $\bar{B}$

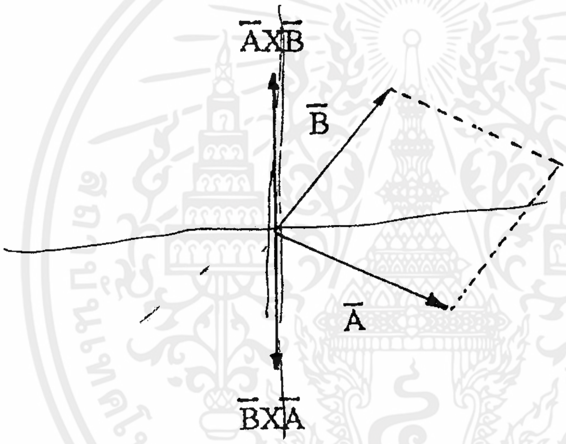
u คือเวกเตอร์หนึ่งหน่วยที่ตั้งฉากกับ $\bar{A}$ และ $\bar{B}$

จะเห็นว่า ถ้า $\bar{A} \,//\, \bar{B}$ แล้ว $\bar{A} \times \bar{B}$ = เวกเตอร์ 0

1.5 คุณสมบัติการคูณกัน ของ เวกเตอร์

1. $\bar{A}x\bar{B}$ และ $\bar{B}x\bar{A}$ จะมีขนาดเท่ากันแต่ มีทิศทาง ตรงข้ามกันดังนั้นกฎการสลับที่จึงใช้ไม่ได้ในผลคูณเวกเตอร์

$$\bar{A}x\bar{B} = -\bar{B}x\bar{A}$$



รูป 1.7

2. ถ้า m เป็นปริมาณสเกลล่าร์ แล้ว

$$m(\bar{A}x\bar{B}) = m\bar{A} \times \bar{B} = \bar{A} \times m\bar{B}$$

3. กฎการกระจายจะใช้ได้ในผลคูณเวกเตอร์

$$\bar{A}x(\bar{B}x\bar{C}) = \bar{A}x\bar{B} + \bar{A}x\bar{C}$$

4. ผลคูณเวกเตอร์หนึ่งหน่วย ในระบบพิกัดฉาก

ixj = k

jxk = i

kxi = j

jxi = -k

kxj = -i

ixk = -j

แต่ ixi = jxj = kxk = 0

5. ถ้าทราบองค์ประกอบของ เวกเตอร์ 2 เวกเตอร์ จะสามารถหาผลคูณของเวกเตอร์ 2 เวกเตอร์ได้เช่น

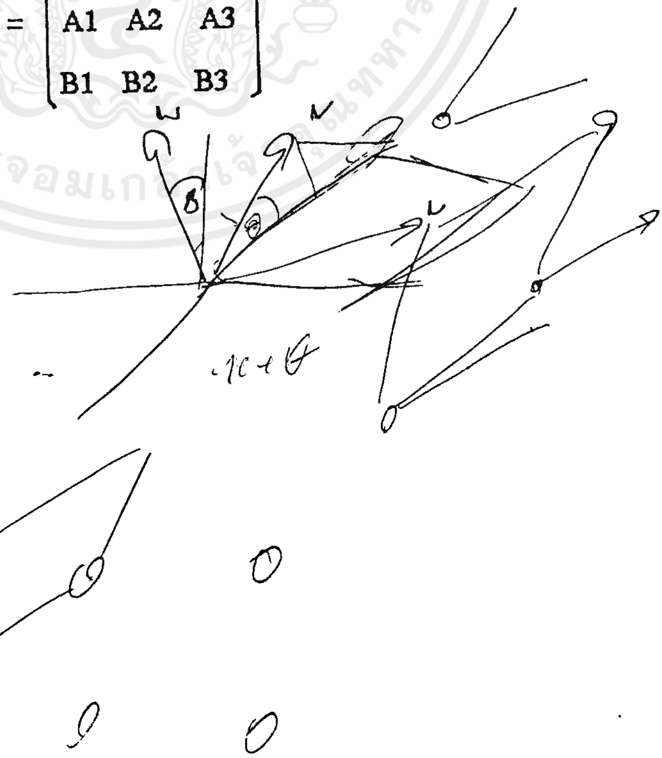
$$\bar{A} = A_1i + A_2j + A_3k \text{ และ } \bar{B} = B_1i + B_2j + B_3k$$

แล้วโดยกฎการกระจายจะได้ว่า

$$
\begin{aligned}
\bar{A}x\bar{B} &= (A_1i + A_2j + A_3k) \, x \, (B_1i + B_2j + B_3k) \\
&= A_1i \, x \, (B_1i + B_2j + B_3k) + A_2j \, x \, (B_1i + B_2j + B_3k) + \\
&\quad A_3k \, x \, (B_1i + B_2j + B_3k) \\
&= A_1B_2k - A_1B_3j - A_2B_1k \cancel{+ A_2B_3i + A_3B_1j - A_3B_2i} \\
&= (A_2B_3 - A_3B_2)i + (A_3B_1 - A_1B_3)j + (A_1B_2 - A_2B_1)k
\end{aligned}
$$

หรือเพื่อจำง่าย

$$\bar{A}x\bar{B} = \begin{bmatrix} \bar{i} & \bar{j} & \bar{k} \\ A1 & A2 & A3 \\ B1 & B2 & B3 \end{bmatrix}$$



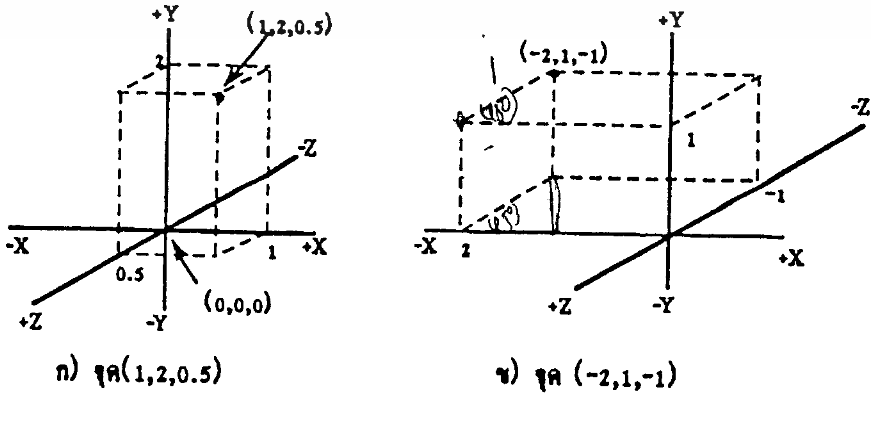
บทที่ 2

ระบบภาพ 3 มิติ

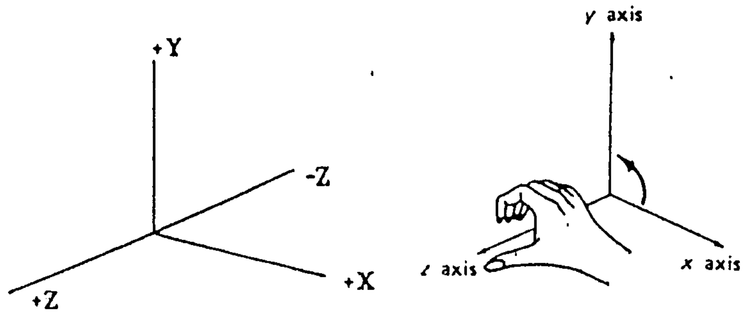
ในระบบ 2 มิติ มีแกน เพียง 2 แกนเท่านั้น คือ แกน X กับ แกน Y แต่ในระบบ 3 มิติ ต้องเพิ่ม แกน เข้าไปอีกหนึ่งแกน คือ แกน Z การกำหนดทิศทางของแกน Z มี 2 แบบ คือ แบบระบบ มือขวา และ แบบระบบ มือซ้าย

ในระบบมือขวา เราใช้นิ้วโป้งของมือขวาชี้ไปในทิศทางของแกน Z งอนิ้วที่เหลือทั้งสี่ นิ้ว ทิศทาง การหมุนของนิ้วทั้ง สี่ จะหมุนจากแกน X เข้าหา แกน Y ส่วนในระบบมือซ้าย เราใช้หัว นิ้วโป้ง ของมือซ้ายชี้ไปทาง แกน Z ทำให้นิ้วที่เหลือทั้ง สี่นิ้วจะชี้ไปทางจาก แกน X ไปยัง แกน Y โดยทั่วไป แล้วเรามักจะใช้ โคออร์ดิเนต ระบบมือขวา เช่น ในทางคณิตศาสตร์ หรือการใช้งาน ใน ระบบภูมิศาสตร์ แต่สำหรับในระบบคอมพิวเตอร์กราฟฟิกส์ มักจะนิยม ใช้ระบบ มือซ้าย ทั้งนี้เพราะ เราจะถือว่า ระนาบของ ~~จอมอนิเตอร์เป็นระนาบ XY (เป็น แกน X และ แกน Y)~~ และให้ระยะทาง ความลึก เข้าไป หลังจอภาพมีค่าเป็นบวก (ค่า โคออร์ดิเนต Z เป็น บวก) การกำหนดจุดในระบบ ระบบ 3 มิติ ต้องใช้จำนวน สาม จำนวน เพื่อเป็น การระบุ จุด นั้น ห่างจาก จุดกำเนิด (0,0,0) ไปตามแนว แกน X แกน Y และ แกน Z เป็นเท่าใด เช่นจุด (1,2,0.5) กำเนิดคือจุดที่ห่างจากจุด กำเนิด ไปในแนวแกน X หนึ่งหน่วย และ ห่างจากจุด กำเนิด ไปในแนว แกน Y สอง หน่วย และ ห่างจากจุดกำเนิด 0.5 หน่วยไปใน แนว แกน Z ดังตัวอย่างที่แสดงในรูป

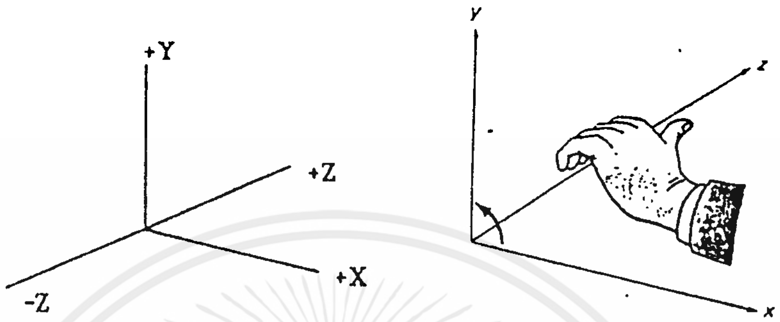
(x+ 2cos45, y + 2sin45)



รูป 2.1 ตัวอย่างตำแหน่งของจุดในระบบ 3 มิติ



ก) ระบบมือขวา



ข) ระบบมือซ้าย

2.1 เส้นตรง และ เวคเตอร์

เส้นตรงใน ระบบ 3 มิติ ยังคงกำหนดได้ด้วย จุดปลาย 2 จุด เช่นเดียวกันกับในระ บบ 2 มิติ เราใช้จุด 2 จุดนี้กำหนดความยาวและทิศทางของ เส้นตรง สมการ ของเส้นตรงจำเป็น ต้องใช้ สมการ สอง สมการ ประกอบกันซึ่งสามารถแสดงได้ดังนี้

$$\frac{Y - Y_1}{X - X_1} = \frac{Y_2 - Y_1}{X_2 - X_1}$$

$$\frac{Z - Z_1}{X - X_1} = \frac{Z_2 - Z_1}{X_2 - X_1} \qquad \text{.................(2.1)}$$

โดยที่จุด (X_1,Y_1,Z_1) และ (X_2,Y_2,Z_2) คือจุดที่อยู่บนเส้นตรงเดียวกัน ตัวอย่างเช่น มีจุด (1,2,1) และ (0,1,2) อยู่บนเส้นตรงเดียวกัน สมการ ของ เส้นตรงนี้หาได้โดยแทนค่าของ $X_1,Y_1,Z_1,X_2,Y_2,$ และ Z_2 ลงในสมการ (2.1)

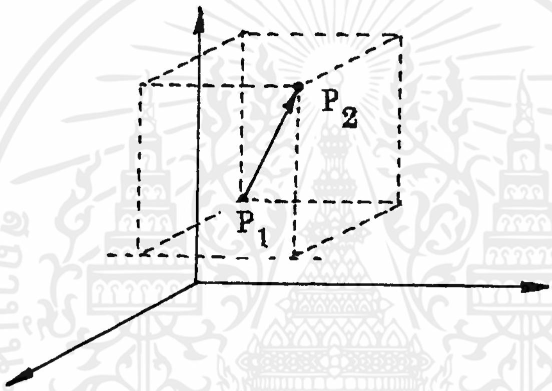
$$\frac{Y - 2}{X - 1} = \frac{1 - 2}{0 - 1}$$

$$\frac{Z - 2}{X - 1} = \frac{2 - 1}{0 - 1}$$

ดังนั้นสมการเส้นตรงในที่นี้ก็คือ

$$Y = X + 1$$
$$Z = -X + 3 \qquad \text{.....................(2.2)}$$

สำหรับเวคเตอร์ในระบบ 3 มิติก็คล้ายกับส่วนของเส้นตรง คือ สามารถกำหนดได้ ด้วยจุด 2 จุด ในระบบ แต่เวคเตอร์ต่างกับส่วนเส้นตรงคือ เวคเตอร์ที่มีทิศทาง โดยจุดหนึ่ง เป็น จุดเริ่ม ต้น ของเวคเตอร์ และอีกจุดหนึ่งเป็นจุดปลาย ตัวอย่างเช่น ถ้าเรามีเวคเตอร์ที่มีจุดเริ่มต้นคือ $P_1(X_1,Y_1,Z_1)$ และ มีจุดปลายคือ $P_2(X_2,Y_2,Z_2)$ เวคเตอร์นี้จะมีทิศทางดังแสดงให้เห็น ในรูป 2.3 สมการสามารถหาได้จากสมการ



รูป 2.3 ตัวอย่างเวคเตอร์ในระบบ 3 มิติ

ขนาดของเวคเตอร์ = $((X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2)^{1/2}$ โดย ทั่วไปแล้ว เรามัก จะสนใจเฉพาะทิศทางของ เวคเตอร์เท่านั้น ซึ่งเราสามารถแทนทิศทางของเวคเตอร์ ได้ด้วย เมตริกซ์ [x y z] ค่าของ x,y และ z เป็นอัตราส่วนที่จะแสดงให้เห็นว่าเวคเตอร์นั้นหันไปในทิศทางในแนว แกน X แกน Y และ แกน Z มากน้อยเท่าใดตามลำดับสำหรับ ในระบบคอมพิวเตอร์กราฟฟิกส์เรามักจะใช้ เมตริกซ์ [x y z] แทนทิศทางนั้นไปเลยถ้าจุดตั้ง ต้นของ เวคเตอร์ก็คือ $P_1(X_1,Y_1,Z_1)$ และจุดปลายของเวคเตอร์ก็คือ $P_2(X_2,Y_2,Z_2)$ เรา สามารถหาทิศทางของเวคเตอร์นี้ได้ โดยที่

$$x = X_2 - X_1$$
$$y = Y_2 - Y_1$$
$$z = Z_2 - Z_1$$

ยกตัวอย่างเช่น จุด P1 คือจุด (5,0,2) และจุด P_2 คือ (4,3,3) ดังนั้น

$$x = 4 - 5 = -1$$

$$y = 3 - 0 = 3$$

$$z = 3 - 2 = 1$$

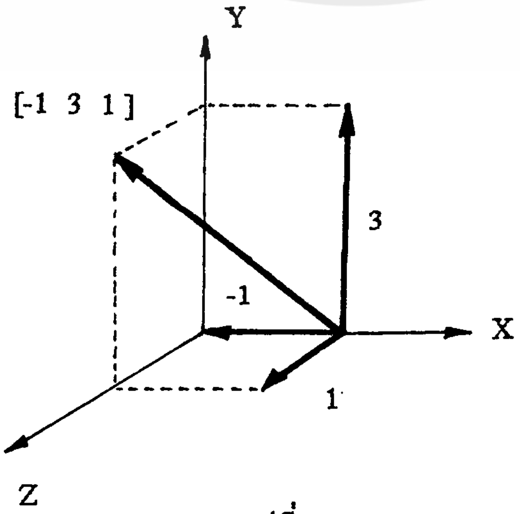
เวกเตอร์นี้จะหันไปในแนว แกน X แนว แกน Y แนว แกน Z ด้วยอัตราส่วน -1 : 3 : 1 ตาม ลำดับซึ่งจะสามารถแสดงทิศทางของเวกเตอร์ได้ดังในรูป 2.4 ซึ่งเราไม่สนใจว่าขนาด และ จุดตั้งต้น ของ เวกเตอร์ ดังนั้นเราอาจใช้เวกเตอร์ที่มีขนาดเล็กกว่า หรือ ยาว กว่า ได้ ขอให้เป็น เวกเตอร์ที่มีทิศทางเดียวกันก็พอ นั่นก็คือเวกเตอร์ที่มีขนาดของ [x y z] สามารถเปลี่ยนแปลง ได้ด้วย อัตราส่วน ที่เท่ากัน เช่นเวกเตอร์ [-1 3 1] จะมี ทิศทางเดียวกันกับ [-2 6 2] หรือ [-1/3 1 1/3] แต่เพื่อให้สอดคล้องกัน เรามักจะใช้เวกเตอร์ที่มีขนาดหนึ่งหน่วย (เรียกสั้น ๆ ว่า Unit Veotor) เพื่อกำหนดทิศทาง เวกเตอร์ หนึ่งหน่วยที่ มีทิศทางตรงกันกับ เวกเตอร์ที่เกิดจากจุด $P_1(X_1,Y_1,Z_1)$ และ $P_2(X_2,Y_2,Z_2)$ คือ [X Y Z] โดยที่

$$X = (X_2 - X_1) / S$$

$$Y = (Y_2 - Y_1) / S$$

$$Z = (Z_2 - Z_1) / S$$

โดยที่ S คือ ขนาดของเวกเตอร์ $= [(X_2-X_1)^2 + (Y_2-Y_1)^2 + (Z_2-Z_1)^2]$



รูปที่ 2.4

ตัวอย่างเช่น ถ้าต้องการหา เวคเตอร์ หนึ่งหน่วย ที่มีทิศทางเดียวกันกับ เวคเตอร์ [-1 3 1] ต้องทราบขององเวคเตอร์ก่อน

$$S = [(-1)^2 + (3)^2 + (1)^2]^{1/2}$$
$$= 11^{1/2}$$
$$= 3.32$$

ดังนั้นเวคเตอร์หนึ่งหน่วย คือ [-1/3.32 3/3.32 1/3.32]

