โปรแกรมจำลองภาพ 3 มิติ ของสมการในพีชคณิตที่มีกำลังสอง

MODELING 3 – DIMENSION OF QUADRATIC EQUATION

กวีพจน์ ลีลาบุญเปี่ยม
รัตนาภรณ์ โพธิ์แก้ว
ศิริวรรณ รักวงษ์

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2545

MODELING 3– DIMENSION OF QUADRATIC EQUATION

KAVEEPOJ LEELABOONPEIM

RATTANAPORN PHOKAEW

SIRIWAN RAKWONG

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT 'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2002

หัวข้อปัญหาพิเศษ	โปรแกรมจำลองภาพ 3 มิติ ของสมการในพีชคณิตที่มีกำลังสอง
	MODELING 3–DIMENSION OF QUADRATIC EQUATION
ชื่อนักศึกษา	นายกวีพจน์ ลีลาบุญเปี่ยม 42050004
	นางสาวรัตนาภรณ์ โพธิ์แก้ว 42050038
	นางสาวศิริวรรณ รักวงษ์ 42050046
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชา	คณิตศาสตร์ประยุกต์
อาจารย์ที่ปรึกษา	รศ.ภัคคินี ชิตสกุล
	รศ.ธีรวัฒน์ ประกอบผล

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิชาคณิตศาสตร์ประยุกต์ ปีการศึกษา 2545

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	ผศ.ไพรบูลย์ พันธรักษ์พงษ์	
กรรมการ	อ.พรชัย ชัยสนิท	
กรรมการและอาจารย์ที่ปรึกษา	รศ.ภัคคินี ชิตสกุล	
กรรมการและอาจารย์ที่ปรึกษา	รศ.ธีรวัฒน์ ประกอบผล	ธีรวัฒน์ ประกอบผล

(ผู้ช่วยศาสตราจารย์ไพรบูลย์ พันธรักษ์พงษ์)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

หัวข้อปัญหาพิเศษ โปรแกรมจำลองภาพ 3 มิติ ของสมการในพีชคณิตที่มีกำลังสอง

ชื่อนักศึกษา นายกวีพงษ์ ลีลาบุญเปี่ยม 42050004

นางสาวรัตนาภรณ์ โพธิ์แก้ว 42050038

นางสาวศิริวรรณ รักวงษ์ 42050046

ปริญญา วิทยาศาสตรบัณฑิต

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สาขาวิชา คณิตศาสตร์ประยุกต์

ปีการศึกษา 2545

อาจารย์ที่ปรึกษา รศ.ภัคคินี ชิตสกุล

รศ.ธีรวัฒน์ ประกอบผล

บทคัดย่อ

ปัญหาพิเศษฉบับนี้ เป็นการนำความรู้ทางคณิตศาสตร์ มาประยุกต์ใช้ทางคอมพิวเตอร์กราฟฟิก เพื่อสร้าง ซอฟต์แวร์ สำหรับจำลองภาพ 3 มิติ ของสมการในพีชคณิตที่มีกำลังสองและรูปทรงวงแหวน

การพัฒนาโปรแกรมการจำลองภาพ 3 มิติดังกล่าว ใช้ภาษา Visual C++ 6.0 ในการเขียนโปรแกรม โดยนำความรู้ทางคณิตศาสตร์มาประยุกต์มาใช้ในการเขียนอัลกอริทึมเพื่อสร้างภาพจำลอง โดยใช้ความสัมพันธ์ ของค่า ϕ และ θ ของสมการกับแกนในแนวต่างๆ มาประยุกต์ในการหมุนภาพ ในส่วนของการย่อ/ขยายนั้น จะใช้ความรู้เกี่ยวกับการคูณกันของเมตริกซ์ สำหรับการแสดงภาพ 3 มิติบนหน้าจอ 2 มิติ จะอาศัยความรู้เกี่ยวกับ พีธากอรัส มาใช้ในการคำนวณหาจุดแสดงผลของค่าในแกนขวาง

Special Project Title	MODELING 3–DIMENSION OF QUADRATIC EQUATION
Students	Mr.Kaveepoj Leelaboonpiem 42050004
	Miss Rattanaporn Phokaew 42050038
	Miss Siriwan Rakwong 42050046
Degree	Bachelor of Science
Department	Mathematics and Computer science , Faculty of Science
Programme	Applied Mathematics
Academic Year	2002
Special Project Advisor	Assoc.Prof. Pakkinee Chitsakul
	Assoc.Prof. Teerawat Prakobpol

ABSTRACT

This special problem is design software for modeling 3 Dimensions of quadratic equation and torus which applied mathematics to computer graphic.

A program is developed by Visual C++ 6. We use relation of ϕ and θ of equation with each axis for rotation, for zoom in and zoom out image we use dot product of matrix, for showing 3 Dimensions image on screen, that is 2 Dimensions, we use pythagorus 's theorem to compute a point that show of across axis.

กิตติกรรมประกาศ

ปัญหาพิเศษพิเศษฉบับนี้ สำเร็จลงได้ด้วยดี ก็เพราะคำแนะนำและกำลังใจที่ดีจากบุคคลรอบข้าง โดยเฉพาะอย่างยิ่ง อาจารย์ที่ปรึกษา

รศ.ภัคคินี ชิตสกุล

รศ.ธีรวัฒน์ ประกอบผล

ที่ได้ให้แนวทางในการทำปัญหาพิเศษตลอดจนคำปรึกษาอันก่อให้เกิดแนวความคิด ที่สามารถแก้ไขปัญหาต่าง ๆ ที่เกิดขึ้นในระหว่างการทำปัญหาพิเศษฉบับนี้ นอกจากนี้ยังช่วยแนะแนวทางในการดำเนินงานและตรวจทานแก้ไขด้วยความเอาใจใส่เป็นอย่างยิ่ง

คณะผู้จัดทำขอกราบขอบพระคุณอาจารย์ทุกท่านที่ประศาสน์วิชาความรู้ทั้งในภาคทฤษฎีและภาคปฏิบัติแก่ผู้จัดทำ จนกระทั่งปัญหาพิเศษฉบับนี้สัมฤทธิ์ผลได้ด้วยดีทุกประการ

ขอขอบพระคุณ

คณะผู้จัดทำ

มีนาคม 2545

สารบัญ

สารบัญภาพ

ภาพ หน้า

บทที่ 2 ความรู้พื้นฐานและทฤษฎี

2.1 ทฤษฎีทางคณิตศาสตร์

2.2 การแปลงภาพเรขาคณิตในระบบภาพ 2 มิติ

2.3 ระบบภาพ 3 มิติ

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหาพิเศษ

เนื่องด้วยการจำลองภาพรูปทรง 3 มิตินั้น มนุษย์สามารถจำลองขึ้นมาได้โดยการทำ Model (แบบจำลอง) เพื่อทำให้สามารถมองเห็นได้ทุกด้าน ซึ่งสามารถทำได้โดยไม่ยุ่งยากมากนักสำหรับรูปสี่เหลี่ยมลูกบาศก์ หรือรูปทรงเรขาคณิตทั่วไป แต่สำหรับรูปที่มีความซับซ้อนมากขึ้น เช่น รูปที่มีหลายเหลี่ยม หรือมีลักษณะโค้งมีความลึกจะทำได้ค่อนข้างยุ่งยากและต้องใช้เวลามากในการสร้าง เสียงบประมาณในการสร้าง และบำรุงรักษา มนุษย์เราจึงนำเครื่องคอมพิวเตอร์มาช่วย โดยการประยุกต์ความรู้ในเรื่องคอมพิวเตอร์กราฟฟิก แล้วพัฒนาสร้างเป็น Software (โปรแกรมใช้งาน) ดังที่มีใช้อยู่ในปัจจุบันแต่จะมีราคาค่อนข้างสูงเพราะมีเรื่องของลิขสิทธิ์มาเกี่ยวข้องและมีความซับซ้อนในการใช้งาน ดังนั้นการศึกษาและพัฒนาจึงทำได้ยาก ทั้งที่เป็นสิ่งที่น่าสนใจมาก

ปัญหาพิเศษฉบับนี้ได้นำความรู้ทางด้านคณิตศาสตร์ มาประยุกต์เพื่อสร้างภาพจำลอง 3 มิติ เพื่อให้เห็นภาพที่มีลักษณะที่เหมือนจริงมากที่สุด และสามารถนำไปประยุกต์ใช้งานในด้านต่าง ๆ ได้อย่างเหมาะสม

1.2 วัตถุประสงค์ของปัญหาพิเศษ

1) ใช้ความรู้ทางด้านคณิตศาสตร์มาประยุกต์ในการสร้างภาพจำลอง 3 มิติโดยการเขียนโปรแกรมด้วยภาษา Visual C++

2) สร้างภาพจำลอง 3 มิติโดยมีลักษณะใกล้เคียงภาพจริงพร้อมทั้งสามารถหมุนภาพดูได้

3) เพื่อเป็นพื้นฐานสำหรับการพัฒนางานในด้านคอมพิวเตอร์กราฟฟิกต่อไป

1.3 ขอบเขตของปัญหาพิเศษ

1) จำลองภาพ 3 มิติ จากความรู้ทางด้านคณิตศาสตร์และคอมพิวเตอร์กราฟฟิก

2) สามารถหมุนภาพที่จำลอง 3 มิติได้ เพื่อทำให้มองได้หลายมุมมอง

3) ปรับปรุงภาพจำลอง 3 มิติ ให้มีลักษณะใกล้เคียงวัตถุจริงมากขึ้น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) เพื่อให้เกิดความรู้ความเข้าใจ ในการสร้างซอร์ฟแวร์สำเร็จรูป จากภาษาการเขียนโปร - แกรมแบบภาษา Visual C++

2) เพื่อให้เกิดความรู้ความเข้าใจ ในการนำความรู้ทางคณิตศาสตร์ มาประยุกต์ใช้งานในการสร้างภาพจำลอง 3 มิติ

3) สามารถนำภาพจำลอง 3 มิติที่สร้างขึ้นนี้ไปประยุกต์ใช้งานในด้านต่างๆได้

1.5 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

เครื่องไมโครคอมพิวเตอร์ โดยมีขนาดเครื่อง ดังนี้

- CPU Pentium 4 2.4A MHz
- RAM 256 MB
- HARDDISK 60 GB
- การ์ดแสดงผล 64 บิต

1.6 ขั้นตอนในการดำเนินงาน

1) ศึกษาเนื้อหาทางคณิตศาสตร์เกี่ยวกับสมการ Quadratic , สมการของ Torus และการแปลงเรขาคณิต , ความรู้เกี่ยวกับคอมพิวเตอร์กราฟิกและการเขียนโปรแกรมด้วยภาษา Visual C++

2) ออกแบบต้นแบบของโปรแกรมระบบงานกำหนดและแยกส่วนต่างๆของระบบที่ต้องดำเนินการสร้าง

3) ดำเนินการสร้างและพัฒนาโปรแกรมที่ได้ออกแบบ

4) ทำการทดสอบโปรแกรมที่ได้สร้างไว้และศึกษาถึงปัญหาที่พบในการสร้างและดำเนินการแก้ไข

5) สรุปประสิทธิภาพและปัญหาที่เกิดขึ้น

6) จัดทำเอกสารประกอบโปรแกรม

1.7 แผนการทำงานของระบบ

ID	Tasks Name	June				July				August				September				October				November				December				January				February				March			
		1	8	15	25	1	10	22	27	8	16	22	28	1	7	15	25	1	10	20	27	7	12	24	28	1	14	20	26	1	12	20	28	1	15	20	26	1	10	15	15
1	ศึกษาปัญหาและที่มาของปัญหาพิเศษ																																								
2	ศึกษาเครื่องมือ software, ขอบเขตและความเป็นไปได้																																								
3	จัดทำแบบขออนุมัติ ทำปัญหาพิเศษ																																								
4	ศึกษา Software ที่ใช้																																								
5	ศึกษาทฤษฎีและหลักการที่เกี่ยวข้อง																																								
6	ออกแบบ interface, algorithm ของภาพ																																								
7	เขียนโปรแกรมโดยใช้ Visual C++																																								
8	ทดสอบและแก้ไขโปรแกรม																																								
9	สรุปโครงงานปัญหาพิเศษ																																								
10	จัดทำเอกสารประกอบโครงงานปัญหาพิเศษ																																								

บทที่ 2
ความรู้พื้นฐานทางคอมพิวเตอร์กราฟฟิก

2.1 ทฤษฎีทางคณิตศาสตร์

รูปทรงที่จะจำลองในโปรแกรมนี้ มี 6 รูป คือ

2.1.1 ทฤษฎีทางคณิตศาสตร์สำหรับทรงรี (Ellipsoid)

เขียนในรูปสมการ

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}=1 \qquad \text{โดยที่} \quad a>0\,,\,b>0\,,\,c>0$$

พิจารณารูปทรงรี ดังรูป

รูปที่ 2.1-1 ทรงรี (ellipsoid)

พิจารณารายละเอียดของทรงรี จะได้ว่า

จุดตัดแกน จุดตัดแกน x คือ (a,0,0) และ (-a,0,0)

จุดตัดแกน y คือ (0,b,0) และ (0,-b,0)

จุดตัดแกน z คือ (0,0,c) และ (0,0,-c)

สมมาตร พื้นผิวมีสมมาตรกับจุดกำเนิด

พื้นผิวมีสมมาตรกับแกนทั้งสาม (คือแกน x,y และ z)

พื้นผิวมีสมมาตรกับระนาบทั้งสาม (คือระนาบ xy,yz และ xz)