= [-0.30 0.90 0.30]

ระนาบ

ระนาบภาพ 3 มิติ นอกเหนือจากจุดและเส้นตรงแล้ว เรายังต้อง ยุ่งเกี่ยวอยู่ กับระนาบ อีก ด้วย ระนาบ เปรียบได้กับ แผ่นกระดาษเรียบที่ไม่มีความหนา แต่มีขนาดใหญ่มาก ๆ ไม่จำกัดขอบ เขต ตั้งอยู่ ใน ระบบ โคออร์ดิเนต และเช่นเดียวกันกับ เส้นตรง ระนาบ มีสมการในการกำหนดหรือ บ่งชี้ระนาบนั้น ๆ สมการระนาบมีรูปแบบดังนี้

$$Ax + By + Cz + D = 0 \qquad ....(2.7)$$

เมื่อ A ,B และ C เป็นค่าคงที่ และ (x,y,z) คือจุดใด ๆ ที่อยู่บนระนาบ ถ้าเรา ทราบ 3 จุด ที่อยู่บนระนาบเดียวกัน คือ $P_1(X_1,Y_1,Z_1)$,$P_2(X_2,Y_2,Z_2)$ และ $P_3(X_3,Y_3,Z_3)$ เราสามารถหาสมการระนาบได้โดยการแทนค่าลงในสมการ ดังนี้

$$Ax_1 + By_1 + Cz_1 + D = 0 \qquad ....(2.8)$$
$$Ax_2 + By_2 + Cz_2 + D = 0 \qquad ....(2.9)$$
$$Ax_3 + By_3 + Cz_3 + D = 0 \qquad ....(2.10)$$

จากสมการ (2.8) ถึง (2.10) เราสามารถแก้สมการเพื่อหาค่าของ A , B และ C ได้ คือ

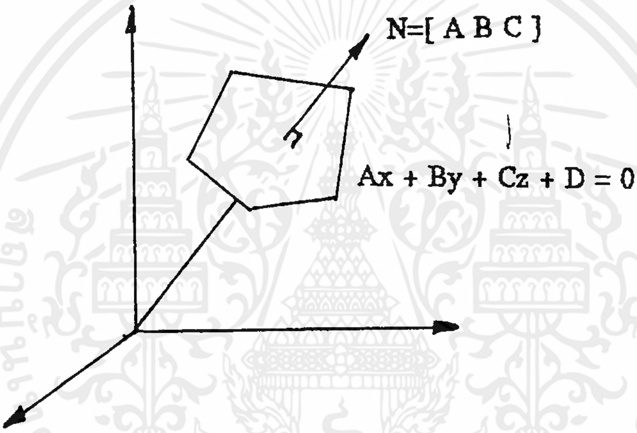
$$A = Y_1(Z_2-Z_3) + Y_2(Z_3-Z_1) + Y_3(Z_1-Z_2)$$
$$B = Z_1(X_2-X_3) + Z_2(X_3-X_1) + Z_3(X_1-X_2)$$

$$C = X_1(Y_2-Y_3) + X_2(Y_3-Y_1) + X_3(Y_1-Y_2)$$

$$D = -X_1(Y_2Z_3-Y_3Z_2) - X_2(Y_3Z_1-Y_1Z_3) - X_3(Y_1Z_2-Y_2Z_1) \quad \text{..........(2.11)}$$

การจัดเรียงตัวของระนาบในระบบโคออร์ดิเนต สามารถกำหนด ได้ด้วย เวกเตอร์ หนึ่งเวกเตอร์ ซึ่งเป็น เวกเตอร์มีทิศทาง ตั้งฉากกับพื้นผิวของของระนาบ เราเรียกเวกเตอร์นี้ว่า เวกเตอร์ ตั้งฉาก (Normal Vector) ดังแสดงในรูป 2.3 ถ้าสมการของระนาบคือ

$Ax + By + Cz + D = 0$ เวกเตอร์ตั้งฉากของระนาบนี้คือ [A B C]



รูป 2.5 เวกเตอร์ตั้งฉาก N

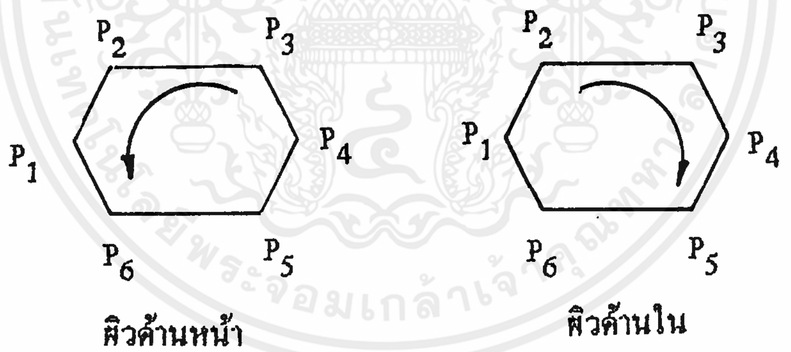
เวกเตอร์ตั้งฉากของระนาบเป็นสิ่งที่ บ่งบอกทิศทางของระนาบได้ด้วยพื้นผิวของระนาบ ทุกระนาบ มีอยู่ สองด้าน คือ ผิวด้าน ใน และ ผิวด้านนอก เวกเตอร์ตั้งฉากจะพุ่งเข้าหา พื้น ผิวระนาบด้านใน และจะ พุ่งออกจากพื้นผิวของระนาบด้านนอก พื้นผิวของระนาบที่เห็นในรูปเป็น พื้นผิวด้านนอก การกำหนดทิศทางของระนาบทำให้เราสามารถทราบได้ว่าจุดต่าง ๆ ในระบบเป็น จุดที่อยู่ฝั่งด้าน ใน หรือ อยู่ที่ฝั่งด้านนอก ของระนาบ จุด (x y z) ใด ๆ ที่อยู่ด้านนอกของระ นาบ จะทำให้อสมการเป็นจริง

$$Ax + By + Cz + D > 0 \quad \text{.............(2.12)}$$

และในทำนองเดียวกัน จุด (x y z) ใด ๆ ที่อยู่ด้านในระนาบ จะทำให้อสมการ (2.13) เป็นจริง

$$Ax + By + Cz + D < 0 \qquad ....(2.13)$$

สำหรับรูปหลายเหลี่ยมในระบบมิติต้องวางตัวอยู่บนระนาบใดระนาบหนึ่ง รูปหลายเหลี่ยม จึงมีผิว ด้านใน ผิวด้านนอก ด้วยเช่นกัน การกำหนดรูป หลายเหลี่ยม สามารถกำหนดได้โดยการกำหนด จุดยอดทุกจุดของรูปหลายเหลี่ยมตามลำดับ และลากเส้นจากจุดแรกไปยังจุดสุดท้าย ตามลำดับ ถ้าเรามอง จากจุด นอกระนาบ ของรูป หลายเหลี่ยม แล้วสามารถมองเห็น จุดยอดต่าง ๆ ของรูปหลายเหลี่ยม เรียงกัน ไป ตามทิศทางทวนเข็มนาฬิกา เราจะถือว่าผิวที่มองเห็นนั้นจะเป็นผิวด้านนอก และใน ทางตรงข้าม ถ้ามอง เห็น จุดยอดต่าง ๆ นั้นเรียงกันไป ในทิศทางตามเข็มนาฬิกา เราจะมองเห็น ผิว ด้านใน ของรูป หลาย เหลี่ยม ดังรูป 2.6



รูป 2.6 ตัวอย่างพื้นผิวของรูปหลายเหลี่ยม

การที่เรากำหนดพื้นผิวด้าน ใน ด้านนอกของระนาบ หรือรูปหลายเหลี่ยม ก็เพื่อ ประโยชน์ในการ สร้างภาพ โดยทั่วไปแล้วเรามักจะตัดภาพในส่วนที่เป็นผิวด้านในออก (Back face Removal) ทั้งนี้ ก็เพื่อไม่ให้ภาพที่ออกมาดู สับสนมาก เกินไปและ ได้ภาพที่มี ความเหมือนจริง มากที่สุด

2.2 การแปลงในระบบ 3 มิติ

การแปลงในระบบ 3 มิติมีความคล้ายคลึงกับการแปลงในระบบ 2 มิติ เรายังคงใช้ เมตริกส์ ช่วยในการแปลงใน ระบบ 3 มิติ แต่จะต้องเพิ่มขนาดของ เมตริกส์ ไว้สำหรับ โคออร์ดิเนต Z ดังนั้นจุด ในระบบ 3 มิติ ต้องแทนด้วย เมตริกส์ ขนาด 1 x 3 จุด (x,y,z) จะถูกแทนด้วย [X Y Z]

ในระบบ 2 มิติเรามีเมตริกส์การแปลงแบบเสกล ขนาด 2x2 มีรูปแบบเป็น

$$S = \begin{bmatrix} Sx & 0 \\ 0 & Sy \end{bmatrix} \qquad \text{..............(2.14)}$$

และถ้าเป็นโฮโมจีเนียสโคออร์ดิเนตเมตริกส์ มีรูปแบบเป็น

$$S = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{..............(2.15)}$$

โดยที่ Sx และ Sy ก็คือเสกลแฟคเตอร์สำหรับโคออร์ดิเนต X และ Y ตามลำดับ

ในระบบ 3 มิติ การแปลแบบเสกลจะกลายเป็นเมตริกส์ ขนาด 3 x 3 โดยมีรูปแบบเป็น ดังนี้

$$S = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & Sz \end{bmatrix} \qquad \text{..............(2.16)}$$

และถ้าเป็นโฮโมจีเนียสโคออร์ดิเนตเมตริกส์ จะมีรูปแบบเป็นเมตริกส์ ขนาด 4 x 4 ดังนี้

$$S = \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{..............(2.17)}$$

โดยที่ Sx Sy และ Sz เป็น สเกลแฟกเตอร์ สำหรับโคออร์ดิเนต X Y และ Z ตามลำดับ การแปลงจุดต่าง ๆ ทำได้โดย นำเมตริกส์ การแปลงไป คูณกับจุดนั้น เหมือนกับกรณีของ การแปลงแบบ ระบบ 2 มิติ

$$P_2 = [\,X_2\ Y_2\ Z_2\,]$$

$$= P_1 T$$

$$= [\,X_1\ Y_1\ Z_1\ W\,] \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= [\,SxX_1\ \ SyY_1\ \ SzZ_1\ \ W\,] \quad \text{.............(2.18)}$$

ดังนั้น

$$X_2 = SxX_1$$
$$Y_2 = SyY_1$$
$$Z_2 = SzZ_1 \quad \text{.............(2.19)}$$

สำหรับการแปลงแบบย้าย เมตริกส์ การแปลงก็คือ

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ Tx & Ty & Tz & 1 \end{bmatrix} \quad \text{.............(2.20)}$$

เมื่อ Tx Ty และ Tz คือระยะที่ จะต้อง เลื่อนไปตาม โคออร์ดิเนต X Y และ Z ตาม ลำดับ ในทำนอง เดียวกันกับระบบ 2 มิติ การ สเกล วัตถุใด ๆ ที่ไม่ได้อยู่ที่จุดกำเนิด ต้องย้าย วัตถุนั้นไปยังจุดกำเนิดเสียก่อน แล้วจึงทำการแปลงแบบ สเกล จากนั้นจึงย้ายวัตถุไปยังตำแหน่ง เดิม

2.3 การแปลงแบบหมุน

ในระบบ 2 มิติ การหมุนรอบจุดกำเนิด (0,0) มีเมตริกส์การแปลง คือ

$$R = \begin{bmatrix} \cos(a) & \sin(a) \\ \sin(a) & \cos(a) \end{bmatrix} \qquad \text{.............(2.22)}$$

แต่ในระบบ 3 มิติ การแปลงแบบ หมุนมิได้เป็น การหมุนรอบจุดใดจุดหนึ่ง แต่เป็นการ แปลงหมุน รอบแกน แกนใด แกนหนึ่ง ถ้าเราเปลี่ยนเมตริกส์การแปลง ในสมการ (2.22) ให้ใช้ ได้ใน ระบบ 3 มิติ เมตริกส์ที่ได้จะเป็นเมตริกส์การแปลงแบบหมุนรอบแกน Z เพราะว่าการหมุน แบบนี้ ค่าของ โคออร์ดิเนต Z คงที่ไม่เปลี่ยนแปลง เราเขียน เมตริกส์นี้ให้อยู่ในรูปของเมตริกส์ การแปลง แบบ โฮโมจิเนียส โคออร์ดิเนตเมตริกส์ คือ

$$Rz = \begin{bmatrix} \cos(a) & \sin(a) & 0 & 0 \\ -\sin(a) & \cos(a) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{.........(2.23)}$$

การหมุนแบบนี้ท่านคงคิดว่าเป็นการหมุนวัตถุไปรอบ ๆ แกนที่หยุดนิ่ง หรืออาจคิดว่าแกนหมุน เคลื่อน ที่ไปรอบ ๆ วัตถุที่อยู่นิ่ง ๆ กับที่ก็ได้ ความ แตกต่าง ของการแปลความหมายของการหมุน ก็คือทิศทาง ของการหมุนไปเท่านั้น การหมุน วัตถุไปรอบ ๆ แกนที่อยู่กับที่ ในทิศทางทวนเข็มนาฬิกา ก็คือการหมุนแกนไปรอบ ๆ วัตถุที่หยุดนิ่งในทิศทางตามเข็มนาฬิกา ถ้าหมุนจุด $P_1(x_1, y_1\ z_1)$ รอบแกน Z เป็นมุม a จะได้จุด $P_2(x_2, y_2, z_2)$ คือ

$$X_2 = X_1\cos(a) - Y_1\sin(a)$$
$$Y_2 = X_1\sin(a) + Y_1\cos(a)$$
$$Z_2 = Z_1 \qquad \text{..........(2.24)}$$

สำหรับการหมุนรอบ ๆ แกน X และแกน Y ก็มีสมการคล้าย ๆ กันกับสมการ (2.24) เมตริกส์ การแปลง แบบหมุนรอบแกน X คือ

$$Rx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(a) & \sin(a) & 0 \\ 0 & -\sin(a) & \cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{.........(2.25)}$$

ถ้าหมุนจุด $P_1(x_1 ,y_1\ z_1)$ รอบแกน X เป็นมุม a จะได้จุด $P_2(x_2 ,y_2 ,z_2)$ คือ

$$X_2 = X_1$$
$$Y_2 = Y_1\cos(a) - Z_1\sin(a)$$
$$Z_2 = Y_1\sin(a) + Z_1\cos(a) \qquad \text{..........(2.26)}$$

เมตริกส์ การแปลง แบบหมุนรอบแกน Y คือ

$$Ry = \begin{bmatrix} \cos(a) & 0 & -\sin(a) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(a) & 0 & \cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{.........(2.27)}$$

ถ้าหมุนจุด $P_1(x_1 ,y_1\ z_1)$ รอบแกน Y เป็นมุม a จะได้จุด $P_2(x_2 ,y_2 ,z_2)$ คือ

$$X2 = Z1\sin(a) + X1\cos(a)$$
$$Y2 = Y1$$
$$Z2 = Z1\cos(a) - X1\sin(a) \qquad \text{..........(2.28)}$$

2.4 การหมุนรอบแกนใด ๆ

การแปลงแบบหมุนที่กล่าวมาแล้วนั้น เป็นการหมุนรอบโคออร์ดิเนต ทั้ง สาม เท่านั้น อย่างไรก็ตามเราสามารถที่จะหมุนรอบแกนใด ๆ ในระบบ นอกเหนือจากแกนทั้ง สามแล้ว ได้ด้วยเช่นกัน โดยปรกติ แล้วเส้นตรงใน ระบบสามารถ เป็นแกนหมุน ได้ซึ่งต่อไป เราจะหา เมตริกส์ การแปลงแบบ หมุนรอบ แกนใด ๆในระบบ 3 มิติ โดยเริ่มจากการ ย้ายจุดกำเนิด