รอยตัด ตัดด้วยระนาบ xy ได้รูปวงรี $\dfrac{x^2}{a^2}+\dfrac{y^2}{b^2}=1,\ z=0$

ตัดด้วยระนาบ yz ได้รูปวงรี $\dfrac{y^2}{b^2}+\dfrac{z^2}{c^2}=1,\ x=0$

ตัดด้วยระนาบ xz ได้รูปวงรี $\dfrac{x^2}{a^2}+\dfrac{z^2}{c^2}=1,\ y=0$

ตัดด้วยระนาบที่ขนานกับระนาบพิกัด

ตัดด้วยระนาบที่ขนานกับระนาบ xy คือ ตัดด้วยระนาบ $z=k$ จะได้รูปวงรี

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1 \quad \frac{k^2}{c^2},\ z=k \quad \text{เมื่อ} \quad |k|<c$$

ตัดด้วยระนาบที่ขนานกับระนาบ yz คือ ตัดด้วยระนาบ $x=k$ จะได้รูปวงรี

$$\frac{y^2}{b^2}+\frac{z^2}{c^2}=1 \quad \frac{k^2}{a^2},\ x=k \quad \text{เมื่อ} \quad |k|<a$$

ตัดด้วยระนาบที่ขนานกับระนาบ xz คือ ตัดด้วยระนาบ $y=k$ จะได้รูปวงรี

$$\frac{x^2}{a^2}+\frac{z^2}{c^2}=1 \quad \frac{k^2}{b^2},\ y=k \quad \text{เมื่อ} \quad |k|<b$$

ส่วนขอบเขตของรูปพื้นผิวทรงรีได้จากการพิจารณารอยตัด คือ

ขอบเขตของ x คือ $[-a,a]$ หรือ x ที่สอดคล้องกับอสมการ $|x|<a$

ขอบเขตของ y คือ $[-b,b]$ หรือ y ที่สอดคล้องกับอสมการ $|y|<b$

ขอบเขตของ z คือ $[-c,c]$ หรือ z ที่สอดคล้องกับอสมการ $|z|<c$

จัดให้อยู่ในรูปของรัศมี มุมละติจูด และ มุมลองติจูดได้ดังนี้

$x=a\cos\phi\cos\theta$ โดยที่ $-\dfrac{\pi}{2}\le\phi\le\dfrac{\pi}{2}$

$y=b\cos\phi\sin\theta$ โดยที่ $-\pi\le\theta\le\pi$

$z=c\sin\phi$

<u>หมายเหตุ</u> 1. รูปทั่วไปของทรงรี คือ

$$Ax^2 + By^2 + Cz^2 = 1 \quad \text{โดยที่} \quad A > 0, B > 0, C > 0$$

2. รูปสมการพื้นผิวทรงรี คือพื้นผิวทรงกลม ถ้า $A = B = C$ หรือ $a = b = c$

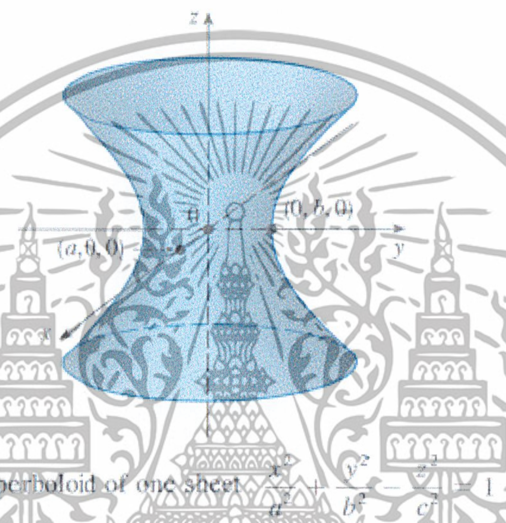
3. ในการพิจารณาการตัดด้วยระนาบที่ขนานกับระนาบพิกัด xy คือ $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 - \frac{k^2}{c^2}, z = k$ เมื่อ $|k| < c$ จะเห็นว่าถ้า $|k| = c$ คือจุดตัดของพื้นผิวและแกนพิกัด และถ้า $|k| > c$ จะไม่มีรอยตัดของพื้นผิว

2.1.2 ทฤษฎีทางคณิตศาสตร์สำหรับไฮเปอร์โบลอยด์ชนิดผิวเดียว (Hyperboloid of one sheet)

เขียนในรูปสมการ

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=1 \quad \text{โดยที่} \quad a>0\,,b>0\,,c>0$$

พิจารณารูปไฮเปอร์โบลอยด์ชนิดผิวเดียว ดังรูป



รูปที่ 2.1-2 ไฮเพอร์โบลอยด์ชนิดผิวเดียว (elliptic hyperboloid of one sheet)

พิจารณารายละเอียดของไฮเพอร์โบลอยด์ชนิดผิวเดียว จะได้ว่า

จุดตัดแกน จุดตัดแกน x คือ $(a,0,0)$ และ $(-a,0,0)$

จุดตัดแกน y คือ $(0,b,0)$ และ $(0,-b,0)$

พื้นผิวไม่ตัดแกน z

สมมาตร พื้นผิวมีสมมาตรกับจุดกำเนิด

พื้นผิวมีสมมาตรกับแกน พิกัดทั้งสาม

พื้นผิวมีสมมาตรกับระนาบพิกัดทั้งสาม

รอยตัด ตัดด้วยระนาบ xy ได้รูปวงรี $\frac{x^2}{a^2}+\frac{y^2}{b^2}=1, z=0$

ตัดด้วยระนาบ yz ได้รูปไฮเพอร์โบลา $\frac{y^2}{b^2}-\frac{z^2}{c^2}=1, x=0$

ตัดด้วยระนาบ xz ได้รูปไฮเพอร์โบลา $\frac{x^2}{a^2}-\frac{z^2}{c^2}=1, y=0$

ตัดด้วยระนาบที่ขนานกับระนาบพิกัด

ตัดด้วยระนาบที่ขนานกับระนาบ xy คือ ตัดด้วยระนาบ $z=k$ จะได้รูปวงรี $\frac{x^2}{a^2}+\frac{y^2}{b^2}=1+\frac{k^2}{c^2}$ สำหรับทุกๆ ค่า k และรูปวงรีจะมีขนาดโตขึ้นเมื่อ $|k|$ มีค่าเพิ่มขึ้นนั่นคือบนระนาบ xy วงรีมีขนาดเล็กที่สุดแล้วค่อยๆ โตขึ้นเมื่อตัดด้วยระนาบที่ขนานกับระนาบ xy

ตัดด้วยระนาบที่ขนานกับระนาบ yz คือ ตัดด้วยระนาบ $x=k$ จะได้ว่า

ได้รูปไฮเพอร์โบลา $\frac{y^2}{b^2}-\frac{z^2}{c^2}=1-\frac{k^2}{a^2}, x=k$ เมื่อ $|k|<a$

ได้รูปไฮเพอร์โบลา $\frac{z^2}{c^2}-\frac{y^2}{b^2}=\frac{k^2}{a^2}-1, x=k$ เมื่อ $|k|>a$

ได้เส้นตรง $\frac{y^2}{b^2}-\frac{z^2}{c^2}=0$ หรือ $y=\pm\frac{b}{c}z$ เมื่อ $|k|=a$

ตัดด้วยระนาบที่ขนานกับระนาบ xz (คือ $y=k$) จะได้ว่า

ได้รูปไฮเพอร์โบลา $\frac{x^2}{a^2}-\frac{z^2}{c^2}=1-\frac{k^2}{b^2}, y=k$ เมื่อ $|k|<b$

ได้รูปไฮเพอร์โบลา $\frac{z^2}{c^2}-\frac{x^2}{a^2}=\frac{k^2}{b^2}-1, y=k$ เมื่อ $|k|>b$

ได้เส้นตรง $\frac{x^2}{a^2}-\frac{z^2}{c^2}=0$ หรือ $x=\pm\frac{a}{c}z$ เมื่อ $|k|=b$

ส่วนขอบเขตของรูปพื้นผิวได้จากการพิจารณารอยตัด คือ

ขอบเขตของ x บนพื้นผิว คือ $R-(-a,a)$

ขอบเขตของ y บนพื้นผิว คือ $R-(-b,b)$

ขอบเขตของ z บนพื้นผิว คือ $(-\infty,\infty)$

จัดให้อยู่ในรูปของรัศมี มุมละติจูด และ มุมลองติจูดได้ดังนี้

$x = a\cos\theta\cosh\phi$ โดยที่ $0 \le \theta \le 2\pi$

$y = b\sin\theta\sinh\phi$ โดยที่ $-\pi \le \phi \le \pi$

$z = c\sinh\phi$

หมายเหตุ 1. รูปทั่วไปของไฮเพอร์โบลอยด์ชนิดผิวเดียว คือ

$$Ax^2 + By^2 - Cz^2 = D$$

หรือ $$Ax^2 - By^2 + Cz^2 = D$$

หรือ $$-Ax^2 + By^2 + Cz^2 = D$$

เมื่อ $A > 0, B > 0, C > 0$ และ $D > 0$

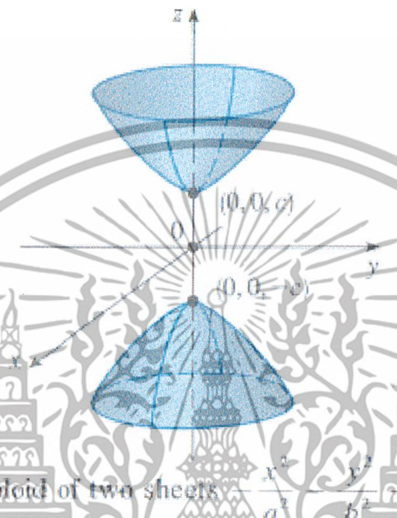
2. พื้นผิว $\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1$ มีแกน z เป็นแกนกลางของพื้นผิวซึ่งจะได้ว่าพื้นผิวไม่ตัดแกน z ในทำนองเดียวกัน $\frac{x^2}{a^2} - \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$ พื้นผิวไม่ตัดแกน y และสมการ $-\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$ แสดงว่าพื้นผิวไม่ตัดแกน x

2.1.3 ทฤษฎีทางคณิตศาสตร์สำหรับไฮเปอร์โบลอยด์ชนิดสองผิว (Hyperboloid of two sheets)

เขียนในรูปสมการ

$$-\frac{x^2}{a^2}-\frac{y^2}{b^2}+\frac{z^2}{c^2}=1 \text{ โดยที่ } a>0, b>0, c>0$$

พิจารณารูปไฮเปอร์โบลอยด์ชนิดสองผิว ดังรูป



รูปที่ 2.1-3 ไฮเปอร์โบลอยด์ชนิดสองผิว (elliptic hyperboloid of two sheets)

พิจารณารายละเอียดของไฮเปอร์โบลอยด์ชนิดสองผิว จากการพิจารณาสมการ

$$-\frac{x^2}{a^2}-\frac{y^2}{b^2}+\frac{z^2}{c^2}=1 \quad \text{จะได้ว่า}$$

จุดตัดแกน ไม่มีจุดตัดแกน x

ไม่มีจุดตัดแกน y

ตัดแกน z ที่จุด $(0,0,c)$ และ $(0,0,-c)$

สมมาตร พื้นผิวมีสมมาตรกับจุดกำเนิด

พื้นผิวมีสมมาตรกับแกนพิกัดทั้งสาม

พื้นผิวมีสมมาตรกับระนาบพิกัดทั้งสาม

รอยตัด ตัดด้วยระนาบ xy ไม่มีจุดตัด (ไม่มีรอยตัด)

ตัดด้วยระนาบ yz ได้รูปไฮเปอร์โบลา $-\frac{y^2}{b^2}+\frac{z^2}{c^2}=1, x=0$

ตัดด้วยระนาบ xz ได้รูปไฮเปอร์โบลา $-\frac{x^2}{a^2}+\frac{z^2}{c^2}=1, y=0$

ตัดด้วยระนาบที่ขนานกับระนาบพิกัด

ตัดด้วยระนาบที่ขนานกับระนาบ xy (คือ $z = k$) จะได้รูปวงรี

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}=\frac{k^2}{c^2}-1, z=k \text{ เมื่อ } |k|>c$$

สำหรับ $|k|<c$ ไม่มีรอยตัด และ $|k|=c$ จะได้จุดตัดคือ $(0,0,c)$ และ $(0,0,-c)$ ซึ่งขนาดของวงรีจะโตขึ้นเมื่อ $|k|$ มากขึ้น

ตัดด้วยระนาบที่ขนานกับระนาบ yz (คือ $x = k$) จะได้รูปไฮเพอร์โบลา

$$-\frac{y^2}{b^2}+\frac{z^2}{c^2}=1+\frac{k^2}{a^2}, x=k \text{ สำหรับทุกๆ ค่า } k$$

ตัดด้วยระนาบที่ขนานกับระนาบ xz (คือ $y = k$) จะได้รูปไฮเพอร์โบลา

$$-\frac{x^2}{a^2}+\frac{z^2}{c^2}=1+\frac{k^2}{b^2}, y=k \text{ สำหรับทุกๆ ค่า } k$$

สำหรับขอบเขตของพื้นผิวไฮเพอร์โบลอยด์ชนิดสองผิวได้จากการพิจารณารอยตัดของพื้นผิวดังนี้