ไปอยู่บนเส้นตรงหรือแกนหมุนนั้น จากนั้น จะหมุนไปรอบแกน X และแกน Y เพื่อจัดให้การหมุนนั้นมีทิศทางเดียวกันกับแกน Z ดังนั้น การหมุน รอบแกน Z จึงเป็น การหมุนรอบแกนนั้นด้วย และท้ายสุดทำการแปลงกลับ เพื่อให้แกน หมุนกลับ ไปในแนวเดิม และย้ายจุดกำเนิดให้ ไปอยู่ในแนวเดิม

2.5 การกำหนดเส้นตรงในระบบ โคออร์ดิเนต

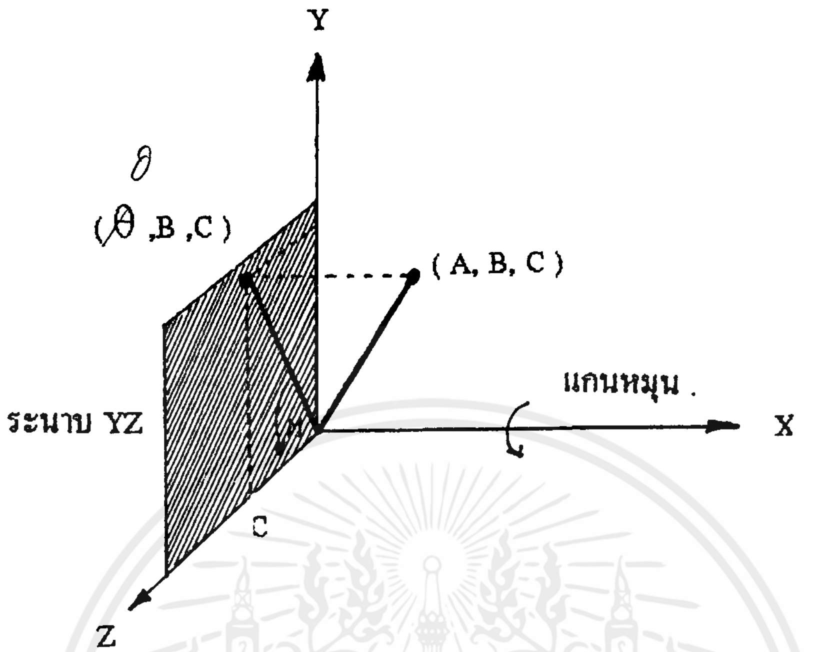
สามารถกำหนดได้ด้วยจุดหนึ่งจุดบนเส้นตรงนั้น กับเวคเตอร์ที่มีทิศทางเดียวกันกับเส้นตรงนั้น ถ้าจุด P1(x1,y1,z1) คือจุดบนเส้นตรงที่เป็นแกนหมุน และ [A B C] คือเวคเตอร์ ในแนวแกนของเส้นตรงนั้น เราสามารถหา เมตริกส์ การแปลงแบบหมุน รอบแกนนี้ได้ โดย เริ่มจากการสร้างเมตริกส์การแปลงแบบย้ายเพื่อย้ายจุดกำเนิดไปอยู่บนเส้นตรงนี้ เมตริกส์นี้คือ

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ X_1 & Y_1 & Z_1 & 1 \end{bmatrix} \qquad ....(2.29)$$

หลังจากการแปลงด้วยเมตริกส์นี้แล้ว จุด P จะย้ายไปอยู่ที่จุดกำเนิด หลังจากการแปลงแบบหมุน สำเร็จเสร็จสิ้นแล้ว เราต้องการ อินเวอร์สเมตริกส์ ของเมตริกส์ T เพื่อแปลง จุด P กลับ ไปยัง ตำแหน่ง เดิม ก่อนย้าย เมตริกส์นี้คือ

$$T\text{-}1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -X_1 & -Y_1 & -Z_1 & 1 \end{bmatrix} \qquad ....(2.30)$$

ลำดับต่อไปคือการหมุนเส้นตรงรอบแกน X จนกระทั่งเส้นตรงจะอยู่บน ระนาบ XY เพื่อให้ เข้าใจการหามุมของการหมุน เราจะฉายเงาของเส้นตรงให้เกิดขึ้นบนระนาบ XY สมมุติว่าเรามี ส่วนของเส้นตรงนี้ ซึ่งอยู่ระหว่างจุด (0,0,0) และจุด (A,B,C) เราจะได้เงาของส่วน ของเส้นตรงนี้ อยู่บนระนาบ YZ เป็นเส้นตรงที่อยู่ระหว่างจุด (0,0,0) และ (0,B,C) ดูรูป (2.7)



รูป 2.7 เงาของส่วนของเส้นตรงบนระนาบ YZ

การหมุนเส้นตรงนี้ให้มาอยู่บนระนาบ XZ เงาของเส้นตรงจะเคลื่อนมาทับแกน Z พอดี นั่นคือมุม ที่ใช้ในการหมุนคือ M ดังแสดงในรูป 5.9 ถ้า V คือความยาวของเงาบนระนาบ YZ ดังนั้น

$$V = (B^2 + C^2)^{1/2} \qquad \text{.........(2.31)}$$

$$\sin(M) = (B / V) \qquad \text{.........(2.32)}$$

$$\cos(M) = (C / V) \qquad \text{.........(2.33)}$$

เมตริกส์การแปลงแบบหมุนรอบแกน X คือ

$$Rx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(M) & \sin(M) & 0 \\ 0 & -\sin(M) & \cos(M) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

เมื่อแทนค่าในสมการ (2.32) และ (2.33) ลงในสมการ ข้างบน จะได้

$$Rx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C/V & B/V & 0 \\ 0 & -B/V & C/V & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad ....(2.35)$$

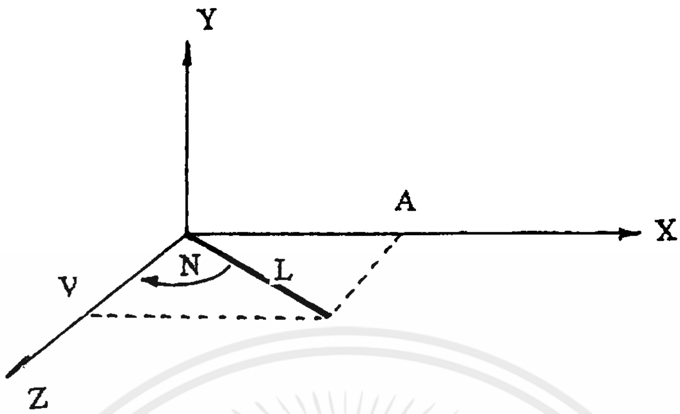
สำหรับการแปลงกลับของเมตริกซ์ในสมการ (2.35) คือการหมุนด้วยมุมเท่าเดิม แต่ในทิศทาง ตรงกันข้าม ทำให้ได้อินเวอร์สเมตริกส์ คือ

$$Rx\text{-}1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C/V & -B/V & 0 \\ 0 & B/V & C/V & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad ....(2.36)$$

หลังจากที่หมุนเส้นตรงไปบนระนาบ XZ แล้ว เราจะหมุนเส้นตรงนี้รอบแกน Y จนกระทั่งเส้น ตรงนี้ทับแกน Z พอดี ดังแสดงในรูป 5.10 ซึ่งมุมที่ใช้ในการหมุนคือ N ถ้าความยาวของ เส้นตรง นี้คือ L ดังนั้น

$$L = (A^2 + V^2)^{1/2}$$
$$= (A^2 + B^2 + C^2)^{1/2}$$

$$\sin(N) = A / L$$
$$\cos(N) = V / L$$



รูป 28 การหมุนส่วนของแกนเส้นตรงให้ไปกับแกน Z

เมตริกส์การแปลงแบบหมุนรอบแกน Y คือ

$$Ry = \begin{bmatrix} \cos(N) & 0 & -\sin(N) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(N) & 0 & \cos(N) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} V/L & 0 & A/L & 0 \\ 0 & 1 & 0 & 0 \\ A/L & 0 & V/L & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{........(2.40)}$$

และทำนองเดียวกันการแปลงกลับจะต้องหมุนรอบแกน Y ในทิศทางตรงข้ามเป็นมุม N ดังนั้น อินเวอร์ สเมตริกส์คือ

$$Ry\text{-}1 = \begin{bmatrix} V/L & 0 & -A/L & 0 \\ 0 & 1 & 0 & 0 \\ A/L & 0 & V/L & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{........(2.41)}$$

ต่อไปเราจะหมุนวัตถุรอบแกน Z เป็นมุม a โดยที่มุม a นี้คือมุมที่เรา ต้องการจะหมุนวัตถุไปรอบ ๆ แกนหรือเส้นตรงที่เรากำลังกล่าวถึงอยู่ ดังนั้นเมตริกซ์การแปลงแบบหมุนรอบแกน Z คือ

$$Rz = \begin{bmatrix} \cos(a) & \sin(a) & 0 & 0 \\ -\sin(a) & \cos(a) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{.........(2.42)}$$

ดังนั้นเมตริกซ์การแปลงแบบหมุนรอบแกนใด ๆ Ro คือ

$$\mathbf{Ro = T\ Rx\ Ry\ Rz\ Ry\text{-}1\ Rx\text{-}1\ T\text{-}1} \qquad \text{........(2.43)}$$

บทที่ 3

การหาค่าสีของภาพ

เมื่อแสง ตกกระทบพื้นผิวอาจถูก ดูดกลืน (Absorbed) สะท้อน (Reflect) หรือ ผ่านไปได้ (Transparent) แสงส่วนที่ถูกดูดกลืนจะ กลายเป็นความร้อน ส่วนที่เหลือจะเป็นส่วน ที่เรามองเห็นวัตถุได้ สิ่งที่ชี้ได้ว่า พลังงานจะถูก ดูดกลืน .หรือสะท้อน ก็คือ ความยาวคลื่น (WaveLength) ของแสงเช่น วัตถุสามารถดูดกลืน แสงได้ทุกความถี่เท่า ๆ กัน ดังนั้นเมื่อมี แสงสีขาวมาตกกระทบ วัตถุนั้นจะมีสี เทา ถ้าวัตถุนั้นดูดกลืน แสงได้มาก วัตถุนั้นจะมีสี ดำ แต่ถ้า วัตถุนั้น สามารถดูดกลืนแสง ได้เฉพาะ บางความ ยาว คลื่น วัตถุนั้นจะมองเห็นเป็นสีของความยาว คลื่น ที่ไม่ถูกดูดกลืน

คุณสมบัติ การสะท้อน จากพื้นผิวของวัตถุขึ้นอยู่กับส่วนประกอบ ทิศทาง คุณสมบัติ ทางเรขาคณิต ของ แหล่งกำหนดแสง รูปทรงของวัตถุ คุณสมบัติของพื้นผิววัตถุ แสงที่ สะท้อนมาจาก วัตถุสามารถ แบ่ง ตามคุณสมบัติของได้เป็น 2 ชนิด

3.1 __Diffusely Reflected__ คือแสงส่วนที่ทะลุผ่านพื้นผิวของวัตถุ มีบางส่วนที่ ถูกดูดกลืน และ สะท้อนออกมา คุณสมบัติของแสง สะท้อน แบบนี้คือ จะ สะท้อน ออกมาเท่ากันทุกทิศทาง ขึ้นอยู่ กับ ตำแหน่ง ของผู้สังเกตุ

3.2 __Specular Reflected__ คือแสงส่วนที่สะท้อนออกจากผิวของวัตถุตาม Lambert's Cosin Law กล่าวว่า แสงจากแหล่งกำเนิดแสงที่เป็นจุด เมื่อส่องไปยังวัตถุ ที่มีการสะท้อน แบบสมบูรณ์ จะทำให้เกิด ความเข้มตามสัดส่วน ของ Cosin ของมุมระหว่างทิศทางของแสง กับ เส้นตั้งฉากของพื้นผิว

$$I = I1\ Kd\ COS\ (\theta) \qquad \text{............(3.1)}$$

$$0 < Kd < 1,\ 0 < \theta < \pi/2$$

เมื่อ I คือ ความเข้มของแสงสะท้อน

I1 คือ ความเข้มของแสงที่ตกกระทบ

Kd คือ ค่าคงที่ Diffuse Reflected ขึ้นอยู่กับชนิดของ วัตถุ และมีความสัมพันธ์ กับความยาว คลื่นของ แสงที่ ตกกระทบ ด้วย

θ คือ มุมที่แสงตกกระทบทำกับเส้นตั้งฉากของพื้นผิว

ถ้า $\theta > \pi/2$ แสดงว่าแหล่งกำเนิดแสงอยู่หลังวัตถุ

วัตถุที่ทำการ Rendering ด้วยวิธีการของ Lambertian Diffuse Reflected Illumination Model หรือ การทำ Shading ปรากฏว่าเป็นพื้นผิวที่ไม่เรียบ เป็นเหลี่ยม เป็น คม เนื่องจากเรากำหนด แหล่ง กำเนิดแสงเป็นจุด พื้นผิวของวัตถุที่ไม่ได้รับแสงจะเป็นสีดำ แต่ใน ความเป็นจริงวัตถุจะได้รับแสงที่สะท้อนมาจากสิ่งที่อยู่รอบตัว เช่น จากผนังของห้อง ดังนั้นจึงต้องพิจารณาแสงสะท้อนจากสิ่งแวดล้อมดังนี้

$$I = Ia\ Ka + Il\ Kd\ COS(\theta) \qquad ....(3.2)$$

เมื่อ Ia คือ ความเข้มของแสงตกกระทบเนื่องจากการสะท้อนของสิ่งแวดล้อม
Ka คือ ค่าคงที่ของการสะท้อนของสิ่งแวดล้อม

จากวิธีการข้างบน ในกรณีที่ วัตถุ มีรูปทรงเหมือนกันทุกประการ แต่อยู่ที่ระยะทาง ต่างกัน จะทำให้ได้ความเข้มเท่ากัน ดังนั้น เมื่อวัตถุ สองชิ้นอยู่เหลื่อมล้ำกัน เราจะไม่สามารถแยก แยะ ความแตก ต่างของวัตถุได้ ซึ่งในความเป็นจริง ความเข้มของ แสงจะเป็นปฏิภาคผกผันกับ ระยะ ทางกำลังสอง ถ้าใน ภาพ ใช้ระบบ Perspective Transformation ระยะทางระหว่าง ViewPoint ถึงวัตถุ สามารถ กำหนด ให้เป็น ค่าคงที่ (d) ได้ แต่อย่างไรก็ตาม เมื่อ Perspective ViewPoint อยู่ใกล้วัตถุ เทอม 1/d จะเปลี่ยนแปลงอย่างรวดเร็ว ทำให้ได้ภาพที่มีความเข้ม ผิดความจริง เราสามารถ แก้ปัญหา นี้ได้โดย การใช้กฎ Linear Attenuation ดังความสัมพันธ์

$$I = Ia\ Ka + (Il\ Kd\ COS(\theta)) / (d+k) \qquad ......(3.3)$$

เมื่อ K คือค่าคงที่ใด ๆ

ค่าความเข้มของแสงที่เกิดจาก Specular Reflected ขึ้นอยู่กับแสงตกกระทบ ความยาวคลื่น ของ แสง ที่ตกกระทบ และ คุณสมบัติของเนื้อ วัตถุ ในกรณีของ วัตถุที่มี การสะท้อนโดยสมบูรณ์ เช่น กระจก เงา ค่ามุมของแสงที่ตกกระทบ จะเท่ากับมุมของแสงสะท้อน ดังนั้นผู้สังเกตุจะสามารถมองเห็นวัตถุได้ ก็ ต่อเมื่อ ผู้สังเกตุ อยู่ในระนาบ เดียวกับแสง ที่สะท้อนออกมาเท่านั้น สำหรับ ในกรณีที่วัตถุ ที่มีการสะท้อน ออกมา อย่างไม่ สมบูรณ์ แสงที่สะท้อนออกมายังผู้สังเกตุขึ้นกับค่า การกระจายในระนาบต่าง ๆ ของ Specular Reflected

ค่าของการกระจายนี้ ขึ้นอยู่กับ คุณสมบัติของผิววัตถุ ด้วย คือ วัตถุ ที่มีผิวเรียบ จะมีการสะท้อนกระจายเป็นรังสีแคบ ๆ และใน ทางกลับกัน เมื่อผิวของวัตถุหยาบ การ กระจายจะมีลักษณะที่แผ่ออกไปไม่จำกัดเป็นรังสี ในทางธรรมชาติ ความเข้มที่จุดใด ๆ ของวัตถุยังขึ้นอยู่กับ แสงที่สะท้อนจากสิ่งแวดล้อมและ Diffuse Reflection อีก ดังนั้น เมื่อนำผลทั้งสามอย่างมารวมกันกับค่า Specularly Reflected

$$I = Ia\ Ka + I1(Kd\ COS(\theta) + W(i,\lambda)\ COS^{n}\ \alpha\ ) / (d+k) \qquad \ldots\ldots\ldots\ldots\ldots(3.4)$$

ค่า W(i,λ) เป็นความสำพันธ์ที่ยุ่งยากมาก ดังนั้นจากการทดลองเราสามารถแทนได้ด้วยค่าคงที่ Ks

$$I = Ia\ Ka + I1(Kd\ COS(\theta) + Ks\ COS^{n}\ \alpha\ ) / (d+k) \qquad \ldots\ldots\ldots\ldots\ldots(3.5)$$

แต่ จากความรู้เรื่องเวคเตอร์เรารู้ว่า

$$\cos\ \vec{\theta} = \bar{N}\cdot\bar{L} / |\bar{N}||\bar{L}| = \bar{n}\cdot\bar{L} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3.6)$$

เมื่อ $\bar{n}$ และ $\bar{L}$ เป็น เวคเตอร์หนึ่งหน่วย ของผิวและทิศทางของแสง ตามลำดับ ในทำนองเดียวกัน

$$\cos\ \alpha\ = \bar{R}.\bar{S} / |\bar{R}||\bar{S}| = \bar{R}\cdot\bar{S} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3.7)$$

เมื่อ R และ S เป็น เวคเตอร์หนึ่งหน่วย ของรังสีสะท้อน และทิศทางของผู้สังเกตุตามลำดับ ดัง นั้นสมการของความเข้ม สามารถเขียนได้ง่าย ๆ อีกแบบดังนี้แล.

$$I = Ia\ Ka + Il(\ Kd\ (\bar{n}\cdot\bar{L}) + Ka\ (\bar{R}\cdot\bar{S})\ ] / (\ d+k\ ) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3.8)$$

ปรมาน

บทที่ 4

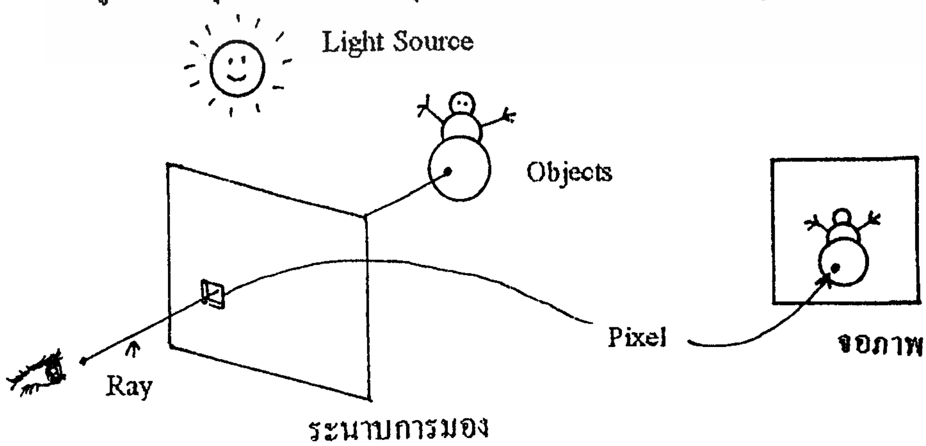
Ray Tracing

การติดตามรังสี เพื่อสร้างความสมจริง

การสร้างภาพที่มีความสมจริง (Photo - Reallism) เป็นเรื่องที่เข้าใจได้ ยากเรื่องหนึ่งในบรรดา ทฤษฎีที่เกี่ยวกับ คอมพิวเตอร์ กราฟฟิกส์ ทั้งนี้เพราะการ สร้างภาพ ที่ สมจริงได้รวม เอาปรากฏการณ์ต่าง ๆ ทางแสงในธรรมชาติทั้งหมดมารวมเอาไว้ และการสร้างภาพที่ สมจริง ด้วยวิธีการ ทางคอมพิวเตอร์กราฟฟิกส์ ก็ตอบสนองได้ เพียงระดับ หนึ่ง ของความสมจริงเท่านั้น เทคนิค ที่ใช้ในการสร้างภาพให้มีความสมจริงนั้นมีมากมาย และวิธีที่ เป็นที่นิยม กันมากใน คอมพิวเตอร์ กราฟฟิกส์ กันก็คือ การติดตามรังสี หรือ Ray Tracing

4.1 Ray Tracing คืออะไร

Ray Tracing หรือจะเรียกเป็นภาษาไทย ๆ เรียกกันว่า " การติดตามรังสี " ก็คือความพยายาม ที่จะจำลอง รังสีในระนาบ 3 มิติ เพื่อจะได้พรรณารูปร่างของ วัตถุ เช่น ทรงกลม รูปทรง หลาย เหลี่ยมหรือกล่อง เป็นต้น ด้วยแหล่งกำเนิดแสงที่ถูกต้อง โดย แหล่งกำเนิดแสงก็คือจุด ที่มีการ แผ่รังสี ของแสง ออกมาทั่วทุกทิศทางดังนั้นเราจะถือว่า วัตถุจะไม่มีการเปล่งแสง ออกมาจากตัวเอง และแหล่ง กำเนิด แสงก็ไม่ จำเป็นต้องอยู่ ในทิศทาง ของการมองเห็น (และต้องคำนึงถึงแสงที่แผ่ ออกจาก ผิวของ วัตถุ อื่นที่อาจได้ จากการคำนวนแต่ในการเขียน โปรแกรมจำเป็นจะต้องตัดทิ้ง) เราก็จะเห็น ภาพจาก จุด ในมิติที่ว่างนั่นคือตาของเรา โดยมองผ่าน กรอบหน้าต่างใน มิติที่ว่าง ซึ่งก็คือระนาบ ของการมองเห็น (View plane) ซึ่งภาพที่ปรากฏในระนาบการมองเห็น ในระนาบ 3 มิติก็คือ ภาพที่จะ แสดงบนจอภาพ โดยอยู่ระหว่างจุดบนจอภาพกับจุดบนระนาบการมองเห็น ดังรูปที่ 4.1



รูปที่ 4.1 การเกิดภาพบนจอมอนิเตอร์

วิธีที่ตรงที่สุดในการสร้างภาพก็คือการติดตามองค์ประกรอบของแสง (Photon) จากแหล่ง กำเนิดแสง ไปยังวัตถุ เรียกว่า การติดตามรังสีแบบไปข้างหน้า (Forward ray traoing) แต่ การทำ Forward ray traoing ก็ไม่อาจใช้ได้จริงเสมอไป เช่น ในกรณีที่เรา ติดตาม การเดินทาง ของรังสีของแสง ที่ตก กระทบวัตถุที่มีคุณสมบัติเป็นเงา จากนั้นก็ติดตาม แสงที่สะท้อน ออกจากวัตถุเงา นั้นไป จนกว่ารังสี นั้นจะ วิ่งเข้ามาชนตาของเรา จะเห็นว่า ถ้าหากใช้วิธีนี้แล้ว เราจะใช้ เวลาสร้างภาพ ได้ช้ามาก เพราะจะ ต้อง พิจารณา ทุก ๆ รังสีที่ออกจากแหล่งกำเนิด ดังนั้นจึงมี การคิดค้น วิธีการใหม่ ที่มีประสิทธิ ภาพมาก กว่าเดิม และง่ายกว่าเดิม โดยมีหลักการขั้นตอน ย้อน กลับจาก ขบวนการติดตามรังสี แบบไปข้างหน้า กล่าวคือเราจะเลือก จุดบนระนาบการ มองเห็นซึ่ง อยู่บนเส้นรังสี ที่พุ่งออกจากตาไปยัง ระนาบการมองเห็น หรือที่เรียกกว่า รังสีตา (Eye Ray) จุดแรก บนรังสีนี้จะเป็น จุดที่เห็นในระนาบการมองเห็น เมื่อรวมจุดเข้าด้วยกันก็จะได้ ภาพ ขึ้นมา วิธีนี้เราเรียกว่า การติดตาม รังสีแบบย้อนกลับ (Reverse Ray Traoing) เพราะเป็น การติดตามรังสีย้อน กลับจากตา กลับไปยัง ระนาบ และเช่นเดียวกับ ระนาบการ มองเห็น จอภาพจะสร้าง ภาพจากการรวมกริด สี่เหลี่ยมที่ เรียกว่า พิกเซล (Pixel) เราจะต้องหาว่าแต่ละพิกเซล บนจอภาพ นั้นควรจะแสดง สีอะไร จึง จะเป็นสีที่ถูกต้องของวัตถุ จริงที่ถูกพุ่งชนโดยรังสี ของแสง ดังเช่น ในรูปที่ 4.1 จะเห็นรังสีออกจากตา ผ่านพิกเซล บนจอภาพแล้วกระทบกับวัตถุทรงกลม

4.2 แสงตกกระทบ

เมื่อเรารู้จักกับการเกิด จุดบนระนาบ การมองเห็นแล้ว เราก็จะมา ดูวิธีการหาค่าสีที่ ถูกต้อง ของ วัตถุ สมมุติว่ารังสีจาก ตาตกกระทบ วัตถุทรงกลมสีขาว ลูกหนึ่งที่จุด P เราต้องการหาค่า สีที่แท้ จริงของแสงที่ออกจาก P และพุ่งกลับไปยังดวงตา ซึ่งสามารถทำได้โดยการติดตามรังสีแบบ ย้อนกลับ เนื่องจากแสงที่ตกกระทบวัตถุเป็นสีขาว ดังนั้นสีของแสง ที่สะท้อนออก มาจากวัตถุที่จุด P จะเป็น สีเดียวกันกับสีที่แหล่งกำเนิดแสง มายังจุด P เราเรียกว่า แสงที่ตกกระทบ (Incident Light) ในกรณีนี้ที่เราจะถือว่า วัตถุจะไม่สามารถเปล่งแสง ออกจากตัวเองได้ เราสามารถแบ่ง แยกแสง ที่ ตก กระทบกับวัตถุ ได้เป็นสองประเภท คือ แสงมากที่สุดที่สามารถสะท้อนออกจากผิว และ แสงที่สองผ่าน ทะลุผิววัตถุในกรณีที่วัตถุ ไม่เป็นวัตถุทึบแสง นอกจากนี้ การสะท้อน หรือการทะลุผ่าน ผิวของ วัตถุของแสงนี้ยังสามารถแบ่งออกได้เป็น สองชนิดคือ Specular Propagation ซึ่งเป็น การสะท้อน ส่องผ่านไปอย่างเป็นระเบียบ เช่นเดียวกับการกระเด้งของลูกบาสบนพื้นผิวเรียบ โดยจะมีมุมการสะท้อน เท่ากับมุมตกกระทบ กับอีกชนิดคือ Diffuse Propagation จะเป็น การสะท้อน

แบบ กระจัดกระจาย ในลักษณะของการ กระเด้งของ ลูกบาสบน พื้นผิวที่ขรุขระ ที่เรา ไม่สามารถ รู้ทิศทาง การกระเด้งที่ แน่นอน ของลูกบาส และถ้าเรา ส่งลูกบาสให้ กระเด้งออกจากพื้นผิวที่ ขรุขระนี้หลาย ๆ ลูก จะเห็นว่าลูกบาส จะกระจายออกไปในทุกทิศทาง ในกรณีที่เป็นแสง การสะท้อน แบบดังกล่าว จะมีความเข้มของแสงเท่า ๆ กันในทุกทิศทาง แต่จะน้อยกว่าแสงที่ตกกระทบ ใน บางครั้งเราจะเรียก Diffuse Propagation นี้ว่า Diffuse Scattering

สรุปว่าแสงจะมีการสะท้อน ออกจากวัตถุ ได้ 4 วิธีคือ การสะท้อนแบบเป็นระเบียบ การสะท้อน แบบไม่เป็นระเบียบ การส่องผ่านแบบ เป็นระเบียบ และการส่อง ผ่านแบบไม่เป็นระเบียบ อีกสิ่ง หนึ่ง ที่เราไม่สามารถละทิ้งได้ในเมื่อพูดถึงการสะท้อน ของแสง นั่นคือต้นกำเนิดของ แสงที่ ตกกระทบ วัตถุ นั้นมาจากที่ใด แหล่งกำเนิดแสงจะมีได้ 2 ชนิดคือ แสงจากแหล่ง กำเนิด แสง และ แสงที่เกิดขึ้นจาก การสะท้อนจากพื้นผิวอื่น ๆ ซึ่งทั้งสอง แหล่งกำเนิดสามารถเกิดขึ้น ได้พร้อมกัน การพิจารณา แหล่งกำเนิด แสง และ การสะท้อน นี้จะนำไปสู่การระบายสี หรือ Shading ต่อไป

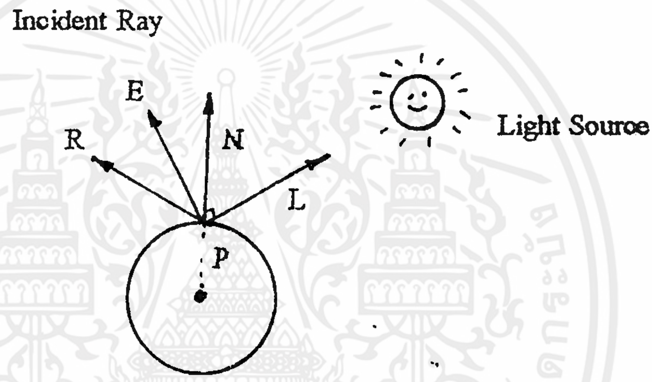
4.3 แสงที่ส่องตรงจากแหล่งกำเนิด

ในการพิจารณา แสงที่ส่องตรง จากแหล่งกำเนิดแสง สิ่งแรกที่จะต้องทำก็คือพิจารณา ว่ามีแหล่ง กำเนิดแสง ใดบ้าง ที่ส่องแสงไปยังวัตถุอยู่ ถ้าจุด P เป็นจุด หนึ่งบนวัตถุที่เราพิจารณา ถูกส่อง โดยแหล่งกำเนิดแสง S แล้วเราจะสามารถบอกได้ว่า ที่แหล่งกำเนิดแสง S เราจะมองเห็น จุด P และในทางกลับกัน ที่จุด P เราจะมองเห็นแหล่งกำเนิดแสงจุด S เช่นกัน รังสีที่มีทิศทาง จากจุด P พุ่งไปยังแหล่งกำเนิด S รังสีนี้จะเรียกว่ารังสีเงา หรือ Shadow Ray ในการพิจารณา แสงตรงนี้ เราจะใช้รังสีเงาแทนที่จะใช้รังสีที่ออกจากแหล่งกำเนิดแสง ตามวิธี Reverse Ray Tracing

ถ้ารังสีนี้กลับสู่ S โดยที่ไม่ชนกับวัตถุใด ๆ ก่อนเลยเราสามารถหาได้ว่า มีแสงจาก S ตกกระทบ ที่จุด P แล้วสะท้อนเข้าสู่ตาของเราในปริมาณเท่าใด แต่ถ้ารังสีจาก P ถึง S ถูกขัด ขวาง เช่นรังสีตัดผ่านวัตถุอื่น ๆ เสียก่อนแล้ว P จะอยู่ในเงาที่เกิดจากแหล่งกำเนิดแสง S ดังนั้น เราจึงไม่จำเป็นที่จะต้องพิจารณาผลที่เกิดขึ้นกับจุด P เนื่องจากแหล่งกำเนิดแสง S อีกต่อไป

4.4 การระบายสี

เราจะแสดงตัวอย่างง่าย ๆ ของการระบายสี โดยพยายาม ไม่พูดถึง ทฤษฎีในขั้นสูง มากนัก เพราะจุดประสงค์ของเราก็คือ การกล่าวถึงทฤษฎีเบื้องต้นเท่านั้น สมมุติให้จุด P ถูกส่อง โดยแหล่ง กำเนิดแสง S โดยทั่วไปแล้ว จุดบนผิวของวัตถุทั่วไปนั้น จะมีคุณสมบัติที่สำคัญประการหนึ่ง คือ เวคเตอร์ที่ผิววัตถุใด ๆ จะพุ่งออกจากวัตถุและตั้งฉากกับ ระนาบที่สัมผัสกับจุดนั้นเสมอ ยกตัว อย่างเช่น ทรงกลมจะมีเวคเตอร์ของพื้นผิว ในทิศทางพุ่งออกจากจุดศูนย์กลางเสมอ ดังรูปที่ 4.2 เรา เรียก เวคเตอร์นี้ว่า เวคเตอร์ตั้งฉาก หรือ Normal Veotor ในรูปคือเวคเตอร์ N ส่วน เวคเตอร์ L เป็นรังสีเงาจากจุด P ไปยังแหล่งกำเนิดแสง S ซึ่งเราจะพิจารณารังสีสะท้อน L เทียบกับ Normal Veotor N เสมอ



รูปที่ 4.2

ในการหาตำแหน่งของแหล่งกำเนิดแสง S เทียบกับจุด P ที่เรากำลังพิจารณานั้น สามารถกระทำได้โดยการหา Dot Produot ของเวคเตอร์ N และเวคเตอร์ L นั่นคือถ้า $N.L < 0$ แล้ว S จะอยู่ด้านหลังของผิว ถ้า $N.L > 0$ แล้ว S จะอยู่ด้านหน้าของวัตถุ แต่ถ้า $N.L = 0$ ก็หมายความว่า S จะอยู่บนระนาบเดียวกันกับ ระนาบของผิวที่สัมผัสอยู่กับวัตถุ และ แสงที่ตกกระทบ ที่จุด P จะสะท้อน กลับไปยังแหล่งกำเนิดแสง S หมดเราจะสมมุติให้ $N.L > 0$ ดังนั้น ซึ่งจะทำให้เกิดการสะท้อน กลับของแสงจากพื้นผิว เราสามารถ หาความเข้ม ของแสงอย่างไม่ เป็นระเบียบจากพื้นผิวได้ โดยจะกำหนด ให้ความเข้มของแหล่งกำเนิดแสง S มีค่าเป็น 1 และกำหนดให้ DR เป็นค่าของ ความเข้มแสง ที่จะสท้อนกลับจากผิวเราจะได้ความสัมพันธ์ระหว่าง DR และ 1 เป็น $DR = 1(N.L)$ สมการนี้ได้เป็นที่รู้จักกันดีในนามของ Lambert's Law ในการหาจุด ที่เป็นจุดที่ สว่างที่สุดบน วัตถุ นั้น เราจะ ถือเอาหลักการที่ว่า จุดที่เกิดการสะ ท้อน