ขอบเขตของ x บนพื้นผิว คือ $(-\infty,\infty)$

ขอบเขตของ y บนพื้นผิว คือ $(-\infty,\infty)$

ขอบเขตของ z บนพื้นผิว คือ $(-\infty,c]\cup[c,\infty)$

จัดให้อยู่ในรูปของรัศมี มุมละติจูด และ มุมลองติจูด ได้ดังนี้

$x = \pm a\cosh\phi$ โดยที่ $0 \le \theta \le 2\pi$

$y = b\sin\theta\sinh\phi$ โดยที่ $-\pi \le \phi \le \pi$

$z = c\cos\theta\sinh\phi$

<u>**หมายเหตุ**</u> 1. รูปทั่วไปของไฮเพอร์โบลอยด์ชนิดสองผิว คือ

$$Ax^2 - By^2 - Cz^2 = D$$

หรือ $-Ax^2 + By^2 - Cz^2 = D$

หรือ $-Ax^2 - By^2 + Cz^2 = D$ เมื่อ $A>0, B>0, C>0$ และ $D>0$

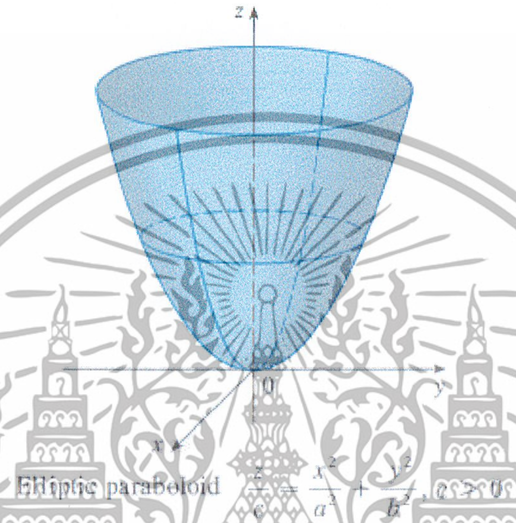
2. ข้อแตกต่างระหว่างไฮเพอร์โบลอยด์ชนิดผิวเดียว และไฮเพอร์โบลอยด์ชนิดสองผิว คือ ไฮเพอร์โบลอยด์ชนิดผิวเดียวมีสัมประสิทธิที่เป็นลบอยู่เพียงพจน์เดียว แต่ไฮเพอร์โบลอยด์ชนิดสองผิวมีสัมประสิทธิที่เป็นลบอยู่สองพจน์

2.1.4 ทฤษฎีทางคณิตศาสตร์สำหรับพาลาโบลอยด์เชิงวงรี (Elliptic Paraboloid)

เขียนได้ในรูปสมการ

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}=z \qquad \text{โดยที่} \quad a>0\,,b>0$$

พิจารณารูปพาลาโบลอยด์เชิงวงรี ดังรูป



รูปที่ 2.1-4 พาราโบลอยด์เชิงวงรี (elliptic paraboloid)

พิจารณารายละเอียดของรูปพาราโบลอยด์เชิงวงรี จะได้ว่า

จุดตัดแกน พื้นผิวผ่านจุดกำเนิด

สมมาตร ไม่มีสมมาตรกับจุดกำเนิด

มีสมมาตรกับแกน z แต่ไม่มีสมมาตรกับแกน x,y

มีสมมาตรกับระนาบ yz และ xz แต่ไม่สมมาตรกับระนาบ xy

รอยตัด ตัดด้วยระนาบ xy ได้จุดกำเนิด

ตัดด้วยระนาบ yz ได้รูปพาราโบลา $y^2=b^2cz; x=0$

ตัดด้วยระนาบ xz ได้รูปพาราโบลา $x^2=a^2cz; y=0$

รอยตัดด้วยระนาบที่ขนานกับระนาบพิกัด

ตัดด้วยระนาบที่ขนานกับระนาบ xy (คือ $z=k$) จะได้รูปวงรี $\frac{x^2}{a^2}+\frac{y^2}{b^2}=ck, z=k$ เมื่อ $k>0$ แต่ถ้า $k=0$ จะได้จุดกำเนิด และถ้า $k<0$ ไม่มีรอยตัด

ตัดด้วยระนาบที่ขนานกับระนาบ yz (คือ $x = k$) จะได้รูปพาราโบลา $y^2 = b^2cz - \dfrac{b^2k^2}{a^2}, x = k$ สำหรับทุกๆ ค่า k

ตัดด้วยระนาบที่ขนานกับระนาบ xz (คือ $y = k$) จะได้รูปพาราโบลา $x^2 = a^2cz - \dfrac{a^2k^2}{b^2}, y = k$ สำหรับทุกๆ ค่า k

สำหรับขอบเขตของพื้นผิว

ขอบเขตของ x บนพื้นผิว คือ $(-\infty, \infty)$

ขอบเขตของ y บนพื้นผิว คือ $(-\infty, \infty)$

ขอบเขตของ z บนพื้นผิว คือ $[0, \infty)$

จัดให้อยู่ในรูปรัศมี มุมละติจูด และ มุม ลองติจูด ได้ดังนี้

$x = a\phi\cos\theta$ โดยที่ $0 \le \theta \le 2\pi$

$y = b\phi\sin\theta$ โดยที่ $0 \le \phi \le \phi_{\max}$

$z = \phi^2$

<u>**หมายเหตุ**</u> 1. รูปทั่วไปของพาราโบลอยด์เชิงวงรี คือ

$Ax^2 + By^2 = Cz; A > 0, B > 0, C \neq 0$

หรือ $By^2 + Cz^2 = Ax; B > 0, C > 0, A \neq 0$

หรือ $Ax^2 + Cz^2 = By; A > 0, C > 0, B \neq 0$

2. รูปพื้นผิว $\dfrac{x^2}{a^2} + \dfrac{y^2}{b^2} = cz$ มีลักษณะคล้ายถ้วยหงายเมื่อ $c > 0$ แต่ถ้า $c < 0$ รูปมีลักษณะเหมือนเดิมแต่รูปคว่ำ

2.1.5 ทฤษฎีทางคณิตศาสตร์สำหรับไฮเพอร์โบลิก พาราโบลอยด์ (Hyperbolic Paraboloid)

เขียนได้ในรูปสมการ

$$\frac{x^2}{a^2}-\frac{y^2}{b^2}=z \qquad \text{โดยที่} \quad a>0\,,b>0$$

พิจารณารูปไฮเพอร์โบลิก พาราโบลอยด์ ดังรูป

รูปที่ 2.1-5 ไฮเพอร์โบลิกพาราโบลอยด์ (hyperbolic paraboloid)

พิจารณารายละเอียดของไฮเพอร์โบลิกพาราโบลอยด์ จะได้ว่า

จุดตัดแกน พื้นผิวผ่านจุดกำเนิด

สมมาตร ไม่มีสมมาตรกับจุดกำเนิด

มีสมมาตรกับแกน z แต่ไม่มีสมมาตรกับแกน x,y

มีสมมาตรกับระนาบ yz และ xz แต่ไม่สมมาตรกับระนาบ xy

รอยตัด ตัดด้วยระนาบ xy ได้เส้นตรงสองเส้นคือ $y=\pm\frac{b}{a}x$

ตัดด้วยระนาบ yz ได้รูปพาราโบลา $y^2=-b^2cz;x=0$

ตัดด้วยระนาบ xz ได้รูปพาราโบลา $x^2=a^2cz;y=0$

รอยตัดด้วยระนาบที่ขนานกับระนาบพิกัด

ตัดด้วยระนาบที่ขนานกับระนาบ xy (คือ $z=k$) จะได้รูปไฮเพอร์โบลา $\frac{x^2}{a^2}-\frac{y^2}{b^2}=ck,z=k$ สำหรับทุกๆ ค่า k

ตัดด้วยระนาบที่ขนานกับระนาบ yz (คือ $x = k$) จะได้รูปพาราโบลา $y^2 = -b^2c(z - \frac{k^2}{a^2}); x = k$ สำหรับทุกๆ ค่า k

ตัดด้วยระนาบที่ขนานกับระนาบ xz (คือ $y = k$) จะได้รูปพาราโบลา $x^2 = a^2c(z + \frac{k^2}{b^2c}); y = k$ สำหรับทุกๆ ค่า k

สำหรับขอบเขตของรูปพื้นผิว

ขอบเขตของ x บนพื้นผิว คือ $(-\infty, \infty)$

ขอบเขตของ y บนพื้นผิว คือ $(-\infty, \infty)$

ขอบเขตของ z บนพื้นผิว คือ $(-\infty, \infty)$

จัดให้อยู่ในรูปรัศมี มุมละติจูด และ มุม ลองติจูด ได้ดังนี้

$x = a\phi\cosh\theta$ โดยที่ $-\pi \le \theta \le \pi$

$y = b\phi\sinh\theta$ โดยที่ $\phi_{\min} \le \phi \le \phi_{\max}$

$z = \phi^2$

ϕ *คือ มุมระหว่างแกน Z กับระนาบ XY*

หมายเหตุ 1. รูปทั่วไปของไฮเพอร์โบลิกพาราโบลอยด์ คือ

$$Ax^2 - By^2 = Cz; A > 0, B > 0, C \neq 0$$

หรือ $By^2 - Cz^2 = Ax; B > 0, C > 0, A \neq 0$

หรือ $Ax^2 - Cz^2 = By; A > 0, C > 0, B \neq 0$

2. รูปพื้นผิวไฮเพอร์โบลิกพาราโบลอยด์เรียกอีกชื่อหนึ่งว่ารูปพื้นผิวอานม้า

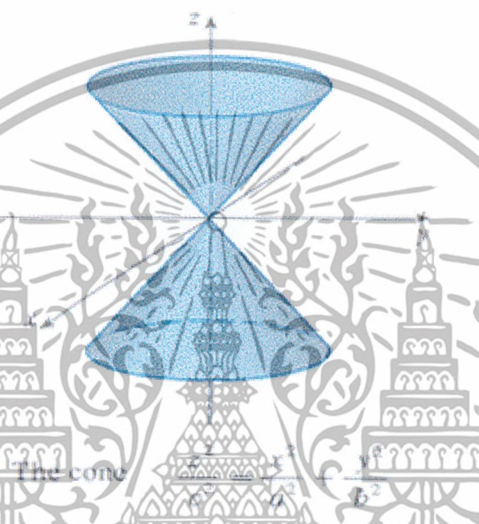
3. รูปพื้นผิว $\frac{x^2}{a^2} - \frac{y^2}{b^2} = cz$ ถ้า $c < 0$ รูปพื้นผิวยังคงเหมือนเดิมแต่รูปพื้นผิวมีลักษณะเป็นอานม้าหงาย

2.1.6 ทฤษฎีทางคณิตศาสตร์สำหรับกรวยเชิงวงรี (Elliptic cone)

เขียนในรูปสมการ

$$z^2 = \frac{x^2}{a^2} + \frac{y^2}{b^2} \qquad \text{โดยที่ } a > 0 , b>0$$

พิจารณารูปกรวยเชิงวงรี ดังรูป



รูปที่ 2-6 กรวยเชิงวงรี (elliptic cone)

พิจารณารายละเอียดของกรวยเชิงวงรี จะได้ว่า

จุดตัดแกน พื้นผิวผ่านจุดกำเนิด

สมมาตร พื้นผิวมีสมมาตรกับจุดกำเนิด

พื้นผิวมีสมมาตรกับแกนพิกัดทั้งสาม

พื้นผิวมีสมมาตรกับระนาบพิกัดทั้งสาม

รอยตัด ตัดด้วยระนาบ xy ได้จุดกำเนิด

ตัดด้วยระนาบ yz ได้เส้นตรงสองเส้น คือ $y = \pm\frac{b}{a}x , x = 0$

ตัดด้วยระนาบ xz ได้เส้นตรงสองเส้น คือ $x = \pm\frac{a}{c}x , y = 0$

รอยตัดด้วยระนาบที่ขนานกับระนาบพิกัด

ตัดด้วยระนาบที่ขนานกับระนาบ xy คือ ตัดด้วยระนาบ $z = k$ จะได้รูปวงรี

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = \frac{k^2}{c^2}, z = k \quad \text{สำหรับทุก ๆ ค่า } k \text{ ซึ่ง} |k| \neq 0$$

ตัดด้วยระนาบที่ขนานกับระนาบ yz คือ ตัดด้วยระนาบ $x = k$ จะได้รูปไฮเพอร์โบลา

$$\frac{z^2}{c^2} - \frac{y^2}{b^2} = \frac{k^2}{a^2}, x = k \quad \text{สำหรับทุก ๆ ค่า } k \text{ ซึ่ง} |k| \neq 0$$

ตัดด้วยระนาบที่ขนานกับระนาบ xz คือ ตัดด้วยระนาบ $y = k$ จะได้รูปไฮเพอร์โบลา

$$\frac{z^2}{c^2} - \frac{x^2}{a^2} = \frac{k^2}{b^2}, y = k \quad \text{สำหรับทุก ๆ ค่า } k \text{ ซึ่ง} |k| \neq 0$$