กลับมาสู่ตาของผู้สังเกตุ โดยตรง จะเป็น จุดที่มีความสว่างมากที่สุด ซึ่งจะเกิดขึ้นกับตำแหน่งที่อยู่ของตาด้วย ส่วนวิธีการหานั้นเราจะสร้างเวกเตอร์ E จากจุด P ไปยังดวงตา ดังรูปที่ 3.2 โดย L จะพุ่งเข้าสู่แหล่งกำเนิดแสง และ เวกเตอร์ R เป็นทิศทาง แสงที่สะท้อนจากพื้นผิว จากกฎ ของการอย่างเป็นระเบียบ เราจะได้ความสำพันธ์ของ L, R และ N ว่า

$$R = 2Nd - L \qquad \text{เมื่อ } d \text{ คือ } N.L$$

ทำให้สามารถหาจำนวนรังสี ที่พุ่งเข้าสู่ตาได้โดยการหาว่ามี R และ E เกิดขึ้น เท่าไร โดย ใช้สมการ

$$SR = [\ (E.R)^{k}\ ]$$

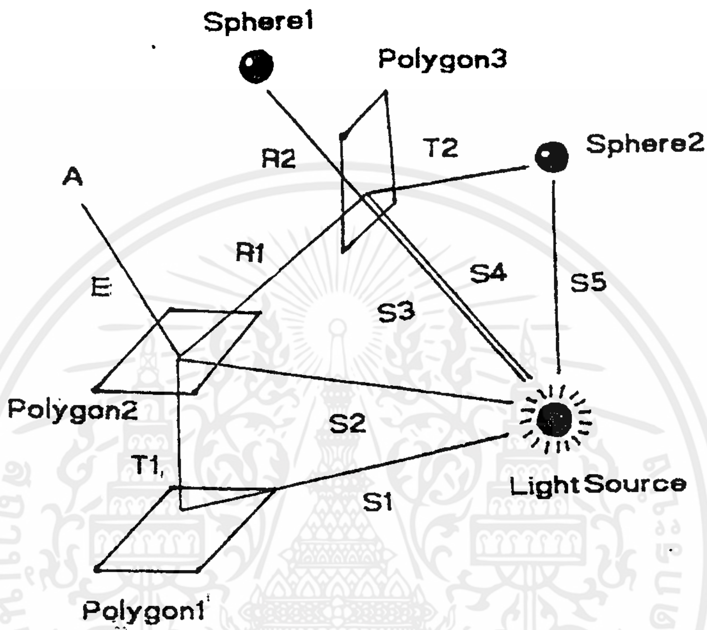
เมื่อ SR คือ ค่าของความเข้มของการสะท้อนกลับอย่างเป็นระเบียบ
k คือ ค่าความอยาบของพื้นผิว

ขั้นตอนต่อไปคือ การหาแสง ตกกระทบที่มาจากวัตถุ อื่น ๆ โดยใช้รังสี ตาพิจารณา ในทิศทางย้อน กลับรังสีแสง ด้วยวิธี Reverse Ray Tracing ว่ามีแสงใด ที่จะสะท้อนกลับสู่ตาบ้าง จากสมการ

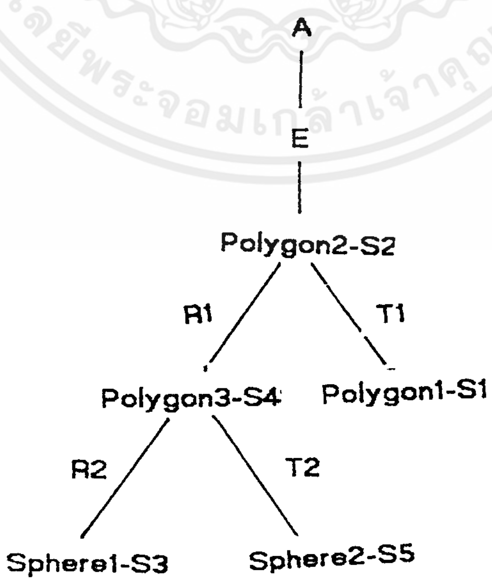
$$R' = 2N(E.N) - E$$

เหตุที่ใช้ R' แทน R ก็เพราะในการคำนวนนี้ เราใช้วิธีย้อนกลับของรังสี ดังนั้นแสงที่สะท้อน กลับ จาก E ซึ่งเป็นรังสีตาจะเขียนในรูป R' ต่อไปเราจะ พิจารณาว่า มีรังสีใดที่ พุ่งมาทางทิศทาง R' บ้างโดยการย้อนทิศทาง ของรังสี ตกกระทบ ที่พุ่งจากวัตถุมายังตา แล้ว พิจารณาว่าที่จุดนั้น ได้รับแสงสะท้อน จากวัตถุ ใดเป็นสีใดบ้าง สีที่เรามองเห็นจะเป็น สีที่เกิด จากการผสมสีเหล่านี้ ใน การ คำนวน จริง จะมีการการสะท้อน จากวัตถุหลาย ๆ ชิ้น แต่ละชิ้นก็จะมีการสะท้อนเกิดขึ้นเช่นกัน ซึ่งจะทำให้การคำนวณเป็นไปได้ยากมาก เราอาจ ใช้แผนภูมิต้นไม้เพื่อแสดง รังสีได้ดังรูปที่ 4.4 ขึ้นเพื่อ ให้สามารถ หาสีที่แท้จริงได้ง่ายขึ้น โดยพิจารณาจากรังสีสะท้อนโดยตรงไล่ไป ตัวอย่างเช่น เรา อยู่ที่จุด A และกำลังพิจารณา Polygon 2 ซึ่งมีรังสี R1 และ T1 ตกกระทบอยู่ และจะเห็นว่ารัง สี R1 นั้นเกิดจากรังสีสะท้อน R2 กับ T2

และจากแหล่ง กำเนิดแสง ตรง S4 ในขณะเดียวกัน R2 จากทรงกลม Sphere 1 ก็เกิด จากรังสี จากแหล่งกำเนิดแสง S3 และรังสี T2 ก็เกิดจากทรงกลม Sphere2 ซึ่งมีรังสี S5 จากแหล่งกำเนิดแสง ตกกระทบ อยู่ เราสามารถเขียนแผนภูมิ ต้นไม้ได้ดัง รูป 4.4 ข้าง ๆ จากนั้น เราจะค่อย ๆ คำนวน หาค่าสีที่แท้จริงจากรังสีต่าง ๆ ไล่ขึ้นมาจากส่วน ล่างของแผน ภูมิ จนได้สีที่แท้จริงในที่สุด



รูปที่ 4.3



รูป 4.4

4.5 การตัดกันของรังสี กับ ทรงกลม

ทรงกลมเป็นรูปทรงหนึ่งที่นิยมใช้กันมาก ในการติดตามรังสี ด้วยที่มันเป็นการง่ายที่จะ ทดสอบว่า รังสีตัดกับ ทรงกลม ซึ่งอาศัยขอบเขตของทรงกลมมาใช้ในการ ควบคุมอันดับแรก สมการทรงกลมจะถูกนำมา วิเคราะห์ ก่อน หลังจากนั้น จะต้อง ทดสอบ เงื่อนไขบางอย่างก่อน จึงจะแน่ชัด ว่า รังสี ตัดกับทรงกลมหรือ ไม่ ดังแสดงให้ได้ดังนี้.

กำหนดให้ รังสีเริ่มต้นที่ $Ro = [X_0 \ Y_0 \ Z_0]$

มีทิศทางชี้ไปที่ $Rd = [X_d \ Y_d \ Z_d]$

ปรกติแล้ว $X_d^2 + Y_d^2 + Z_2^2 = 1$

หรือจะเขียนให้อยู่ในรูปของสมการพาราเมตริกส์ได้ดังนี้

$$R(t) = Ro + Rd^*t \qquad \text{ขณะที่ } t > 0 \qquad (3.1)$$

จุดใด ๆ ที่อยู่บน $t < 0$ จะเป็นจุดที่อยู่หลังจุดกำเนิด ขณะที่ $t = 0$ จะเป็นจุดที่อยู่บนจุดกำเนิด

และทรงกลมก็จะมี สมการทั่ว ๆ ไปดังนี้

จุดศูนย์กลางของทรงกลม $Sc = [\ Xo \ Yo \ Zo\ ]$

เส้นรัศมี ทรงกลม Sr

ผิวของทรงกลม $[\ Xs \ Ys \ Zs\ ]$

$$(Xs - Xo)^2 + (Ys - Yo)^2 + (Zs - Zo)^2 = Sr^2 \qquad (3.2)$$

เราไม่สามารถ ที่จะหาจุดบนทรงกลม ได้โดยตรง แต่เรา สามารถ หาจุดบนทรงกลม ได้โดย การทดสอบเงื่อนไข ต่าง ๆ อีก

ในการแก้ปัญหา เพื่อที่จะหาจุดตัด เราจะมาพิจารณา สมการของ รังสี ที่ เป็นสมการ พาราเมตริกส์ ซึ่งจะสามารถ แปรค่าได้ตามค่าของ t โดยการแตกสมการออกเป็น สมการของ X ,Y และ Z

ที่จุด P [X Y Z] ใด ๆ สามารถเขียนให้อยู่ในรูปของสมการพาราเมตริกส์ได้ดังนี้

$$X = X_0 + X_d\ t$$
$$Y = Y_0 + Y_d\ t$$
$$Z = Z_0 + Z_d\ t$$

แทนค่า X , Y และ Z ลงในสมการของทรงกลมจะได้

$$(X_0 + X_d t - X_o)^2 + (Y_0 + Y_d t - Yo)^2 + (Z_0 + Z_d t - Z_o)^2 = Sr^2$$

ทำให้อยู่ในเทอมของ t จะได้ $At^2 + Bt + C = 0$ ขณะที่

$$A = X_d^2 + Y_d^2 + Z_d^2$$
$$B = 2(X_d(X_0 - X_o) + Y_d(Y_0 - Y_o) + Z_d(Z_0 - Z_o))$$
$$C = (X_0 - X_o)^2 + (Y_0 - Y_o)^2 + (Z_0 - Z_o)^2 - Sr^2$$

ค่าสัมประสิทธิ A จะต้องทำให้เป็น 1 เสมอโดยการเอา A หารตลอด สมการ และ

$$t_0 = (-B - B^2 - 4C\ )/2$$
$$t_0 = (-B + B^2 - 4C\ )/2$$

เมื่อค่า $B^2 - 4C$ น้อยกว่า 0 หรือเป็นลบ จะหมายความว่า รังสีไม่ตัดกับทรงกลม ถ้าหาค่า t_0 , t_1 ได้ จะหาต่อไปอีกว่า ค่าที่มีค่าที่มีค่าน้อยที่สุดที่มีค่าเป็นบวก และ $t_0 = t_1 = 0$ หมายว่า รังสีสัมผัสกับ ทรงกลม พอดี

และสามารถหาจุดที่ รังสี ตัดกับ ทรงกลมได้ที่จุด $R_{interseot}$

$$R_{interseot} = R_i = [X_i \quad Y_i \quad Z_i] = [X0 + Xdt \quad Y0+Ydt \quad Z0+Zdt]$$

และ เวคเตอร์ตั้งฉาก กับผิวหาได้โดย

$$R_{normal} = [\ (Xi - Xo) \quad (Yi - Yo) \quad (Zi - Zo)\ ]\ /\ Sr$$

<u>ตัวอย่าง</u> รังสีเริ่ม ต้นที่ [1, -2 , -1] และทิศทาง [1 , 2 , 4] จงหาค่าของ t ที่รังสีตัดกับทรงกลม ที่มี จุดกำเนิด อยู่ที่ [3 , 0 , 5] และ มีรัศมีที่ Sr = 3

- อันดับแรก หาค่า UnitVector ของ รังสี

  Rd = [1/ 21 2/ 21 4/ 21] = [0.218 0.436 0.873]

- หาค่า A B C

  A = 1

  B = 2(0.218(1-3) + 0.436(-2-0) + 0.873(-1-5))

  = -13.092

  $C = (1-3)^2 + (-2-0)^2 + (-1-5)^2 - 3$

  = 35

- เช็คค่า $B^2 - 4C$ $(-13.092)^2 - 4*35 = 31.4$ มากกว่า 0 ก็ O.K. เมื่อ ผ่านมาตรงนี้แล้ว เราทราบว่า รังสีตัดกับ ทรงกลม เราสามารถหาค่า t ได้

$t_0 = (-B - B^2 - 4C) / 2 = (13.092 - 31.4) / 2 = 3.744$

- ขณะเดียวกัน $t_1 = (13.092 + 31.4) / 2 = 9.34$ เราเลือกค่า $t = t_0$

Ri = [X0 + Xdt Y0+Ydt Z0+Zdt]

= [1+ 0.218*3.744 -2+0.436*3.744 -1+0.873*3.744]

= [1.816 -0.368 2.269]

- และ UnitVector ของ จุด Ri คือ

Yn = [(Xi - Xo) (Yi - Yo) (Zi - Zo)] / Sr

= [(1.816 -3) (-0.368 - 0) (2.269 - 5)] / 3

= -0.395 -0.123 -0.910

4.6 การตัดกัน ของรังสี กับ ระนาบ

- กำหนดให้ จุด กำเนิด รังสี Ro = [Xo Y0 Z0]

  ทิศทางของรังสี Rd = [Xd Yd Zd]

  $Xd^2 + Yd^2 + Zd^2 = 1$

  หรือ R(t) = Ro + Rdt , t > 0

- กำหนดให้ สมการระนาบ อยู่ในเทอมของ [A B C D]

  Plane = Pn = Ax + By + Cz + D = 0

  $A^2 + B^2 + C^2 = 1$

  UnitVeotor = [A B C]

- แทนค่า X,Y, และ Z จากสมการ พาราเมตริกซ์ ใน Pn

$$A(Xo + Xdt) + B(Yo + Ydt) + C(Zo + Zdt) + D = 0$$

และแก้สมการหาค่า t จะได้

$$t = -(AXo + BY0 + CZo + D) / (AXd + BYd + CZd)$$

ในรูปของเวกเตอร์

$$t = -(Pn\ Ro + D) / (Pn\ Rd)$$

ในการพิจารณาก่อนอื่นจะต้องหาค่า Vd = Pn Rd ก่อน เมื่อ Vd = 0 หมายความว่า ขนานกันกับ ระนาบถ้า t < 0 หมายความว่า รังสีตัดกับรังสีกับ ระนาบที่ด้านหลัง เมื่อหาค่า t ได้แล้วเราสามารถหาจุดตัดกับระนาบได้ ที่

Ri = [Xo+Xdt Yo+Ydt Zo+Zdt]

และถ้า Pn Rd และมีค่าน้อยกว่า 0 เวกเตอร์ ตั้งฉาก จะหาได้จาก m = Pn

ถ้า Pn Rd มีค่ามากกว่า 0 m = -Pn

4.7 การตัดกันของ รังสี กับรูปทรงแบบ Quadric

เราจะสามารถหาจุดตัดระหว่างรังสีกับ Quadrio โดยใช้สมการ เมตริกส์แบบทั่วไป

$$[X\ Y\ Z\ 1]\begin{bmatrix} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ M & N & O & P \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0$$

จากสมการ Matrix ข้างบน สามารถเขียน ให้ เป็นรูปฟังก์ชั่น F(X,Y,Z) = 0 ได้ดังนี้

$$F(X,Y,Z) = AX^2 + 2BXY + 2CXZ + 2DX + EY^2 + 2FYZ + 2GY + HZ^2 + 2IZ + J$$

แล้วแทนค่า X, Y, และ Z ของสมการพาราเมตริกส์ ลงไปจะได้ ว่า

$$Aq = AXd^2 + 2BXdYd + 2CXdZd + EYd^2 + 2FYdZd + HZd^2$$

$$Bq = 2(AXoXd + B(XoYd + XdYo) + C(XoZd + XdZo) + DXd + EYoYd + F(YoZd + YdZo) + GYd + HZoZd + IZd)$$

$$Cq = AXo^2 + 2BX0Y0 + 2CX0Z0 + 2DX0 + EYo^2 + 2FY0Zo + 2GY0 + HZo^2 + 2IZo + J$$

ถ้า Aq = 0 ให้เช็คว่า $Bq^2 - 4AqCq$ ถ้า < 0 จะไม่ตัดกับวัตถุ นอกเหนือจากนี้ ให้คำนวนหา t ได้ดังนี้

$$t_0 = (-B_q - B_q^2 - 4A_qC_q) / 2A$$

$$t_1 = (-B_q + B_q^2 - 4A_qC_q) / 2A$$

ถ้า Aq = 0 สามารถหา t ได้ง่ายมาก

$$t = -Cq / Bq$$

4.8 การหาจุดตัดกันของรังสีกับทรงกระบอกรัศมีหนึ่งหน่วย

สมการผิวของทรงกระบอกคือ $X^2 + Y^2 = 1,\ 0 < Z < 1$ เราสามารถเขียนเป็นสมการพาราเมตริก $At^2 + Bt + C = 0$ ขณะที่

$$A = b_x^{\ 2} + b_y^{\ 2}$$
$$B = a_x b_x + a_y b_y$$
$$C = a_x^{\ 2} + a_y^{\ 2} - 1$$

ถ้า $A = 0$ หมายถึงว่า รังสีไม่ตัดกับวัตถุแต่จะขนานกันไป ถ้า $A \mathrel{!}= 0$ และ $(B^2 - AC) < 0$ หมายถึงรังสีไม่ตัดกับวัตถุ นอกเหนือจากนี้ สามารถหาจุดตัดได้ โดยที่

$$t = \frac{-B \pm \sqrt{B^2 - AC}}{4}$$

และจะต้องอยู่ในขอบเขตของ $(ax+bxt)^2 + (ay+byt)^2 < 1$ ด้วย

4.9 การหาจุดตัดกันของรังสีกับทรงกรวยรัศมีหนึ่งหน่วย

สมการผิวของทรงกรวยคือ $X^2 + Y^2 = (1 - Z)^2 / 4$ เราสามารถเขียนเป็นสมการพาราเมตริก $At^2 + Bt + C = 0$ ขณะที่

$$A = b_x^{\ 2} + b_y^{\ 2} - 0.25bz^2$$
$$B = a_x b_x + a_y b_y + 0.25bz(1 - az)$$
$$C = a_x^{\ 2} + a_y^{\ 2} - 0.25(1 - az)^2$$

ถ้า $A = 0$ หมายถึงว่า รังสีไม่ตัดกับวัตถุแต่จะขนานกันไป ถ้า $A \mathrel{!}= 0$ และ $(B^2 - AC) < 0$ หมายถึงรังสีไม่ตัดกับวัตถุ นอกเหนือจากนี้ สามารถหาจุดตัดได้ โดยที่

$$t = \frac{-B \pm \sqrt{B^2 - AC}}{4}$$

และจะต้องอยู่ในขอบเขตของ $(ax + bxt)^2 + (ay + byt)^2 < 0.25(1\text{-}azt)^2$ ด้วย

บทที่ 5

Raster Scan Graphics

Raster Soan Graphics คือกระบวนการย่อย ๆ สำหรับการกำเนิดภาพทาง Computer graphios ไม่ว่าจะเป็น การลากเส้น การระบายรูปหลายเหลี่ยม วงกลม และอื่น ๆ มากมายเอามาก่ายกันได้เป็นกอง ที่จะกล่าวถึงในรายงานฉบับนี้ จะกล่าวถึง เฉพาะที่ได้ นำมา ใช้งานในปริญญานิพนพ์ เท่านั้น

5.1 การลากเส้น Line

จากการที่ จอมอนิเตอร์ CRT ทั่ว ๆ ไป จะประกอบด้วยกลุ่มของจุดเล็ก ๆ มากมายรวมกันอยู่ โดยที่แต่ละจุด สามารถที่จะกำหนด ให้ สว่าง หรือ มืด ได้ แต่ เราไม่ สามารถที่จะสั่งให้มัน สว่างเป็นเส้นตรง หรือ เป็นวงกลมได้ โดยตรง ดังนั้นในการนี้เรา จะนำเสนอการสร้างเส้นตรง วิธีที่คิดว่า นิยมมากที่สุด มาให้ท่านได้ พิจารณากัน

● วิธีแรกสุด แสนจะง่ายและธรรมดา ๆ โดยการใช้ความรู้ทาง คณิตศาสตร์ พื้นฐานเล็ก ๆ มาช่วย โดยที่เราพอจะทราบกันอยู่แล้วว่า ความชันของเส้นตรง มีค่าเท่ากับ

$$Dy / Dx = \text{Constant} \text{ หรือ } \Delta y / \Delta x = (Y2-Y1) / (X2-X1)$$

เมื่อเราทราบว่าค่าความชัน ไอ้การที่เราอยากจะรู้ค่าที่จุด X,Y ใด ๆ เราก็ สามารถรู้ ได้โดยง่าย เหมือนกินชมพู่ (กินกล้วยต้องเสียเวลาปลอกเปลือก)

● วิธีต่อไป **Bresenham's Algoritm** เป็นวิธีการที่ แยบยล มาก สามารถนำเอาค่า Error มาใช้งานได้อย่างชาญฉลาดมาก นั่นคือ

◆ กำหนดให้ Error $e = (\Delta y / \Delta x) - 0.5$

◆ ถ้าหากค่า $e < 0$ ให้ Dot จุดลงที่ตำแหน่ง X,Y ได้

◆ ถ้าหากค่า $e > 0$ ให้ ลดค่า e ลง 1 หรือ $e = e - 1$ แล้วเพิ่มค่า Y จนกระทั่งค่า $X = X2$

เมื่อรวมกระบวนการแล้วจะได้เส้นตรงขึ้นมาปรากฏแก่สายตาของเราเอง เรามาลองเขียนฟังก์ชั่นภาษา C ให้เขียนเส้นตรงกันบ้าง

5.2 ฟังชั่นก์ วาดเส้นตรงโดยใช้หลักการของ Bresenham's

```
void DrawLine(int x1,int y1,int x2,int y2)
  {
   int        x,y,i;
   float   e,dx,dy;

   dx = (x2-x1);
   dy = (y2-y1);
   e  = dy/dx  -  0.5;
   y  = y1;
   x  = x1;
   for(i=1;i<dx;i++)
     {
      PutPixel(x,y,CurrentColor);
      while(e>=0)
        {
         PutPixel(x,y,CurrentColor);
          e -= 1;
        }
       x ++;
       e += dy/dx;
     }
  }
```

จากความเชี่ยวชาญของ Programmer จะต้องทราบตั้งแต่เห็นการ Deolear แล้วว่า เมื่อได้มีตัวเลข Floating point มายุ่งด้วยแล้วที่นั่นจะมี เต่า อยู่ที่นั่น ความเร็ว ของฟังก์ชั่น เป็นเรื่องสำคัญมากในวงการ Computer Graphics เราจึงได้ค้นหาวิธีการที่จะทำให้เร็วขึ้นได้ โดยอาศัยการคำนวนแบบ เลขจำนวนเต็มทั้งหมด จะได้โปรแกรมใหม่ดังนี้

- การหาค่า e เปลี่ยนเป็น $e = 2 * \Delta y - \Delta x$
- $e = e - 1$ เปลี่ยนเป็น $e = e - 2*\Delta x$
- $e = e + (dy/dx)$ เปลี่ยนเป็น $e = e + 2*\Delta y$

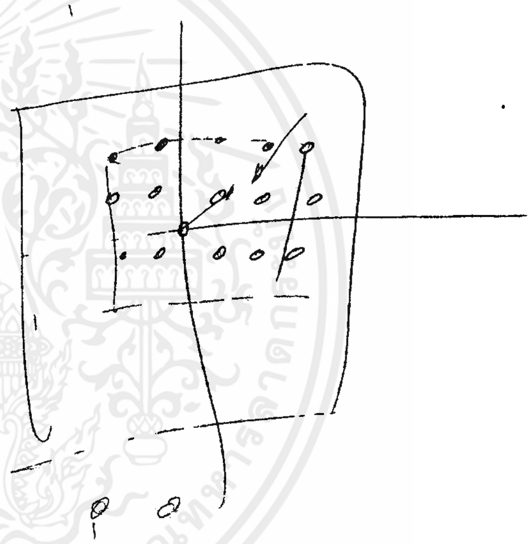
5.3 ฟังชั่นก์ที่มีการแก้ไขให้ ใช้ตัวแปรเป็นค่าตัวเลขจำนวนเต็มทั้งหมด

```
void DrawLine(int x1,int y1,int x2,int y2)
  {
   int        x,y,i,e,dx,dy;

   dx = (x2-x1);
   dy = (y2-y1);
   e  = 2*dy - dx;
   y  = y1;
  x  = x1;
   for(i=1;i<dx;i++)
     {
      PutPixel(x,y,CurrentColor);
      while(e>=0)
        {
         PutPixel(x,y,CurrentColor);
          y++;
          e -= 2*dx;
        }
      x ++;
      e += 2*dy;
    }
  }
```

ที่ผ่านไป สองโปรแกรมนั้นมันลากให้เฉพาะ Quadrant 1 เท่านั้นเราก็ต้องมีการเช็คเครื่องหมายเอาเองในโปรแกรม จะได้โปรแกรมใหม่มาดังนี้

5.4 ฟังชั่นก์ ที่มีการแก้ไขให้ สามารถวาดได้ทุก ทิศทาง

```
void DrawLine(int x1,int y1,int x2,int y2)
  {
   int                x,y,i,e,dx,dy;
   int    s1=0,s2=0,Interohange;

   dx = abs( x2-x1 );
   dy = abs( y2-y1 );
   s1 = ( (x2-x1) > 0 ) ? 1 : -1 ;
   s2 = ( (y2-y1) > 0 ) ? 1 : -1 ;
   if( dy > dx )
     {
      e  = dx;
      dx = dy;
     dy =  e;
      Interohange = 1;
     }
   else Interohange = 0;
   e  = ( 2*dy ) - dx;
   y  = y1;
   x  = x1;
   for( i=1; i<dx ;i++ )
     {
      PutPixel( x,y,CurrentColor );
      while( e>=0 )
        {
         PutPixel( x,y,CurrentColor );
         if(Interohange) x += s1;
         else              y += s2;
         e -= 2*dx;
```



```
      }
    if( Interohange ) y += s2;
    else              x += s1;
    e +=  2*dy;
   }
}
```

5.5 Parametric Cuves

ในการแปรค่า ของจุดแต่ละจุด บนเส้นโค้ง เราสามารถที่จะเขียนให้อยู่ในรูปของ ฟังก์ชั่นตัวแปรเดี่ยว ได้ ตำแหน่งเวกเตอร์ของจุด ใด ๆ บนส่วนโค้ง จะกำหนดให้ได้โดย ค่าของ ตัวแปรนั้น ในระบบ คาร์ทีเซี่ยน จุดใด ๆ บนส่วนโค้งจะกำหนดได้ดังนี้

$$X = X(t)$$

$$Y = Y(t)$$

ตำแหน่งใด ๆ บนส่วนโค้งจึงสามารถเขียนได้ อีกแบบในลักษณะของ เวกเตอร์ คือ

$$P(t) = [\ X(t) \quad Y(t)\ ]$$

แต่ทว่า สมการข้างบนเหมาะสำหรับ การสร้างเส้นโค้งแบบปิดเท่านั้น หากแต่ว่าการที่จะทำส่วนโค้งใด ๆ ในอวกาศนั้นทำไม่ได้ เราจึงต้องใช้ ค่า tangen ของ Veotor บนส่วนโค้งนั้น ๆ มาทำการกำหนด เป็นสมการใหม่ได้ว่า

$$P'(t) = [\ X'(t) \quad Y'(t)\ ]$$

ขณะที่ ' หมายถึง การหา differentialtion ที่อ้างถึงตัวแปรนั้น ๆ ซึ่งก็คือ ค่าความชันของส่วน โค้งนั่นเอง ก็คือ dx / dy

$$\frac{dy}{dx} = \frac{dx}{dt} \cdot \frac{dt}{dx} \qquad \frac{Y'(t)}{X'(t)}$$

เมื่อ $x'(t) = 0$ Slope เท่ากับ infinite

จากการที่ จุดบนส่วนโค้ง เราสามารถ เฉพาะเจาะจงเป็น ค่า ๆ หนึ่งของสมการ และจุดคงที่ จุด หนึ่งของสมการได้ จุดสุดท้ายของส่วนโค้ง และ ระยะทางจะกำหนดได้ ตัวแปรของระยะทางจะกำหนดโดย ตัวแปรของระยะทาง ปกติแล้ว Rang ของเส้น โค้งจะ กำหนด ให้มีค่าตั้งแต่ 0 ถึง 1 ($0 < t < 1$) เพราะมัน เป็นการง่าย ที่จะนำไปใช้งานได้

สมการของ เส้นโค้ง สามารถเขียนให้ง่ายขึ้น โดยการกำหนด เป็น เส้นตรง เล็ก ๆ ต่อไป โดยสมการเส้นตรง เราจะเขียนได้ว่า

$$P(t) = P_1 + (P_2 - P_1)t \qquad ;(0 < t < 1)$$

เมื่อ P (t) คือ ตำแหน่ง Veotor ในแต่ละองค์ประกอบของ P (t) สามารถ เขียนให้อยู่ในรูปของ X (t), Y (t), and Z (t) ดังนี้

$$X(t) = X_1 + (X_2 - X_1)t$$
$$Y(t) = Y_1 + (Y_2 - Y_1)t$$

5.6 Bezier Cuves

เป็นการ สร้างส่วนโค้งโดยอาศัยจุดควบคุม หลาย ๆ จุด อาจเป็น 4 จุด หรือมากกว่า มากำหนดความโค้ง ของเส้นโค้ง ได้ ในลักษณะ ของการ Interpolation และความรู้ทางคณิตศาสตร์บางอย่าง

สมการทาง คณิตศาสตร์ที่เราจะนำมากำหนด Bezier Cuves นี้ได้แก่

$$P(t) = \sum_{i=0}^{n} B_i J_{n,i}(t) \qquad ;(o < t < 1)$$

เมื่อ $J_{n,i}(t)$ คือ อันดับของสมการ Berntein Basis

ตัวอย่าง ที่ n = 3 ค่าสูงสุดของค่า Bernding funotion จะเกิดขึ้นที่ t = i / n จากสมการข้างบน เมื่อเราหาค่า ที่จุด t = i / n

$$J_{n,i}\left(\frac{i}{n}\right) = \binom{n}{i} i^i \frac{(n-i)^{(n-i)}}{n^n}$$

$$J_{3,1}\left(\frac{1}{3}\right) = \frac{4}{9} \quad , \quad J_{3,1}\left(\frac{2}{3}\right) = \frac{4}{9}$$

และ ในการหาค่า ของ จุดแรกของ Bezier Cuves และที่จุด Control เราสามารถกำหนดค่า

$J_{n,i}(t)$ ให้คงที่ได้ เนื่องจากว่า เราวิเคราะห์สมการดังนี้

$$J_{n,0}(0) = \frac{n!(1)(1-0)^{(n-0)}}{n!} = 1 \qquad ; i=0$$

$$J_{n,i}(0) = \frac{n!(0)^i(1-0)^{(n-i)}}{n!} = 0 \qquad ; i=0$$

ดังนั้น $P(0) = B_0 J_{n,0}(0) = B_0$

$$J_{n,n}(1) = \frac{n!(1)^n(0)^{(n-n)}}{n!} = 1 \qquad ; i=n$$

$$J_{n,i}(1) = \frac{n!(t)^i(1-1)^{(n-i)}}{i!(n-i)!} = 0 \qquad ; i=n$$

ดังนั้น $P(1) = B_n J_{n,n}(1) = B_n$

__ตัวอย่าง__ ให้จุด $B_0[1, 1]$, $B_1[2, 3]$, $B_2[4, 3]$ และ $B_3[3, 1]$ จงหาจุดที่จุดต่าง ๆ

จาก

$$P(t) = \sum_{i=0}^{n} B_i J_{n,i}(t) \qquad ;(o<t<1)$$

ขณะที่

$J_{3,0}(t) = (1)t^0(1-t)^3$

$J_{3,1}(t) = (3)t\ (1-t)^2$

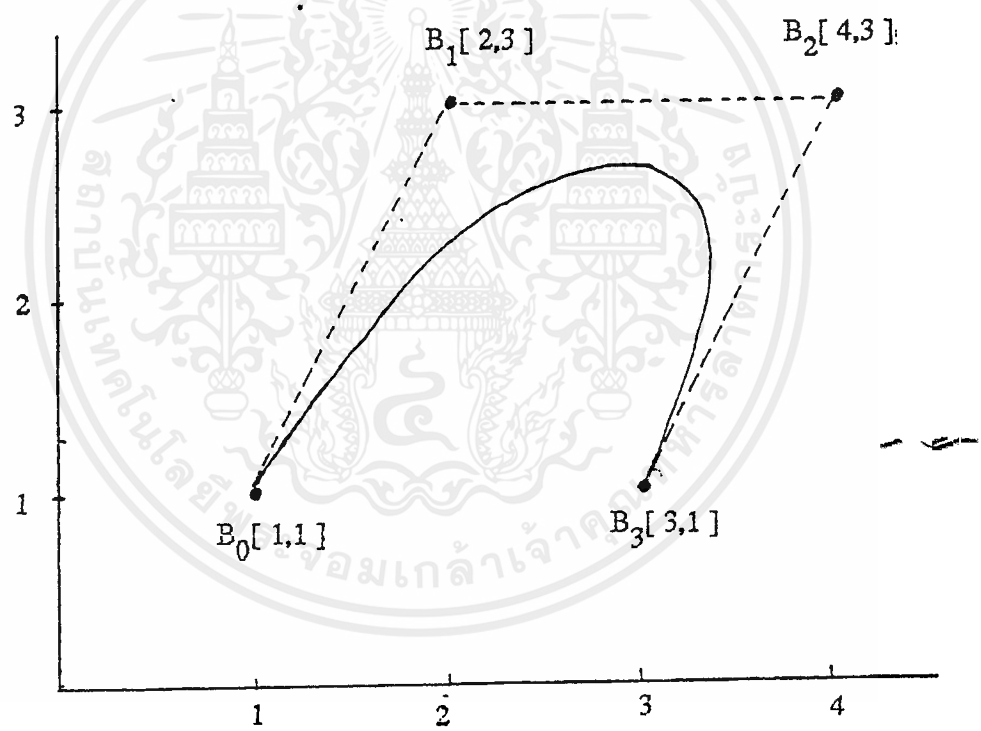
$J_{3,2}(t) = (3)t^2\ (1-t)$

$J_{3,3}(t) = t^3$

ดังนั้น $P(t) = B_0J_{3,0} + B_1J_{3,1} + B_2J_{3,2} + B_3J_{3,3}$

$$= (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)^3 P_2 + t^3 P_3$$

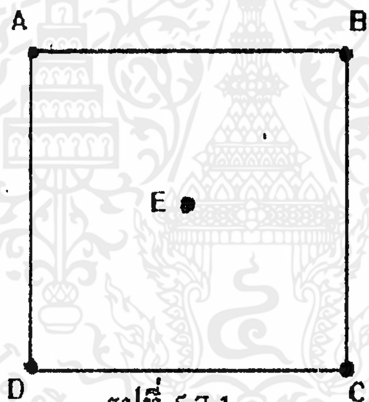
ต้องการหาค่าที่จุดใดก็แทนค่า t ลงในสมการ โดยที่ค่า t จะมีค่าไม่เกิน 1 และมี ค่ามากกว่า 0