จัดให้อยู่ในรูปของรัศมี มุมละติจูด และ มุมลองติจูดได้ดังนี้

$x = a$ โดยที่ $0 \leq \theta \leq 2\pi$

$y = b\phi \sin\theta$ โดยที่ $\phi_{\min} \leq \phi \leq \phi_{\max}$

$z = c\phi$

หมายเหตุ 1. รูปทั่วไปของพื้นผิวกรวยเชิงวงรี คือ

$$Ax^2 + By^2 - Cz^2 = 0$$

หรือ $$Ax^2 - By^2 + Cz^2 = 0$$

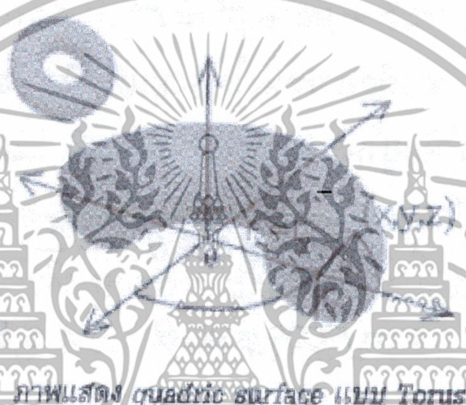
หรือ $$-Ax^2 + By^2 + Cz^2 = D$$ สำหรับทุก ๆ ค่า $A > 0, B > 0, C > 0$

2.1.7 ทฤษฎีทางคณิตศาสตร์สำหรับวงแหวน (TORUS)

เขียนได้ในรูปสมการ

$$\left[R-\sqrt{\left(\frac{x}{a}\right)^2+\left(\frac{y}{b}\right)^2}\,\right]^2+\left[\frac{z}{c}\right]^2=1$$

พิจารณาวงแหวน ดังรูป



รูปที่ 2-7 วงแหวน(Torus)

เป็นวัตถุรูปทรงโดนัทสามารถสร้างได้โดยการหมุนวงกลมรอบแกน ใด ๆ จัดให้อยู่ในรูปรัศมี มุมละติจูด และ มุม ลองติจูด ได้ดังนี้

$$x=a\left(R+\cos\phi\right)\cos\theta \qquad \text{โดยที่} \quad -\pi\le\phi\le\pi$$
$$y=b\left(R+\cos\phi\right)\sin\theta \qquad \text{โดยที่} \quad -\pi\le\theta\le\pi$$
$$z=c\sin\phi$$

<u>หมายเหตุ</u> **เมื่อเทียบกับรูปด้านบน**

$x=\mathrm{X}\,,\, y=\mathrm{Y}\,,\, z=\mathrm{Z}$

$a=\mathrm{r}_x\,,\, b=\mathrm{r}_y\,,\, c=\mathrm{r}_z$

2.2 การแปลงภาพเรขาคณิตในระบบภาพ 2 มิติ

โดยพื้นฐานแล้วภาพของระบบคอมพิวเตอร์กราฟฟิก ถูกสร้างขึ้นด้วยส่วนของเส้นตรงหรือเวกเตอร์มาประกอบกัน ซึ่งเราสามารถกำหนดเส้นตรงหรือเวคเตอร์เหล่านี้ได้ด้วยจุดปลายทั้งสองของมัน การที่จะเปลี่ยนแปลงลักษณะภาพที่วาดจำเป็นจะต้องใช้การคำนวณทางคณิตศาสตร์กระทำกับจุดต่างๆ เหล่านี้ วิธีหนึ่งที่จะช่วยให้เข้าใจได้ง่ายก็คือการใช้เมตริกซ์เข้าช่วย แต่สำหรับผู้ที่ไม่ถนัดการใช้เมตริกซ์ก็อาจจะจำสูตรไปใช้งานได้เช่นกัน

เราใช้เมตริกซ์ 2 มิติ เพื่อช่วยทำการแปลง โดยแทนจุดต่าง ๆ ด้วยเมตริกซ์ขนาด 1 x 2 เช่น จุด (0.5 ,1) แทนด้วยเมตริกซ์ [0.5 1] เมื่อต้องการทำการแปลงใดๆก็เพียงแต่หาเมตริกซ์ที่เหมาะสมมาคูณกับเมตริกซ์ของจุด ผลลัพธ์ที่ได้คือเมตริกซ์ ใหม่ ซึ่งก็คือจุดหรือตำแหน่งใหม่ของจุดปลายของเส้นตรงต่าง ๆ ทำให้เส้นหรือเวกเตอร์เหล่านี้มีลักษณะที่เปลี่ยนไป เมตริกซ์ที่นำมาคูณเพื่อเปลี่ยนตำแหน่งของจุดนี้เราเรียกว่าเมตริกซ์การแปลง (Tansfomation matrix)

2.2.1 การแปลงแบบสเกล (Scaling)

สมมติว่าเรามีจุด $P_1 = [\, x_1 \quad y_1 \,]$ ซึ่งเป็นเมตริกซ์ขนาด 1 x 2 ถ้าเราคูณเมตริกซ์นี้ด้วยเมตริกซ์ T ขนาด 2 x 2 เราจะได้เมตริกซ์ขนาด 1 x 2 เมตริกซ์ใหม่ (P_2) กลับมาโดยที่

$$P_2 = [\, x_2 \quad y_2 \,] = P_1T$$

การคูณเมตริกซ์นี้จะเป็นการคูณทางซ้าย และ T เป็นเมตริกซ์การแปลงที่จะแปลงจุด P_1 ไปเป็นจุด P_2 ถ้าเรานำเมตริกซ์การแปลง T นี้คูณเข้ากับจุดทุก ๆ จุด แล้วอะไรจะเกิดขึ้น คำตอบก็คือเราจะได้ภาพที่มีลักษณะเปลี่ยนแปลงไป แต่จะเปลี่ยนแปลงอย่างไรนั้นขึ้นอยู่กับค่าต่าง ๆ ในเมตริกซ์ T เช่น

ถ้าเราคูณด้วย เมตริกซ์เอกลักษณ์ (Identity matrix)

$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$P_2 = P_1T = \begin{bmatrix} X_1 & Y_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} X_1 & Y_1 \end{bmatrix} = P_1$$

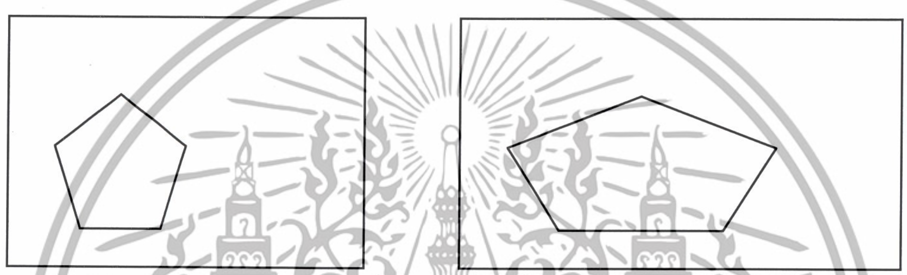
จุด P_1 กับ P_2 จะเป็นจุดเดียวกัน คือไม่มีการเปลี่ยนแปลงใดๆ แต่ถ้าเราเลือกเมตริกซ์ T_1 เป็น

$$T_1 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

$$P_2 = P_1T = \begin{bmatrix} X_1 & Y_1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2X_1 & Y_1 \end{bmatrix}$$

ผลลัพธ์ที่ได้คือโคออร์ดิเนต X ของทุก ๆจุด จะมีค่ามากขึ้นเป็น 2 เท่าของค่าเดิม เส้นตรงในแนวนอนก็จะมีความยาวเป็น 2 เท่าของภาพเดิม ภาพใหม่ที่เปลี่ยนไปจะมีความสูงเท่าเดิม แต่ขยายออกทางแนวนอนออกจากจุดกำเนิด (0 , 0) เป็น 2 เท่าจากของเดิม ดังตัวอย่างใน รูป



รูปที่ 2.2-1 การขยายทางแนวนอน

ในทำนองเดียวกัน ถ้าเมตริกซ์การแปลง T_2 มีค่าเป็น

$$T_2 = \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix}$$

โคออร์ดิเนต X ของทุก ๆ จุด ก็จะลดลงเป็นครึ่งหนึ่งของค่าเดิม ภาพใหม่ที่ได้จะมีความสูงเท่าเดิม แต่จะถูกบีบให้แคบทางแนวนอนเป็นครึ่งหนึ่ง

คราวนี้เราลองขยายภาพออกทางแนวนอนเป็น 2 เท่า แล้วบีบภาพใหม่นี้ให้แคบลงเป็นครึ่งหนึ่ง แน่นอนเราต้องได้ภาพที่มีขนาดเท่าเดิมกลับมา

$$P_1 = (P_1 T_1)T_2 = P_1(T_1 T_2)$$

เราสามารถตรวจสอบคำตอบของเราได้โดย นำเมตริกซ์ T_1 และ T_2 มาคูณกัน

$$T_1 T_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} (2 * 0.5) + (0 * 0) & (2 * 0) + (0 * 1) \\ (0 * 0.5) + (1 * 0) & (0 * 0) + (1 * 1) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

ผลลัพธ์ที่ได้คือเมตริกซ์เอกลักษณ์ นั่นคือ เราจะได้ภาพเดิมก่อนการแปลง เราสามารถที่จะขยายภาพให้สูงขึ้น หรือกดภาพให้เตี้ยลงในแนวตั้งได้เช่นกัน โดยการหาเมตริกซ์การแปลงที่จะทำให้ค่าโคออร์ดิเนต Y เปลี่ยนไปโดยที่ค่า X ไม่เปลี่ยน ตัวอย่างเช่น เมตริกซ์การแปลง T_3

$$T_3 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

$$P_2 = P_1T_3 = \begin{bmatrix} X_1 & Y_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

$$= \begin{bmatrix} X_1 & Y_1 \end{bmatrix}$$

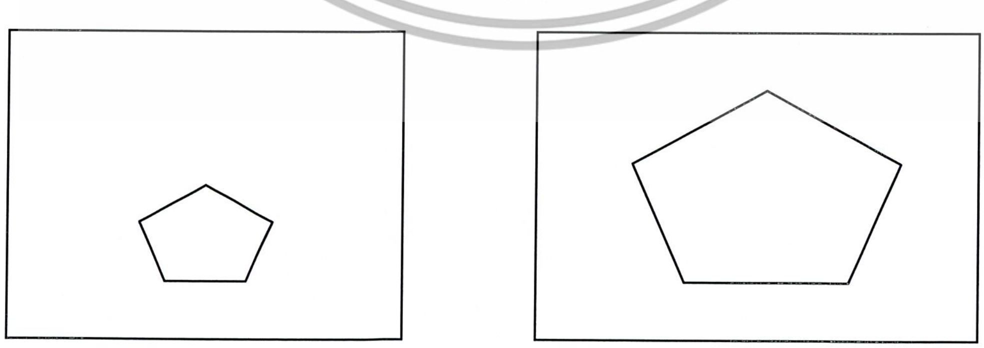
จะได้ภาพที่มีความสูงเพิ่มขึ้นจากเดิม 2 เท่า ดังในภาพ

ก่อนแปลง หลังแปลง

รูปที่ 2.2-2 การขยายในแนวตั้ง

ถ้าเรานำเมตริกซ์ T_1 และ T_3 มาคูณกับ P_1 เราก็จะได้ภาพที่มีการขยายออกทั้งแนวนอนและแนวตั้งเป็น 2 เท่าของภาพเดิม หรือกล่าวได้ว่า จะได้ภาพที่มีความโตเป็น 2 เท่าดังในรูป

$$P_2 = P_1T_1T_3 = P_1(T_1T_3)$$



ก่อนแปลง หลังแปลง

รูปที่ 2.2-3 การขยายทางแนวนอน

เรานำเมตริกซ์ T_1 และ T_3 มาคูณกันเพื่อให้ได้เมตริกซ์ใหม่ T_4 เมตริกซ์เดียว

$$T_4 = T_1 T_3 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

$$T_4 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

เราอาจเขียนเมตริกซ์การแปลงแบบนี้ในรูปทั่วๆไป คือ

$$S = \begin{bmatrix} S_X & 0 \\ 0 & S_Y \end{bmatrix}$$

S_X เรียกว่า สเกลแฟคเตอร์ (Scale Factor) สำหรับโคออร์ดิเนต X ซึ่งจะมีผลต่อขนาดของภาพในแนวนอนหรือแนวแกน X

S_Y เรียกว่า สเกลแฟคเตอร์ สำหรับโคออร์ดิเนต Y ซึ่งจะมีผลต่อขนาดของภาพ ในแนวตั้งหรือแนวแกน Y

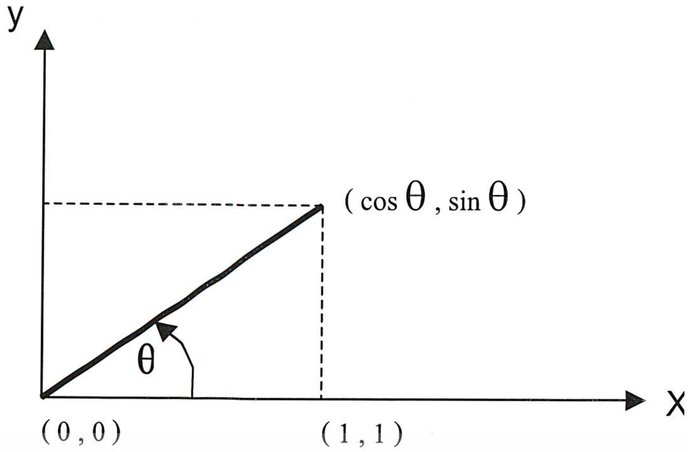
เมตริกซ์ S เป็นเมตริกซ์การแปลงซึ่งจะมีผลต่อขนาดและสัดส่วนของภาพ เราเรียกว่า เมตริกซ์การแปลงแบบสเกล (Scaling Transfomation matrix) การแปลงโดยการคูณด้วยเมตริกซ์ S เราเรียกว่าเป็น การแปลงแบบสเกล (Scaling Transfomation)

สังเกตว่าการสเกลภาพจะทำให้ทุกๆจุดมีการเปลี่ยนแปลง ยกเว้นเพียงจุดเดียวคือจุดกำเนิด (0 , 0)ดังนั้นนอกจากขนาดของภาพจะเปลี่ยนไปแล้ว ตำแหน่งของจะต่างๆก็เปลี่ยนไปด้วย ถ้า S_X มีค่ามากกว่า 1 ก็จะทำให้ภาพเลื่อนไปทางขวา (สำหรับภาพที่อยู่ทางขวาของแกน Y) และมีความกว้างมากขึ้นด้วย ถ้า S_X มีค่าน้อยกว่า 1 ภาพก็จะเลื่อนไปทางซ้ายและมีความแคบลง ในทำนองเดียวกัน ถ้า S_Y มีค่ามากกว่า 1 ก็จะทำให้ภาพขยายห่างออกจากแกน X และมีความสูงเพิ่มขึ้น ถ้า S_Y มีค่าน้อยกว่า 1 ก็จะทำให้ภาพหดเข้าหาแกน X และเตี้ยลง

2.2.2 การแปลงแบบหมุน (Rotation Transformation)

เป็นวิธีการเปลี่ยนภาพโดยการหมุนจุด (หรือภาพ) ในทิศทางทวนเข็มหรือตามเข็มนาฬิกา โดยมีจุดศูนย์กลางของการหมุนอยู่ที่จุดกำเนิด สิ่งที่เราต้องการทราบก็คือเมตริกซ์ซึ่งมาคูณจุดที่เราต้องการหมุนเพื่อไปอยู่ตำแหน่งใหม่ เราเรียกเมตริกซ์นี้ว่า เมตริกซ์การแปลงแบบหมุน (Rotation Transformation Matrix)

สมมติว่าเราทำการหมุนจุด (1,0) ไปในทิศทางบวก (ทวนเข็มนาฬิกา) เป็นมุม θ องศา ตำแหน่งใหม่จะเป็นจุด (cos(θ) , sin (θ)) ดังภาพ



รูปที่ 2.2-4 การหมุนจุด (1,0) ไปในทิศทางทวนเข็มนาฬิกาเป็นมุม θ องศา

ถ้าเมตริกซ์การแปลงแบบหมุนคือ

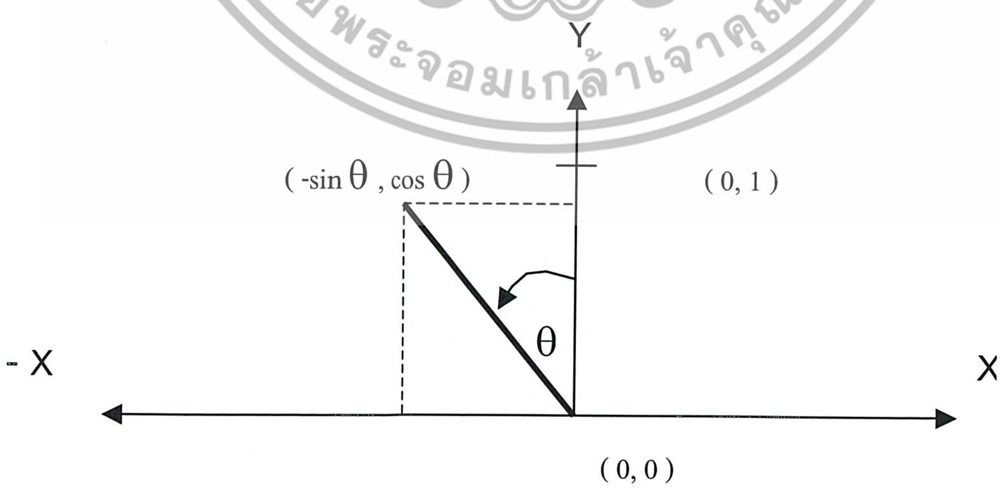
$$S = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \text{..................(1)}$$

ดังนั้นจะได้ว่า

$$[\cos(\theta) \quad \sin(\theta)] = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$= [a \quad b] \quad \text{..................(2)}$$

ถ้าเราหมุนจุด (0, 1) ในทิศทางบวกเป็นมุม θ เช่นกัน นั่นคือใช้เมตริกซ์การแปลงตัวเดิม ตำแหน่งใหม่จะเป็น (-sin (θ) , cos(θ)) ดังรูป



รูปที่ 2.2-5 การหมุนจุด (0,1) ไปในทิศทางทวนเข็มนาฬิกาเป็นมุม θ องศา

ดังนั้น

$$[\ -\sin(\theta) \quad \cos(\theta)\ ] = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$= \begin{bmatrix} c & d \end{bmatrix}$$

จากสมการ (1) และ (2) จะได้ว่า

$$a = \cos(\theta)$$

$$b = \sin(\theta)$$

$$c = \cos(\theta)$$

$$d = \sin(\theta)$$

ดังนั้นเมตริกซ์การแปลงแบบหมุน (ทวนเข็ม) คือ

$$R = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \qquad \ldots\ldots\ldots\ldots\ldots(3)$$

วิธีการหาเมตริกซ์การแปลงแบบหมุนนี้ดูเหมือนค่อนข้างง่ายความจริงแล้วเมตริกซ์การแปลงแบบหมุนนี้หาได้จากการพิสูจน์ทางตรีโกณมิติที่ยุ่งยากจนกระทั่งได้ผลลัพธ์เช่นเดียวกับสมการ (3) ดังนั้นจึงไม่ขอกล่าวไว้ ณ ที่นี้

ในสมการ (3) เป็นเมตริกซ์การแปลงที่ใช้กับการหมุนในทิศทางทวนเข็มนาฬิกา สำหรับการหมุนในทิศทางตามเข็มนาฬิกาคือ

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

ตัวอย่างเช่น ต้องการหมุนจุด $P_1(3, 2)$ ในทิศทางทวนเข็มเป็นมุม 30 องศา เมตริกซ์การแปลงจะเป็น

$$\begin{bmatrix} \cos(30) & \sin(30) \\ -\sin(30) & \cos(30) \end{bmatrix} = \begin{bmatrix} 0.866 & 0.5 \\ -0.5 & 0.866 \end{bmatrix}$$

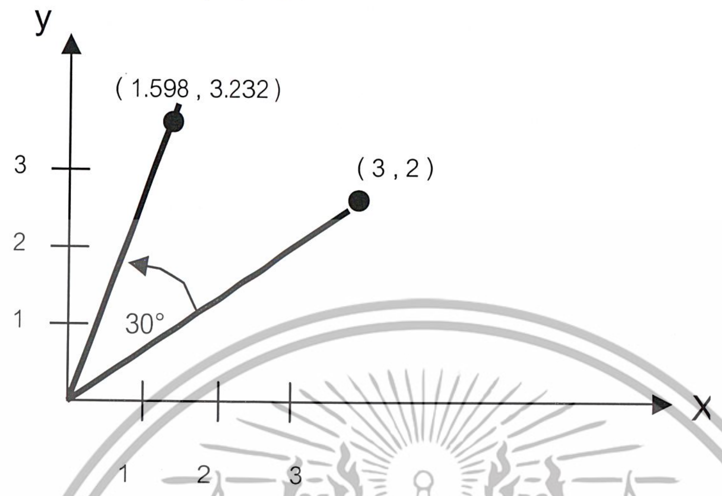
และจุดใหม่หลังการหมุน P_2 คือ

$$P_2 = [\ 3 \quad 2\ ] \begin{bmatrix} 0.866 & 0.5 \\ -0.5 & 0.866 \end{bmatrix}$$

$$= [\ 1.598 \quad 3.232\ ]$$

2.2.3 การแปลงแบบย้ายและโฮโมจีเนียสโคออร์ดิเนต

การย้าย (Translation) เป็นการเลื่อนตำแหน่งของภาพทั้งภาพไปยังตำแหน่งอื่น ๆ เป็นระยะทางเท่ากันทั้งหมด โดยขนาดของภาพไม่เปลี่ยนแปลงและไม่ทำให้ภาพเอียงไปจากแนวเดิม การย้ายนี้ทำได้โดยการบวกจุดทุก ๆ จุดของภาพด้วยระยะทางที่ต้องการให้ภาพเลื่อนไป ดังนั้น

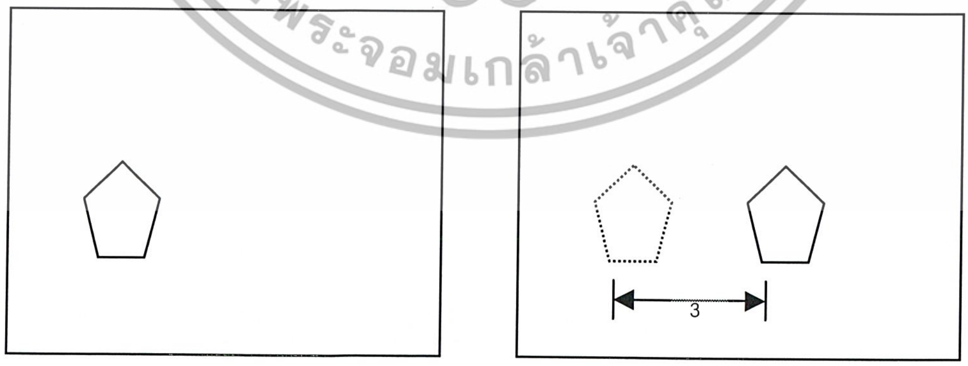


รูปที่ 2.2-6 แสดงการหมุนจุด (3 , 2) ในทิศทางทวนเข็มเป็นมุม 30 องศา

$$x' = x + t_x$$

และ
$$y' = y + t_y$$

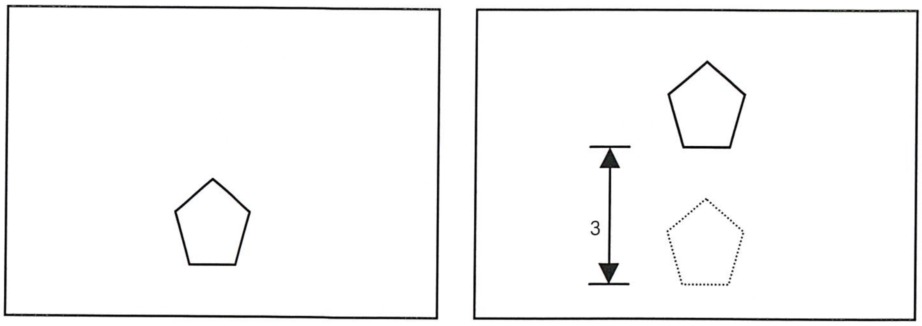
นั่นคือการย้ายตำแหน่งของภาพไปในแนวแกน x เป็นระยะ t_x และแนวแกน y เป็นระยะทาง t_y หน่วย โดยที่ t_x และ t_y เป็นระยะทางที่ต้องการให้ภาพเลื่อนไปในแนวแกน x และแกน y ตามลำดับ ตัวอย่างเช่น ต้องการย้ายภาพให้เลื่อนไปอยู่ทางขวา 3หน่วยเราก็บวก 3เข้ากับค่า x ของทุก ๆ จุดของภาพ ดังในภาพ



ก่อนย้าย หลังย้าย

รูปที่ 2.2-7 การย้ายภาพให้เลื่อนไปอยู่ทางขวา 3หน่วย

หรือถ้าต้องการย้ายภาพขึ้นข้างบน 3 หน่วย เราก็บวก 3 เข้ากับค่า Y ของทุก ๆ จุด ดังรูป



รูปที่ 2.2-8 การย้ายภาพขึ้นข้างบน 3 หน่วย

ในทางตรงกันข้าม ถ้าต้องการเลื่อนภาพไปทางซ้ายหรือลงข้างล่าง ก็นำจำนวนลบมาบวกเข้าไป เพื่อให้ค่า X หรือค่า Y ลดลง การย้ายภาพสามารถย้ายขึ้น-ลง , ไปซ้าย-ขวาได้พร้อมกัน โดยเปลี่ยนทั้งค่า X และ Y ของทุก ๆ จุดพร้อมกัน

จากที่กล่าวมาจะเห็นว่า การแปลงแบบย้ายนั้นไม่มีเมตริกซ์ที่มาใช้คูณเพื่อให้ได้ตำแหน่งใหม่ของจุดต่าง ๆ เราเพียงแค่นำค่าที่เหมาะสมมาบวกเข้ากับค่า X และ Y ของจุดต่างๆ เท่านั้น ลักษณะเช่นนี้ทำให้เรารวมการแปลงแบบย้ายเข้ากับการแปลงแบบสเกลหรือการแปลงแบบหมุนไม่ได้ เราไม่สามารถสร้างเมตริกซ์เพียงเมตริกซ์เดียว มาคูณกับเมตริกซ์ของจุดต่าง ๆ เพื่อให้เกิดการแปลงหลายๆแบบรวมกัน เกิดความไม่สะดวกในการทำงาน แต่ปัญหานี้เราสามารถแก้ไขได้โดยใช้โฮโมจีเนียสโคออร์ดิเนต (Homogeneous Coordinate) เข้าช่วย โฮโมจีเนียสโคออร์ดิเนตจะใช้เมตริกซ์ขนาด 3x3 แทนเมตริกซ์ขนาด 2x2 ที่ได้กล่าวมาแล้ว จุดหรือตำแหน่งใหม่จะขึ้นอยู่กับจำนวน 3 จำนวนแทนที่จะเป็น 2 จำนวน (โคออร์ดิเนต Xและ Y ของจุด)โดยเพิ่มโคออร์ดิเนตใหม่ W ขึ้นมา จำนวน 3 จำนวนนี้ได้แก่ ผลคูณของของโคออร์ดิเนต X กับ W ผลคูณของโคออร์ดิเนต Y กับ W และโคออร์ดิเนต W ดังนั้นโคออร์ดิเนตของจุดต่าง (x ,y) จะถูกแทนด้วยจำนวน 3 จำนวน คือ (XW , YW , W) ถ้าเรามีโคออร์ดิเนตใหม่นี้ และต้องการทราบตำแหน่ง (x ,y) เดิมของมัน ก็ทำได้โดยนำเอาโคออร์ดิเนตที่ 3 หารสองโคออร์ดิเนตแรกเท่านั้น โคออร์ดิเนต W นี้จะถูกใช้งานอย่างแท้จริงในการแปลงในระบบ 3 มิติ ส่วนกรณี 2 มิติเราจะให้ค่า W เป็น 1 เสมอ