รูปที่ 5.6.1 ตัวอย่างของ Bezier Cuves

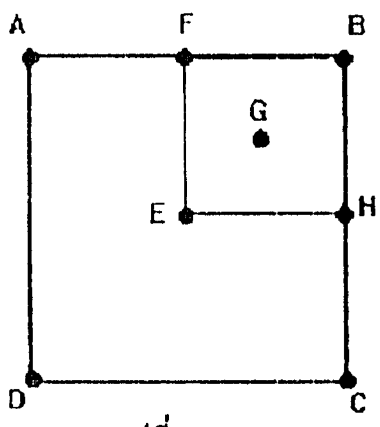
5.7 Anti aliasing technic (Supper sampling)

จากการทดลองเมื่อทำการติดตามรังสีขนาด 320 X 200 จุดแล้วพบว่า มีความระเอียดของภาพน้อยมาก และทำให้มองเห็นว่า ภาพมีรอยหยักตาม ขอบ ซึ่งสาเหตุก็มาจาก ความละเอียด ของพิกเซล มันใหญ่เกินไป ทำให้มองเห็น ว่าภาพ ไม่มี ความต่อเนื่อง ซึ่งเราจะทำการแก้ไข โดยการทำ Supper sampling โดยมีหลักการที่ว่า การติดตามรังสีโดยปรกติเราจะ ติดตาม Pixel ละหนึ่ง รังสีแล้วได้สีมาเลย แต่ในวิธีนี้ จะต้องติดตามรังสี Pixel ละ 5 จุดเป็นอย่างน้อย หลังจากนั้นจะต้องตรวจสอบค่าสี ที่ได้จากมุมทั้ง 5 ถ้าหากว่าที่มุมไหนมีค่าสี แตกต่างไปจากตรงกลาง ก็จะต้องเข้าไปหาค่าสีที่ถูกต้องในส่วนนั้นอีกครั้ง จนกว่าจะครบระดับความลึกของการกำหนด SupperStep จากการทดลองพบว่าไม่ควรทำ SupperStep เกิน 3 ระดับ เนื่องจากต้องสิ้นเปลืองพลังการโปรเซสอย่างมาก



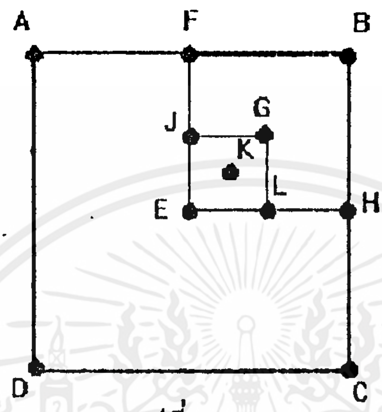
รูปที่ 5.7.1

เมื่อเราเริ่มที่ Pixel E เราจะต้องทำการหาค่าสีที่มุมทั้ง สี่และที่จุดศูนย์กลางด้วย จากนั้นเราจะทำการ เปรียบเทียบสี ระหว่างจุด AE,BE,CE, และ DE สมมุติว่า ที่มุม BE และ CE มีความแตกต่าง เราก็จะทำการหาสีใหม่โดยหาเฉพาะ ระหว่างมุม BE และ CE เท่านั้นดังนี้



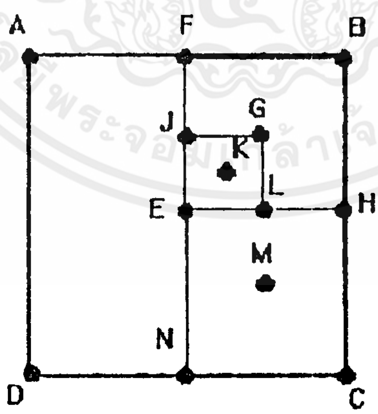
รูปที่ 5.7.2

จากนั้นก็จะพิจารณาที่รูปสี่เหลี่ยมปิดเล็ก ๆ แล้วทำการติดตามรังสีที่มุมทั้ง สี่อีกครั้ง เมื่อได้ค่าสีมาแล้ว ทำการ เปรียบเทียบกันลักษณะเดิมอีก สมมุติว่า ที่มุม GE เท่านั้นที่ แตกต่าง เราจะต้องทำการหาต่อไป ที่สี่เหลี่ยมปิด เล็ก ๆ ระหว่างมุม GE



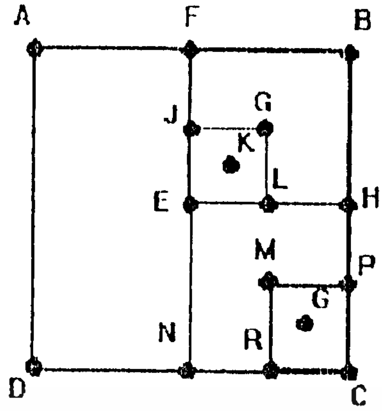
รูปที่ 5.7.3

หลังจากนั้นเราจะพิจารณาที่สี่เหลี่ยมระหว่างจุด CE บ้าง โดยการหาสีใหม่ 3 จุดที่ N,M,และ H จะได้ดังนี้



รูปที่ 5.7.4

แล้วทำการเปรียบเทียบสีที่มุม ทั้งสี่ สมมุติว่า มีสี่จุด CM ต่างกันเราก็ จะหาสีระหว่างมุมทั้งสองอีกครั้งดังนี้



รูปที่ 5.7.5

เมื่อทำการ เปรียบเทียบ สีที่มุมแล้ว สมมุติว่าไม่แตกต่าง เลยจึงถือว่าเป็นอันสิ้นสุด และสามารถ หาค่าสีที่ แท้จริงได้ดังนี้

$$I = (\ 0.5(A+E) + 0.5(D+E)$$

$$+ 0.25(\ 0.5(F+G) + 0.5(B+G) + 0.5(H+G)$$

$$+ 0.25(\ 0.5(J+K) + 0.5(G+K) + 0.5(J+K) + 0.5(E+K) \ )$$

$$+ 0.25(\ 0.5(E+M) + 0.5(H+M) + 0.5(N+M)'$$

$$+ 0.25(\ 0.5(M+Q) + 0.5(P+Q) + 0.5(C+Q) + 0.5(R+Q \ ) \ ) \ )$$

บทที่ 6

RAY TRACING และ ผลการทดลอง

6.1 การกำหนดองค์ประกอบของภาพ

การกำหนดจุดมองภาพ

OBSERVER

LOCATE	X	Y	Z	จุดมองภาพ
LOOKAT	X	Y	Z	จุดที่ต้องการมองไปดู

การกำหนดแหล่งกำเนิดแสง

LAMP

LOCATE	X	Y	Z	จุดที่อยู่ของ แหล่งกำเนิดแสง
INTENSITY	R	G	B	ความเข้มของแสง

การกำหนดวัตถุ

OBJECT

CYLINDER { ทรงกระบอก }

LOCATE	X	Y	Z	จุดกำเนิดของวัตถุ
AMBIEN	R	G	B	ค่าสัมประสิทธิ์ของสีวัตถุ
DIFFUSE	R	G	B	ค่าสัมประสิทธิ์การสะท้อนไม่เป็นระเบียบ
SPECULAR	R	G	B	ค่าสัมประสิทธิ์การสะท้อนเป็นระเบียบ
HEIGTH	Z			ค่าความสูงของทรงกระบอก
RADIUS	R			ค่ารัศมีของทรงกระบอก

CONE { ทรงกรวย }

LOCATE	X	Y	Z	จุดกำเนิดของวัตถุ
AMBIEN	R	G	B	ค่าสัมประสิทธิ์ของสีวัตถุ
DIFFUSE	R	G	B	ค่าสัมประสิทธิ์การสะท้อนไม่เป็นระเบียบ
SPECULAR	R	G	B	ค่าสัมประสิทธิ์การสะท้อนเป็นระเบียบ

HEIGTH	Z			ค่าความสูงของทรงกรวย
RADIUS	R			ค่ารัศมีของทรงกรวย

SPHERE { ทรงกลม }

LOCATE	X	Y	Z	จุดกำเนิดของวัตถุ
AMBIEN	R	G	B	ค่าสัมประสิทธ์ของสีวัตถุ
DIFFUSE	R	G	B	ค่าสัมประสิทธ์การสะท้อนไม่เป็นระเบียบ
SPECULAR	R	G	B	ค่าสัมประสิทธ์การสะท้อนเป็นระเบียบ
RADIUS	R			ค่ารัศมีของทรงกลม

GROUND **{ พื้น }**

LOCATE	X	Y	Z	จุดกำเนิดของวัตถุ
AMBIEN	R	G	B	ค่าสัมประสิทธ์ของสีวัตถุ
DIFFUSE	R	G	B	ค่าสัมประสิทธ์การสะท้อนไม่เป็นระเบียบ
SPECULAR	R	G	B	ค่าสัมประสิทธ์การสะท้อนเป็นระเบียบ
XSIZE	X			ค่าความกว้างทางแกน X
YSIZE	Y			ค่าความกว้างทางแกน Y

END_OBJECT เป็นการจบส่วนของวัตถุ

การกำหนดสีท้องฟ้า

SKYCOLOR	R	G	B	ค่าสัมประสิทธ์ของสีท้องฟ้า

การกำหนด ขอบเขตุของภาพ

WIDTH	Xpixel	ค่าความกว้างของภาพ
HEIGHT	Ypixel	ค่าความสูงของภาพ
FOCAL_LENGHT	Ratio	ค่าอัตราส่วนการขยายภาพ

การจบของการกำหนดภาพโดยคำสั่ง END

*** หมายเหตุ**

R,G, และ B คือค่าสัมประสิทธ์ ของสีแดง,เขียว,และ น้ำเงิน ตามลำดับ

X,Y, และ Z คือตำแหน่งใด ๆ ในระบบ 3 มิติ

```
File name : RAY1.RAY

OBSERVER
   LOCATE          0.0 1500.0 1500.0
   LOOKAT          0.0    0.0    0.0
LAMP
   LOCATE       1500.0 1500.0 1500.0
   INTENSITY       2.0    2.0    2.0
OBJECT
   CYLINDER
     Locate        0.0    0.0    0.0
     Ambi          0.04   0.01   0.01
     Diff          0.4    0.1    0.1
     Spec          0.1
     Heigth      450.0
     Radius      150.0
   SPHERE
     Locate      500.0    0.0  150.0
     Ambi          0.01   0.02   0.03
     Diff          0.1    0.2    0.3
     Spec          0.1
     Radius      150.0
   SPHERE
     Locate        0.0  500.0  150.0
     Ambi          0.03   0.02   0.01
     Diff          0.3    0.2    0.1
     Spec          0.1
     Radius      150.0
   SPHERE
     Locate     -500.0    0.0  150.0
     Ambi          0.01   0.03   0.01
     Diff          0.1    0.3    0.1
     Spec          0.1
     Radius      150.0
   CONE
     Locate        0.0 -500.0    0.0
     Ambi          0.01   0.015  0.02
     Diff          0.1    0.15   0.2
     Spec          0.1
     Heigth      150.0
     Radius      150.0
   GROUND
     Locate        0.0    0.0    0.0
     Ambi          0.01   0.01   0.01
     Diff          0.1    0.1    0.1
     Spec          0.1
     Xsize       650.0
     Radius      650.0
 END_OBJECT

 SKYCOLOR          0.1    0.2    0.4
 WIDTH           320
 HEIGHT          200
 FOCAL_LENGHT      0.046
 END
```

```
File name : RAY2.RAY

OBSERVER
   LOCATE     1500.0  1500.0  700.0
   LOOKAT        0.0     0.0  300.0
LAMP
   LOCATE     1500.0 -1500.0  600.0
   INTENSITY     2.0     2.0    2.0
LAMP
   LOCATE     1290.0  1290.0  700.0
   INTENSITY     2.0     2.0    2.0
OBJECT
   SPHERE
     Locate       0.0     0.0   950.0
     Ambi         0.01    0.03    0.02
     Diff         0.1     0.3     0.2
     Spec         0.4
     Radius     150.0
   CYLINDER
     Locate       0.0     0.0   350.0
     Ambi         0.04    0.01    0.01
     Diff         0.4     0.1     0.1
     Spec         0.5
     Heigth     450.0
     Radius     150.0
   CONE
     Locate       0.0     0.0     0.0
     Ambi         0.01    0.02    0.05
     Diff         0.1     0.2     0.5
     Spec         0.4
     Heigth     350.0
     Radius     150.0
   SPHERE
     Locate    -570.0     0.0     0.0
     Ambi         0.02    0.02    0.02
     Diff         0.2     0.2     0.2
     Spec         0.5
     Radius     150.0
   SPHERE
     Locate       0.0  -570.0     0.0
     Ambi         0.01    0.03    0.04
     Diff         0.1     0.3     0.4
     Spec         0.4
     Radius     150.0
   SPHERE
     Locate     570.0     0.0     0.0
     Ambi         0.02    0.02    0.01
     Diff         0.2     0.2     0.1
     Spec         0.5
     Radius     150.0
   SPHERE
     Locate       0.0   570,0     0.0
     Ambi         0.03    0.02    0.02
     Diff         0.3     0,2     0.2
     Spec         0.5
```

```
      Radius      150.0
    GROUND
      Locate        0.0       0.0     0.0
      Ambi          0.02      0.04    0.01
      Diff          0.2       0.4     0.1
      Spec          0.5
      Xsize       570.0
      Ysize       570.0
END_OBJECT

SKYCOLOR          0.1     0.1     0.2
WIDTH           640
HEIGHT          480
FOCAL_LENGHT      0.46
END
```

File name : RAY3.RAY

```
OBSERVER
   LOCATE        1500.0   1500.0   600.0
   LOOKAT           0.0      0.0   150.0
LAMP
   LOCATE        1500.0  -1500.0  1600.0
   INTENSITY        2.0      2.0     2.0
LAMP
   LOCATE        1290.0   1290.0   700.0
   INTENSITY        2.0      2.0     2.0
OBJECT
   SPHERE
      Locate        0.0      0.0   850.0
      Ambi          0.01     0.03    0.02
      Diff          0.1      0.3     0.2
      Spec          0.4
      Radius      150.0
   CYLINDER
      Locate        0.0      0.0   350.0
      Ambi          0.04     0.01    0.01
      Diff          0.4      0.1     0.1
      Spec          0.5
      Heigth      350.0
      Radius      150.0
   CONE
      Locate        0.0      0.0     0.0
      Ambi          0.01     0.02    0.05
      Diff          0.1      0.2     0.5
      Spec          0.4
      Heigth      350.0
      Radius      150.0
   SPHERE
      Locate      -70.0    170.0   500.0
      Ambi          0.04     0.02    0.01
      Diff          0.4      0.2     0.1
      Spec          0.5
      Radius      150.0
   SPHERE
      Locate      170.0    -70.0   500.0
      Ambi          0.01     0.02    0.05
      Diff          0.1      0.2     0.5
      Spec          0.4
      Radius      150.0
   SPHERE
      Locate     -570.0      0.0     0.0
      Ambi          0.02     0.02    0.02
      Diff          0.2      0.2     0.2
      Spec          0.5
      Radius      150.0
   SPHERE
      Locate        0.0   -570.0     0.0
      Ambi          0.01     0.03    0.04
      Diff          0.1      0.3     0.4
      Spec          0.4
```