ในโฮโมจีเนียสโคออร์ดิเนต เมตริกซ์การแปลงแบบสเกลจะถูกเปลี่ยนจากเดิม

$$\begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

เป็น

$$S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{.....................(4)}$$

เราทดสอบโดยนำเมตริกซ์การแปลงของสมการ (4) มาคูณกับจุด $P_1(x , y)$ซึ่งต้องแทนด้วยเมตริกซ์ [xw yw w]

$$P_2 = P_1 S$$

$$= [\,xw \quad yw \quad w\,] \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$P_2 = [\,S_x xw \quad S_y yw \quad w\,]$$

เมื่อนำโคออร์ดิเนต W มาหารสองโคออร์ดิเนตแรก จะได้จุด P_2เป็น $(S_x xw , S_y yw)$ ซึ่งตรงกับการแปลงแบบเสกลที่กล่าวมาแล้ว

เมตริกซ์การแปลงแบบหมุน ในทิศทางทวนเข็มนาฬิกาจากเดิม

$$\begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

จะเปลี่ยนเป็น

$$R = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{........(5)}$$

นำเมตริกซ์ในสมการที่ 5 คูณกับจุด P_1 (xw , yw , w) เราจะได้

$$P_2 = [\,xw \quad yw \quad w\,] \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= [\,xw*\cos(\theta) - yw*\sin(\theta) \quad xw*\sin(\theta) + yw*\cos(\theta) \quad w\,]$$

ซึ่งจะได้จุด P_2 ที่ได้จากการหมุนจุด P_1คือ (xw*cos(θ) – yw*sin(θ) , xw*sin(θ) + yw*cos(θ) , w) ตรงกับการแปลงแบบหมุนที่กล่าวมาแล้ว

สำหรับการแปลงแบบย้าย เมื่อต้องการเลื่อนภาพหรือจุดไปทางแนวนอน t_x และไปทางแนวตั้ง t_y จะได้เมตริกซ์การแปลงแบบย้ายคือ

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} \quad \text{................(6)}$$

เราสามารถพิสูจน์ได้ว่าเมตริกซ์ในสมการที่6ใช้งานได้โดยนำไปคูณกับจุด$P_1(xw,yw,w)$

$$P_2 = P_1T$$

$$= [\,xw \quad yw \quad w\,] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

$$P_2 = [\,xw + t_x w \quad yw + t_y w \quad w\,]$$

เราจะได้จุด P_2 ($xw + t_x w$, $yw + t_y w$, w) ดังนั้นจุด P_2 ก็คือ $(x + t_x , y + t_y)$

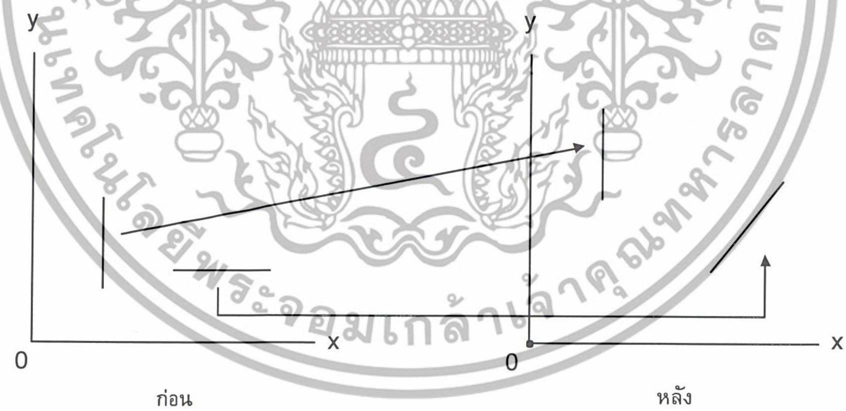
2.2.4 การบิดภาพ (shearing)

การบิดภาพจะทำให้บางส่วนของภาพหรือภาพทั้งหมดเกิดการบิดเบือนขึ้น เราจะพิจารณาเพียง 2 แบบคือ การบิดภาพทางแกน x และการบิดภาพทางแกน y

การบิดภาพทางแกน y จะทำให้เกิดการย้ายจุด (x, y) ไปยังจุด (x', y') โดยที่ $x' = x$

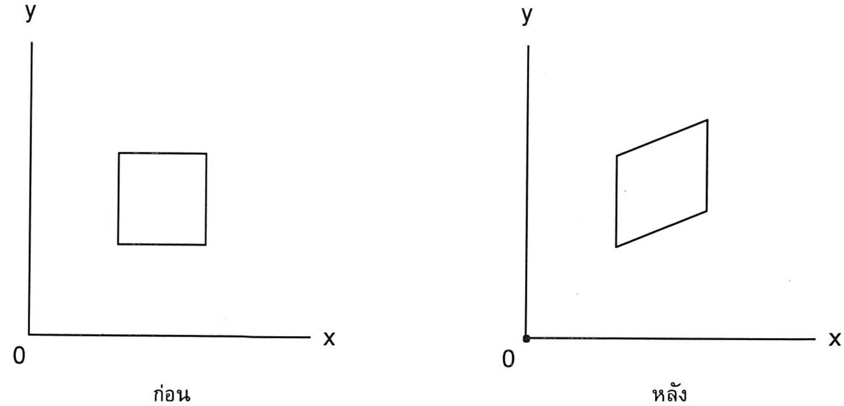
$$y' = Shy * x + y, Shy \neq 0$$

การบิดภาพทางแกน y จะทำให้จุดต่างๆ ในแกน y เลื่อนขึ้นหรือเลื่อนลงขึ้นอยู่กับเครื่องหมายของแฟกเตอร์ Shy เส้นตรงในแนวนอนจะถูกเปลี่ยนให้เป็นเส้นตรงในแนวเฉียงด้วยความลาดชันเท่ากับ Shy พิจารณาจากรูป



รูปที่ 2.2-9 การบิดภาพทางแกน y ทำให้เส้นในแนวนอนเปลี่ยนไปเป็นเส้นในแนวเฉียง

ในรูปต่อไปนี้ จะเป็นการแสดงการบิดภาพสี่เหลี่ยมทางแกน y ซึ่งทำให้ภาพเดิมซึ่งเป็นภาพสี่เหลี่ยมผืนผ้าเปลี่ยนแปลงไปเป็นภาพสี่เหลี่ยมด้านขนาน



รูปที่ 2.2-10 การบิดภาพสี่เหลี่ยมทางแกน y

สำหรับการบิดภาพทางแกน x จะให้ผลตรงข้ามกับการบิดภาพทางแกน y กล่าวคือ จุด (x, y) ของภาพจะถูกแปลงเป็นจุด (x', y') โดยที่

$$x' = x + Shx * y, Shx \neq 0$$
$$y' = y$$

ในกรณีเส้นในแนวนอนก็จะถูกย้ายไปทางซ้ายหรือทางขวา ขึ้นอยู่กับเครื่องหมายของแฟกเตอร์ Shx ส่วนเส้นตรงในแนวดิ่งก็จะถูกบิดไปเป็นเส้นตรงในแนวเฉียงด้วยความลาดชัน Shx

รูปที่ 2.2-11 การบิดภาพสี่เหลี่ยมทางแกน x

จากภาพจะเป็นการบิดภาพสี่เหลี่ยมทางแกน x ภาพจะเปลี่ยนจากสี่เหลี่ยมผืนผ้าให้กลายเป็นสี่เหลี่ยมด้านขนาน

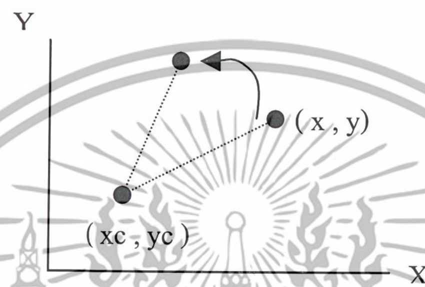
2.2.5 การหมุนรอบจุดใด ๆ

การแปลงแบบหมุนที่กล่าวมาเป็นการหมุนรอบจุดกำเนิด (0,0) เท่านั้น แต่ในหัวข้อนี้เราจะหาเมตริกซ์การแปลงแบบหมุนที่ใช้หมุนรอบจุดใด ๆ (x_c, y_c)

วิธีที่จะทำการหมุนแบบนี้มีขั้นตอนอยู่ 3 ขั้นตามลำดับ คือ

1. ทำการย้ายภาพเพื่อให้จุดศูนย์กลางของการหมุน (x_c, y_c) ไปอยู่ที่จุดกำเนิด
2. ทำการหมุนรอบจุดกำเนิด
3. ย้ายภาพเพื่อให้จุดศูนย์กลางของการหมุนกลับไปอยู่ในตำแหน่งเดิม

แสดงขั้นตอนการหมุนรอบจุด (x_c, y_c) ได้ดังรูป

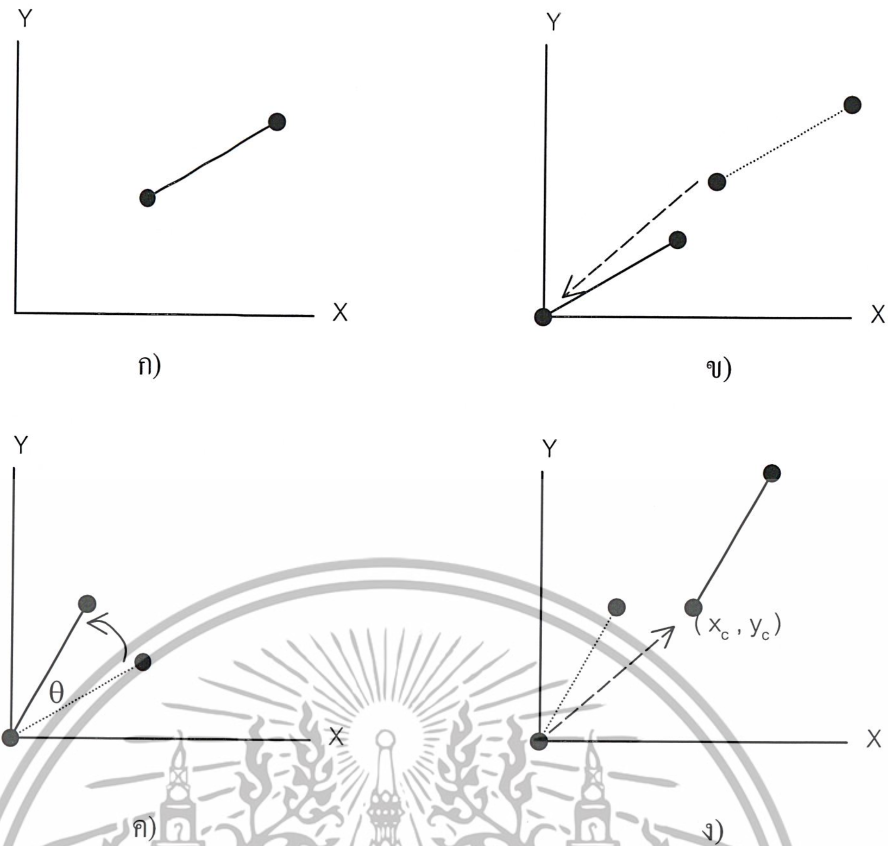


รูปที่ 2.2-12 การหมุนรอบจุด (x_c, y_c)

ขั้นตอนทั้ง 3 ขั้นนี้จะต้องทำทีละขั้นตอน เรียงลำดับให้ถูกต้อง เพราะในการทำแต่ละขั้นตอน ต้องใช้เมตริกซ์การแปลงมาคูณเข้าไป และการคูณเมตริกซ์ไม่มีคุณสมบัติการสลับที่ ดังนั้นถ้าคูณเมตริกซ์การแปลงไม่เรียงลำดับ จะได้ผลลัพธ์ที่ไม่ถูกต้อง

เมตริกซ์ที่จะย้ายจุด (x_c, y_c) ไปยังจุดกำเนิดก็คือ

$$T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_c & -y_c & 1 \end{bmatrix}$$



รูปที่ 2.2-13 ขั้นตอนการหมุนรอบจุดใด ๆ (x_c, y_c)

เมตริกซ์สำหรับการหมุนคือ

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

และเมตริกซ์ที่จะย้ายกลับไปยังจุดเดิมคือ

$$T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_c & y_c & 1 \end{bmatrix}$$