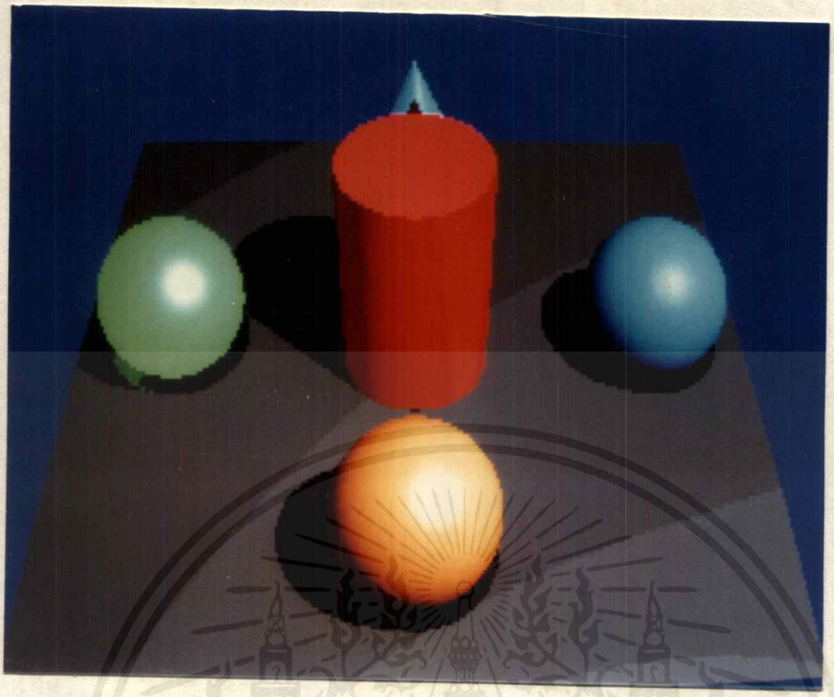
```
      Radius     150.0
    SPHERE
      Locate     570.0      0.0     0.0
      Ambi         0.02     0.02    0.01
      Diff         0.2      0.2     0.1
      Spec         0.5
      Radius     150.0
    SPHERE
      Locate       0.0    570.0     0.0
      Ambi         0.03     0.02    0.02
      Diff         0.3      0.2     0.2
      Spec         0.5
      Radius     150.0
    GROUND
      Locate       0.0      0.0     0.0
      Ambi         0.02     0.04    0.01
      Diff         0.2      0.4     0.1
      Spec         0.5
      Xsize      570.0
      Ysize      570.0
END_OBJECT

SKYCOLOR          0.1     0.1     0.2
WIDTH           800
HEIGHT          600
FOCAL_LENGHT      0.76
END
```

File name : RAY4.RAY

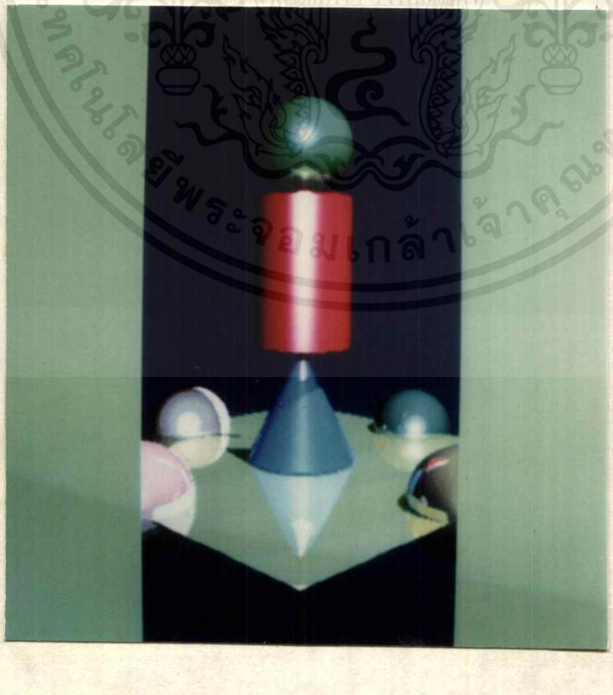
```
OBSERVER
   LOCATE        1500.0  1500.0   400.0
   LOOKAT           0.0     0.0     0.0
LAMP
   LOCATE        1500.0 -1500.0   600.0
   INTENSITY        2.0     2.0     2.0
LAMP
   LOCATE        1290.0  1290.0   700.0
   INTENSITY        2.0     2.0     2.0
OBJECT
   SPHERE
     Locate         0.0     0.0  1090.0
     Ambi           0.01    0.02    0.03
     Diff           0.1     0.3     0.2
     Spec           0.3
     Radius       150.0
   CYLINDER
     Locate         0.0     0.0   590.0
     Ambi           0.04    0.01    0.01
     Diff           0.4     0.1     0.1
     Spec           0.5
     Heigth       350.0
     Radius       450.0
   SPHERE
     Locate         0.0     0.0   470.0
     Ambi           0.01    0.02    0.03
     Diff           0.1     0.3     0.2
     Spec           0.5
     Radius       120.0
   CONE
     Locate         0.0     0.0     0.0
     Ambi           0.01    0.02    0.05
     Diff           0.1     0.2     0.5
     Spec           0.4
     Heigth       350.0
     Radius       150.0
   SPHERE
     Locate      -570.0     0.0     0.0
     Ambi           0.02    0.02    0.02
     Diff           0.2     0.2     0.2
     Spec           0.5
     Radius       150.0
   SPHERE
     Locate         0.0  -570.0     0.0
     Ambi           0.01    0.03    0.04
     Diff           0.1     0.3     0.4
     Spec           0.4
     Radius       150.0
   SPHERE
     Locate       570.0     0.0     0.0
     Ambi           0.02    0.02    0.01
     Diff           0.2     0.2     0.1
     Spec           0.5
```

```
      Radius     150.0
    SPHERE
      Locate        0.0    570.0     0.0
      Ambi          0.03     0.02    0.02
      Diff          0.3      0.2     0.2
      Spec          0.5
      Radius     150.0
    GROUND
      Locate        0.0      0.0     0.0
      Ambi          0.02     0.04    0.01
      Diff          0.2      0.4     0.1
      Spec          0.5
      Xsize      570.0
      Ysize      570.0
END_OBJECT

SKYCOLOR            0.1      0.1     0.2
WIDTH            1024
HEIGHT            768
FOCAL_LENGHT        0.96

END
```

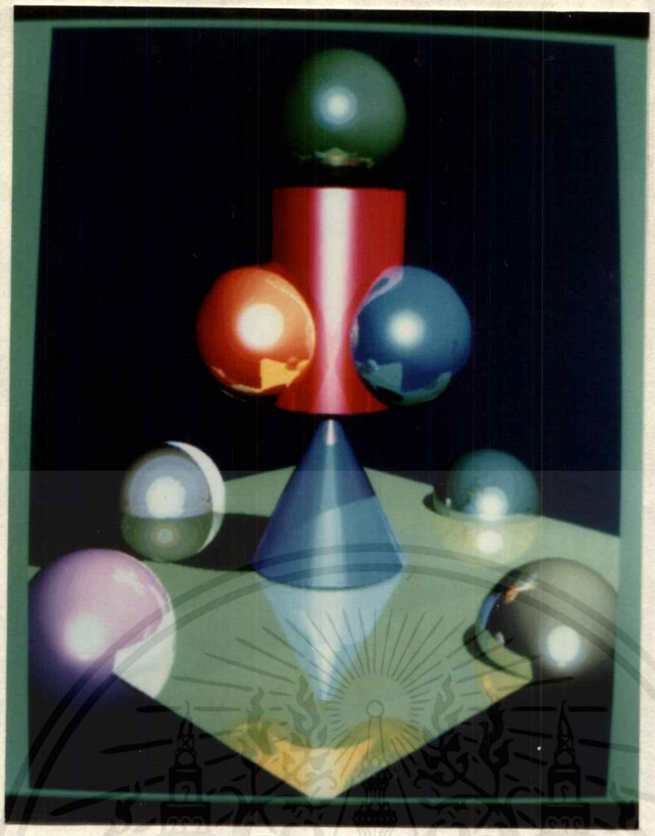
6.2 ผลการทดลอง



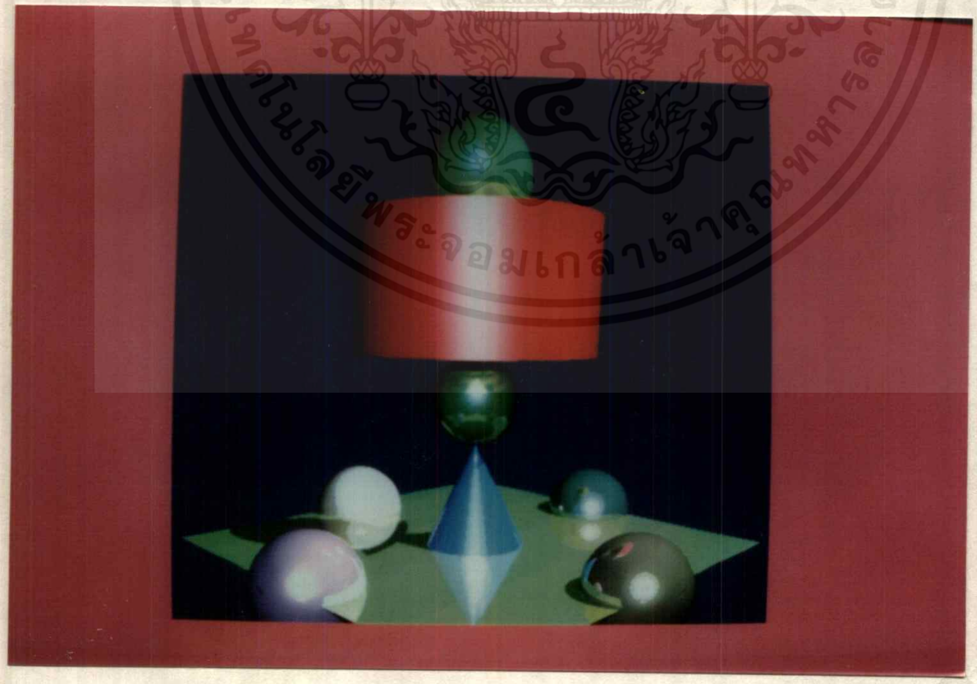
รูปที่ 1 จากไฟล์ ที่ชื่อว่า RAY1.RAY



รูปที่ 2 จากไฟล์ ที่ชื่อว่า RAY2.RAY



รูปที่ 3 จากไฟล์ ที่ชื่อว่า RAY3.RAY



รูปที่ 4 จากไฟล์ ที่ชื่อว่า RAY4.RAY

6.3 สรุปผลการทดลอง

เมื่อศึกษา หลักการต่าง ๆ มาแล้วก็ได้ทำการ ทดลองตามทฤษฎี ต่าง ๆ ดูก็พบว่า มีอุปสรรคนานานับประการ กล่าวคือหาข้อมูลยากมากหรือหาแทบไม่ได้เลย ถ้าเป็น ข้อมูลภาษาไทยส่วนที่พอจะหาได้ก็ล้วนแต่เป็น Text book เป็นส่วนมาก เมื่อได้ลอง พยายาม อ่านดูแล้วก็พบแต่ น้ำหาได้พบ เนื้อไม่ เพราะว่าส่วนใหญ่จะให้มาแต่ Concept หลัก ๆ เท่านั้น ทำให้การพัฒนาโปรแกรม เป็นไปได้อย่าง ลำบาก ลำบน กว่าจะมาเป็น โปรแกรมทดลองเวอร์ชั่นแรกได้นั้น ก็ล่วงเลย มาเป็นเวลา กว่าสองเดือนแล้ว เมื่อได้ทดลองขั้นตอนต่าง ๆ ก็ ประสบพบเจอแต่ ปัญหา ทั้งนั้น อาทิ เช่น การแสดงผลของ การ์ดแสดงผล (ในที่นี้ได้ใช้ การ์ด SVGA ของ Trident 8900 III) เนื่องจากต้องการ ความสามารถ ในการแสดงสี และ รายละเอียดที่มากที่สุด เมื่อเราต้องการเล่นในโหมดของ Supper VGA นั้น ใน Libraly ของ ตัวแปรภาษานั้นมิได้มีให้ท่านแต่อย่างใด เราจึงต้องทำ โมดูล ภาษา Assembly ขึ้นมาเพื่อนำมา ใช้ในการเล่นในโหมด SVGA ได้ หลังจากที่เล่นสีได้แล้ว เราก็เจอปัญญหาอีกเมื่อ โปรแกรมสังเคราะห์ภาพนั้นให้เอาท์พุท ออกมามีสีสัน มากมายถึง สาม ถึง แปดพัน สี เลย แล้วเราก็ไม่มีปัญญาในการที่จะซื้อการ์ด ที่เล่นสีได้มาก ขนาดนั้นได้ ในเวลา นั้น จึงต้องหาวิธีในการเลือกสีมาใช้งาน เพียง 256 สีเท่านั้น โดยการหาว่า สีใดที่มีความถี่ ในการใช้งานมากที่สุดก็เลือกเอา สีพวกนั้นมา 256 สีแรกที่มีความถี่ มากที่สุด มา แล้วทำการเซ็ท ค่าสีภายในการ์ด VGA ก่อนที่จะทำการวาดภาพต่าง ๆ ลงไป ส่วนปัญหาต่อไปก็คือ การคิดของโปรแกรมนั้น ต้องใช้คณิตศาสตร์ จำพวก เวคเตอร์ ซึ่งถาเราลอง พิจารณาดู ว่า การ บวก ลบ คูณ หาร เวคเตอร์นั้นจะต้องทำโปรแกรม มาโดย เฉพาะ เมื่อเราจะใช้ โปรแกรมภาษา C ธรรมดา มันจะทำให้อ่านโปรแกรมได้ยาก และ งง เราจึงต้องพิจารณาเอา การเขียนโปรแกรมเชิง วัตถุ หรือ Objeot Oriented Programming มาใช้ในการเขียนโปรแกรมทั้งหมด เมื่อ การเขียนโปรแกรมเชิง วัตถุ หรือ Objeot Oriented Programming เป็นของใหม่ มันก็ย่อมที่จะเสียเวลาในการพัฒนาใน ช่วงแรก บ้าง หลังจากนั้น การเขียนโปรแกรม ในลำดับต่อไปนั้น ก็ล้วนแต่ต้องใช้คณิตศาสตร์ทั้งสิ้น ไม่ว่าจะเป็น Vector หรือ Parametric ต่าง ๆ ทำ ให้เราต้อง สูญเสียเวลาไปในส่วนนี้มากพอสมควร และขบวนการติดตามรังสี ถ้าหากใครได้สัมผัสกับมัน แล้วจะต้องรู้ว่า มันจะต้อง สิ้นเปลืองพลังงาน ในการ ประมวลผล มากมาย มหาศาล เพียงใด การทดลองใน ภาคนี้เรา พยายาม ที่จะใช้เครื่องที่มีพลัง สูง ๆ อย่าง 486 เป็นอย่าง น้อย และถึงแม้ ว่าเราจะใช้ 486 มันก็คงจะเทียบกับ WorkStation มิได้ การทดลอง และ การทำ โครงงานนี้ได้ เสียเวลา ไปในการ รอคอย การคำนวณของโปรแกรมไป ประมาณ 70 % ของเวลาทั้งหมด

แนวทางในการ พัฒนาต่อไป นั้นยังมีอีกมากมาย ทั้งนี้เนื่องจาก การทำ Ray Tracing นี้ต้องการ รายละเอียด ต่าง ๆ มากมาย เช่น การกำหนด ลักษณะของผิว การกำหนดลาย การกำหนดสี และอื่น ๆ อีกมากมาย และยังมีการ กำหนด วัตถุในแบบ B-Spilne ซึ่งเป็นเรื่อง ที่ควรจะ ให้ความสนใจเป็นอย่างมาก เพราะว่า ในต่างประเทศเขาไปถึงไหน ต่อ ไหนแล้ว ทาง ผู้จัดทำ มีความประสงค์ อย่างแรงกล้า ที่จะทำการ วิจัยต่อไป อย่างไม่หยุดยั้งถ้ามีโอกาส เพื่อเป็นแนวทางใน การก้าว ไปข้างหน้า กับเทคโนโลยี ชั้นสูง ของ คอมพิวเตอร์กราฟฟิค ของชนรุนหลัง

กิติกรรมประกาศ

ขอขอบพระคุณ ท่านอาจารย์ เกษตร์ ศิริสันติสัมฤทธิ์ ที่ได้ให้โอกาส และให้คำปรึกษา ในการทำ ปริญญานิพนธ์ ครั้งนี้ ขอขอบพระคุณ บิดา มารดา ที่ให้กำเนิด ขอขอบคุณเทคโนโลยีต่าง ๆ ที่ช่วย ผลักดัน ขอขอบพระคุณ วิลเลี่ยม บิลล์ เกทส์ และ สตีบ จอบส์ ที่ เป็นบุคคลที่ช่วย ผลักดันให้ ข้าพเจ้า ได้เข้ามาสู่โลกแห่ง จินตนาการ ของ คอมพิวเตอร์กราฟฟิค ขอขอบพระคุณ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง ที่ได้ให้โอกาส ข้าพเจ้าได้เข้ามา ศึกษา เล่าเรียน ขอขอบพระคุณ เจ้าหน้าที่ ห้อง COMPUTER LAB ที่เปิดห้อง ให้ใช้ เครื่อง 486 ขอขอบพระคุณ เจ้าหน้าที่ ห้องสมุด ที่ให้ บริการให้ยืม หนังสือ และขอขอบคุณ เพื่อน ๆ ที่ คอยช่วยเหลือ และ ให้กำลังใจ ตลอดจนแม่ค้าโรงอาหาร และผู้ที่ให้ความช่วยเหลือในด้านต่าง ๆ ไว้ ณ ที่นี้ด้วย

เอกสาร อ้างอิง

[1] มนัส สังวรศิลป์,"วารสารการประชุมวิชาการ วิศวกรรมไฟฟ้าครั้งที่ 10 เล่มที่ 2"

[2] Alan Watt,"Fandamental of three dimensional Computer graphics",U.S.A., Addison-Wesley 1989

[3] J.Alan Adams,"Mathematical Element for Computer Graphics", McGraw-Hill, 1990

[4] Andrew S. Glassner," An Introduction to Ray Tracing ", Academic Press, 1991

[5] ผ.ศ. เปี่ยมศรี สุวรรณฎฏ," คณิตศาสตร์ ชั้นสูง สำหรับวิศวกร ", ศูนย์สื่อเสริมกรุงเทพม, 2532