นำเมตริกซ์ทั้ง 3 มาคูณเข้ากับจุดที่เราจะหมุน $P(x, y)$ ได้จุดกำเนิดใหม่เป็น $P_1(x_1, y_1)$

$$P_1 = [(PT_1)R]T_2$$

$$= [P(T_1 R)]T_2$$

$$= P(T_1 R T_2)$$

ดังนั้นเมตริกซ์การแปลงแบบหมุนรอบจุด (x_c, y_c) คือ

$$T_1RT_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_c & -y_c & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_c & y_c & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ -x_c\cos(\theta) + y_c\sin(\theta) + x_c & -x_c\sin(\theta) - y_c\sin(\theta) + y_c & 1 \end{bmatrix}$$

สมการ คือ เมตริกซ์การแปลงที่หมุนภาพในทิศทางทวนเข็มนาฬิกาเป็นมุม a องศารอบจุด $P(x_c, y_c)$ เพื่อความสะดวกเราอาจเขียนเป็นสูตรสำหรับการหมุนจุด $P(x, y)$ รอบจุดเป็นมุม a องศาได้เป็นจุดใหม่ $P'(x', y')$ ดังนี้

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_c & -y_c & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ x_c & y_c & 1 \end{bmatrix}$$

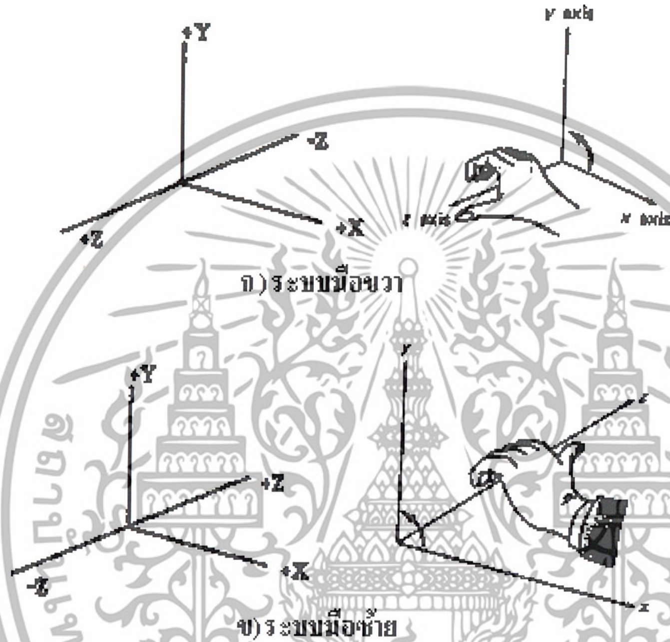
$$x' = (x - x_c)\cos(\theta) - (y - y_c)\sin(\theta) + xc$$

$$y' = (x - x_c)\sin(\theta) + (y - y_c)\cos(\theta) + yc$$

2.3 ระบบภาพ 3 มิติ

2.3.1 ระบบโคออร์ดิเนต

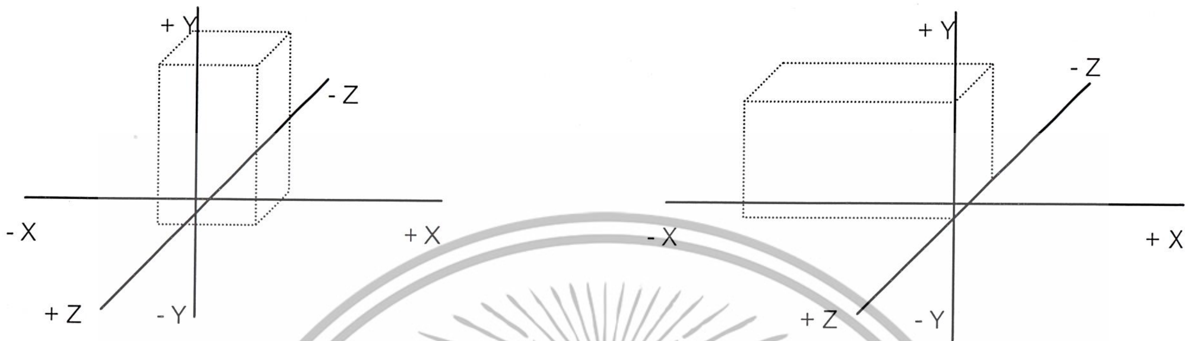
ในระบบ 2 มิติ มีแกนเพียง 2แกนเท่านั้นคือ แกน X และแกน Y แต่ในระบบ 3 มิติต้องเพิ่มแกน Z เข้าไปอีกหนึ่งแกน การกำหนดทิศทางของแกน Z มา 2 แบบ คือ แบบระบบมือขวา และแบบระบบมือซ้าย ดังแสดงดังรูป



รูปที่ 2.3-1 ระบบโคออร์ดิเนต 3 มิติ

ในระบบมือขวาเราใช้นิ้วโป้งของมือขวาชี้ไปในแนวแกน +Z งอนิ้วที่เหลือทั้ง 4 นิ้ว ทศทางการหมุนของนิ้วทั้ง 4 จะหมุนจากแกน +X เข้าหาแกน +Y ส่วนในระบบมือซ้าย เราใช้นิ้วโป้งของมือซ้ายชี้ไปในแนวแกน +Z นิ้วทั้ง 4 จะหมุนจากแกน +X เข้าหาแกน + Y โดยทั่วไปแล้วเรามักจะใช้โคออร์ดิเนตแบบระบบมือขวา เช่นในงานทางคณิตศาสตร์หรือการใช้งานในระบบภูมิศาสตร์ แต่สำหรับระบบคอมพิวเตอร์กราฟิก มักนิยมใช้ระบบมือซ้ายทั้งนี้เพราะเราจะถือว่าระนาบของจอภาพเป็นระนาบ XY(เป็นแกนXและY) และให้ระยะความลึกเข้าไปหลังจอภาพมีค่าเป็นบวก (ค่าโคออร์ดิเนต Z เป็นบวก)

การกำหนดจุดในระบบ 3 มิติ ต้องใช้จำนวน 3 จำนวน เพื่อเป็นการระบุว่าจุดนั้นห่างจากจุดกำเนิด (0,0,0) ไปตามแนวแกน X Y และ Z เป็นค่าเท่าใด เช่นจุด (1,2,0.5) คือจุดที่ห่างจากจุดกำเนิดไปในแนวแกน +X 1หน่วย แกน +Y 2 หน่วยและแกน +Z 0.5 หน่วย จุด (-2,1,-1) คือจุดที่ห่างจากจุดกำเนิดไปในแนวแกน - X 2 หน่วย แกน +Y 1 หน่วย และแกน -Z 1 หน่วย ดังแสดงในรูป



ก) จุด(1, 2, 0.5) ข) จุด(-2 , 1 , -1)

รูปที่ 2.3-2 การกำหนดจุดในระบบ 3 มิติ

2.3.2 เส้นตรงและเวคเตอร์

เส้นตรงในระบบ 3 มิติ ยังคงกำหนดด้วยจุดปลายทาง 2 จุด เช่นเดียวกับในระบบ 2 มิติ เราใช้จุด 2 จุดนี้กำหนดความยาว และทิศของเส้นตรง สมการของเส้นตรงต้องใช้ 2สมการประกอบกันซึ่งมีรูปแบบของสมการดังนี้

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

$$\frac{z - z_1}{x - x_1} = \frac{z_2 - z_1}{x_2 - x_1} \qquad .....(1)$$

โดยที่จุด (x_1 , y_1 , z_1) และ (x_2 , y_2 , z_2) คือจุดที่อยู่บนเส้นตรงเดียวกัน ตัวอย่างเช่น มีจุด (1 , 2 , 1) และจุด (0 , 1 , 2) อยู่บนเส้นตรงเดียวก้าน สมการของเส้นตรงนี้หาได้โดยแทนค่า x_1 y_1 z_1 x_2 , y_2 และ z_2 ลงในสมการ (1)

$$\frac{y - 2}{x - 1} = \frac{1 - y_1}{0 - 1}$$

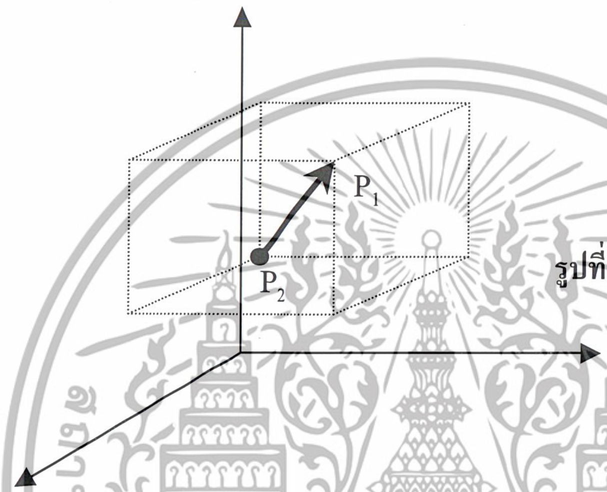
$$\frac{z - 2}{x - 1} = \frac{2 - 1}{0 - 1}$$

ดังนั้นสมการของเส้นตรงนี้คือ

$$y = x + 1$$

$$z = -x + 3$$

สำหรับเวคเตอร์ในระบบ 3 มิติ ก็คล้ายกับส่วนของเส้นตรงคือ สามารถกำหนดได้ด้วยจุด 2 จุดในระบบ แต่เวคเตอร์ต่างกับส่วนของเส้นตรงคือ เวคเตอร์มีทิศทาง โดยจุดหนึ่งเป็นจุดเริ่มต้นของเวคเตอร์ และอีกหนึ่งจุดคือจุดปลาย ตัวอย่างเช่น ถ้าเรามีเวคเตอร์ที่มีจุดเริ่มต้นคือ $P_1(x_1, y_1, z_1)$ และมีจุดปลายคือ $P_2(x_2,y_2,z_2)$ เวคเตอร์นี้จะมีทิศทางแสดงดังภาพ



รูปที่ 2.3-3 เวคเตอร์ในระบบ 3 มิติ

ขนาดของเวคเตอร์นี้หาได้จากสมการ

$$\textbf{ขนาดของเวคเตอร์} = [(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2]^{1/2}$$

โดยทั่วไปเรามักสนใจเฉพาะทิศทางของเวคเตอร์เท่านั้นซึ่งเราสามารถแทนแนวทิศทางของเวคเตอร์ได้ด้วยเมตริกซ์ [x y z] ค่าของ x y และ z เป็นอัตราส่วนซึ่งแสดงว่าเวคเตอร์นั้นหันเหไปในแนวแกน X Y และ Z มากน้อยเท่าใดตามลำดับ สำหรับในระบบคอมพิวเตอร์กราฟฟิกเรามักใช้เมตริกซ์ [x y z] แทนเวคเตอร์นั้นไปเลย ถ้าจุดตั้งต้นของเวคเตอร์คือ $P_1(x_1, y_1, z_1)$ และจุดปลายของเวคเตอร์คือ $P_2(x_2, y_2, z_2)$ เราสามารถหาเวคเตอร์นี้ได้คือ [x y z] โดยที่

$$x = x_2 - x_1$$

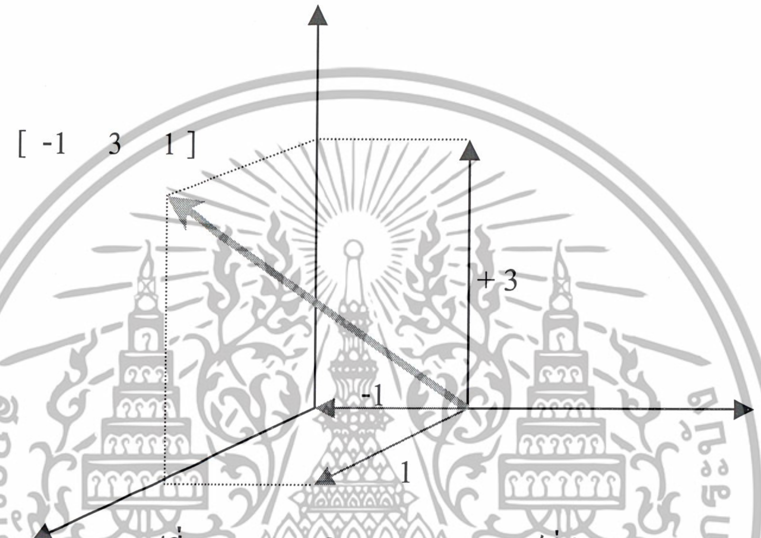
$$y = y_2 - y_1$$

$$z = z_2 - z_1$$

ยกตัวอย่าง เช่น จุด P_1 คือ(5 , 0 , 2) และ P_2 คือ (4 , 3 , 3) ดังนั้น

$$x = 4 - 5 = -1$$
$$y = 3 - 0 = 3$$
$$z = 3 - 2 = 1$$

เวคเตอร์นี้จะหันเหไปในแนวแกน X Y และ Z ด้วยอัตราส่วน –1 : 3 : 1 ตามลำดับ ซึ่งสามารถแสดงทิศทางของเวคเตอร์นี้ได้ดังในรูป



รูปที่ 2.3-4 แสดงทิศทางของเวคเตอร์ที่มีอัตราส่วน –1 : 3 : 1

จะสังเกตได้ว่าเวคเตอร์ในรูปเป็นการแสดงเฉพาะแนวทิศทางของเวคเตอร์เท่านั้น ซึ่งเราไม่สนใจขนาดและจุดตั้งต้นของเวคเตอร์ ดังนั้นเราอาจใช้เวคเตอร์ที่มีขนาดเล็กกว่า หรือยาวกว่าก็ได้ ขอเพียงให้เป็นเวคเตอร์ที่มีทิศทาง เดียวกันก็พอ นั่นคือค่าของ x y และ z ในเมตริกซ์ [x y z] สามารถเปลี่ยนแปลงได้ด้วยอัตราส่วนที่เท่ากัน เช่นเวคเตอร์ [-1 3 1] จะมีทิศทางเดียวกับเวคเตอร์ [-2 6 2] หรือ [-1/3 1 1/3] แค่เพื่อให้สอดคล้องกัน เรามักจะใช้เวคเตอร์ที่มีขนาดหนึ่งหน่วย (เรียกว่า เวคเตอร์หนึ่งหน่วย) เพื่อกำหนดทิศทาง เวคเตอร์หนึ่งหน่วย ที่มีทิศทางตรงกับเวคเตอร์ที่เกิดจากจุด $P_1(x_1, y_1, z_1)$ 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 คือขนาดของเวคเตอร์ $= [(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2]^{1/2}$

ตัวอย่างเช่นถ้าต้องการหาเวคเตอร์หนึ่งหน่วย ที่มีทิศทางเดียวกันกับเวคเตอร์[-1 3 1] ต้องทราบขนาดของเวคเตอร์ก่อน

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

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

$$= [-0.30 \quad 0.90 \quad 0.30]$$

2.3.3 ระนาบ

ในระบบภาพ 3 มิติ นอกเหนือจากจุดและเส้นตางแล้ว เรายังต้องยุ่งเกี่ยวกับระนาบอีกด้วย ระนาบเปรียบได้กับแผ่นกระดานราบเรียบที่ไม่มีความหนา แต่มีขนาดใหญ่ไม่จำกัดตั้งอยู่ในระบบโคออร์ดิเนตเช่นเดียวกันกับเส้นตรง ระนาบมีสมการในการกำหนดหรือบ่งชี้ระนาบนั้น ๆ สมการระนาบมีรูปแบบดังนี้

$$Ax + By + Cz + D = 0$$

เมื่อ A B และ C คือค่าคงที่ และ (x, y, z) คือจุดใด ๆ ที่อยู่บนระนาบ

ถ้าเราทราบ 3 จุดที่อยู่บนระนาบเดียวกันคือ $P_1(x_1, y_1, z_1)$ $P_2(x_2, y_2, z_2)$ และ $P_3(x_3, y_3, z_3)$ เราสามารถหาสมการของระนาบได้โดยแทนค่าจุดต่าง ๆ ลงบนสมการข้างต้นได้ดังนี้

$$Ax_1 + By_1 + Cz_1 + D = 0$$
$$Ax_2 + By_2 + Cz_2 + D = 0$$
$$Ax_3 + By_3 + Cz_3 + D = 0$$

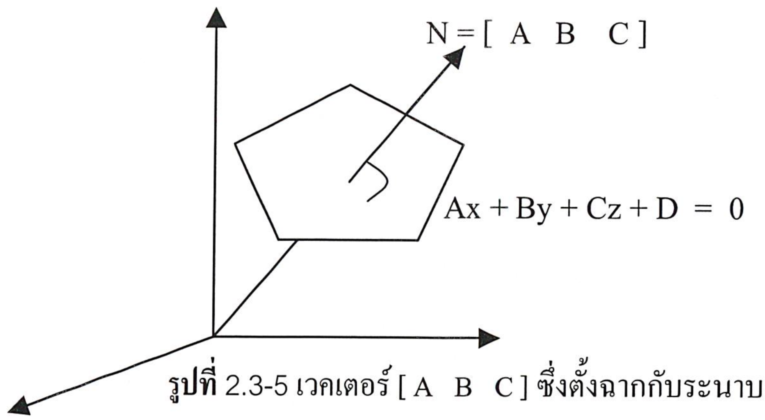
จากสมการทั้ง 3ข้างต้นนี้ เราสามารถแก้สมการเพื่อหาค่าของ A B และ C ได้ คือ

$$A = y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2)$$
$$B = z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2)$$
$$C = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)$$
$$D = -x_1(y_2z_3 - y_3z_2) - x_2(y_3z_1 - y_1z_3) - x_3(y_1z_2 - y_2z_1)$$

การจัดเรียงตัวของระนาบในระบบโคออร์ดิเนต สามารถกำหนดได้ด้วยเวคเตอร์หนึ่งเวคเตอร์ซึ่งเป็นเวคเตอร์มีทิศทางตั้งฉากกับพื้นผิวของระนาบเราเรียกเวคเตอร์นี้ว่าเวคเตอร์ตั้งฉาก (Normal Vector) ถ้าสมการของระนาบคือ

$$Ax + By + Cz + D = 0$$

เวคเตอร์ตั้งฉากของระนาบนี้ คือ [A B C] แสดงได้ดังรูป



รูปที่ 2.3-5 เวคเตอร์ [A B C] ซึ่งตั้งฉากกับระนาบ

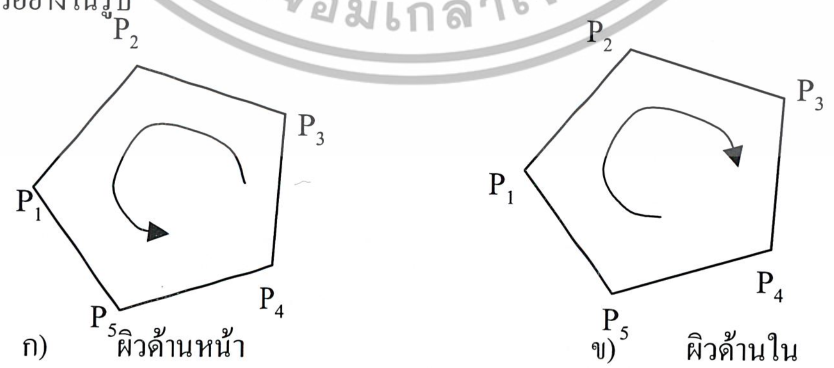
เวคเตอร์ตั้งฉากของระนาบเป็นสิ่งที่บ่งบอกทิศทางของระนาบได้ด้วย พื้นผิวของระนาบทุกระนาบมีอยู่ 2 ด้าน คือ ผิวด้านนอกและผิวด้านใน เวคเตอร์ตั้งฉากของระนาบจะพุ่งเข้าสู่ระนาบที่ผิวด้านในและพุ่งออกระนาบจากผิวด้านนอก พื้นผิวของระนาบที่เห็นในรูปต่างๆในระบบเป็นจุดที่อยู่ฝั่ง " ด้านใน " หรือเป็นจุดที่อยู่ฝั่ง " ด้านนอก" ของระนาบ จุด (x , y , z) ใด ๆ ที่อยู่ด้านนอกของระนาบ จะทำให้สมการ

$$Ax + By + Cz + D > 0 \quad \text{เป็นจริง}$$

และในทำนองเดียวกัน จุด (x , y , z) ใดๆ ที่อยู่ด้านในของระนาบ จะทำให้สมการ

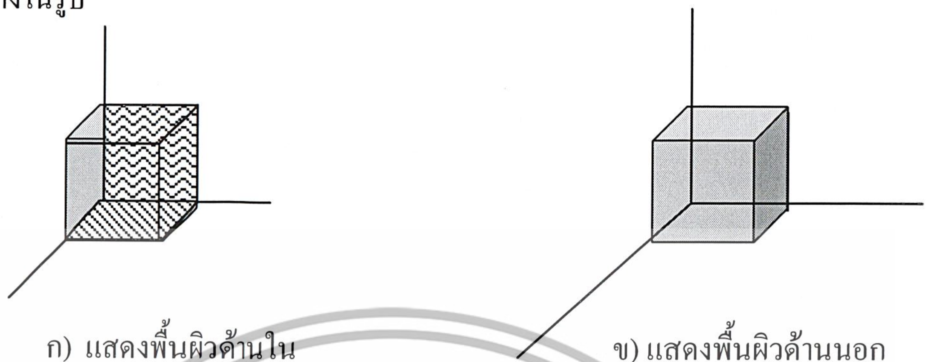
$$Ax + By + Cz + D < 0 \quad \text{เป็นจริง}$$

สำหรับรูปหลายเหลี่ยมในระบบ 3 มิติ ต้องวางตัวอยู่บนระนาบใดระนาบหนึ่ง รูปหลายเหลี่ยมจึงมีผิวด้านนอกและด้านในด้วยเช่นกัน การกำหนดรูปหลายเหลี่ยมทำได้โดยกำหนดจุดยอดทุกจุดของรูปหลายเหลายมเรียงตามลำดับและลากเส้นจากจุดแรกไปจนถึงจุดสุดท้าย ถ้าเรามองจากจุดนอกระนาบของรูปหลายเหลี่ยมแล้วเห็นจุดยอดต่าง ๆ ของรูปหลายเหลี่ยม เรียงกันไปในทิศทางทวนเข็มนาฬิกา เราจะถือว่า ผิวของรูปหลายเหลี่ยมที่เห็นนั้นคือผิวด้านนอก และในทางตรงกันข้าม ถ้าจุดยอดต่างๆ เรียงกันไปในทิศทางตามเข็มนาฬิกา เราจะมองเห็นผิวด้านในของรูปหลายเหลี่ยม ดังตัวอย่างในรูป



รูปที่ 2.3-6 การกำหนดผิวด้านนอก–ในของระนาบหรือรูปหลายเหลี่ยม

การที่เรากำหนดผิวด้านนอก–ในของระนาบหรือรูปหลายเหลี่ยม ก็เพื่อประโยชน์ในการสร้างภาพ โดยทั่วไปเรามักจะตัดภาพของส่วนที่เป็นพื้นผิวของด้านใน (Back Face Removal) ทั้งนี้เพื่อไม่ให้ภาพที่ปรากฏขึ้นมาไม่มีสับสนเกินไป และได้ภาพที่มีลักษณะเหมือนกับความเป็นจริง ดังเช่นตัวอย่างในรูป



ก) แสดงพื้นผิวด้านใน ข) แสดงพื้นผิวด้านนอก

ค) แสดงทั้งผิวด้านในและด้านนอก ง) แสดงเฉพาะผิวด้านนอก

รูปที่ 2.3-7 แสดงผิวด้านนอก–ในของกล่องสี่เหลี่ยม

เรามีรูปหลายเหลี่ยมหลายรูป ประกอบกันเป็นกล่องสี่เหลี่ยมกล่องหนึ่ง

ในรูป ก) พื้นผิวที่ถูกแรเงาคือพื้นผิวด้านใน

ในรูป ข) พื้นผิวถูกแรเงาคือพื้นผิวด้านนอก

ถ้าเราตัดภาพของรูปหลายเหลี่ยมที่หันผิวด้านในออก ภาพที่ได้จะเป็นภาพในรูป ง)

ซึ่งจะเป็นภาพที่เราเห็นในความเป็นจริง

ส่วนในรูป ค) เป็นการวาดพื้นผิวทั้งด้านนอกและด้านใน

2.3.4 การแปลงในระบบ 3 มิติ

การแปลงในระบบ 3 มิติ มีความคล้ายคลึงกับการแปลงในระบบ 2 มิติ เรายังคงใช้เมตริกซ์ช่วยในการทำการแปลง แต่ต้องเพิ่มขนาดของเมตริกซ์ไว้สำหรับโคออร์ดิเนต Z ดังนั้น จุดในระบบ 3 มิติ ต้องแทนด้วยเมตริกซ์ขนาด 1 X 3 จุด (x , y , z) จะถูกแทนด้วย [X Y Z]

2.3.4.1 การแปลงแบบสเกล (Scaling)

ในระบบภาพ 2 มิติ เรามีเมตริกซ์การแปลงแบบสเกล ขนาด 2x2 มีรูปแบบเป็น

$$S = \begin{bmatrix} S_X & 0 \\ 0 & S_Y \end{bmatrix}$$

และถ้าเป็นโฮโมจีเนียสโคออร์ดิเนตเมตริกซ์ มีรูปแบบเป็น

$$S = \begin{bmatrix} S_X & 0 & 0 \\ 0 & S_Y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

โดยที่ S_X และ S_Y คือ สเกลแฟคเตอร์สำหรับโคออร์ดิเนต X และ Y ตามลำดับ

ในระบบภาพ 3 มิติ เมตริกซ์การแปลงแบบสเกลจะกลายเป็นเมตริกซ์ขนาด 3x3 โดยมีรูปแบบเป็น

$$S = \begin{bmatrix} S_X & 0 & 0 \\ 0 & S_Y & 0 \\ 0 & 0 & S_Z \end{bmatrix}$$

และถ้าเป็นโฮโมจีเนียสโคออร์ดิเนตเมตริกซ์ จะมีรูปแบบเป็นเมตริกซ์ขนาด 4x4 คือ

$$S = \begin{bmatrix} S_X & 0 & 0 & 0 \\ 0 & S_Y & 0 & 0 \\ 0 & 0 & S_Z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

โดยที่ S_X S_Y และ S_Z คือ สเกลแฟคเตอร์สำหรับโคออร์ดิเนต X Y และ Z ตามลำดับ

การแปลงจุดต่างๆทำได้โดยนำเมตริกซ์การแปลงไปคูณกับจุดนั้นเหมือนกับการแปลงในระบบ2มิติ

$$P_2 = [x_2 \quad y_2 \quad z_2 \quad w]$$

$$= P_1 T$$

$$= [x_1 \quad y_1 \quad z_1 \quad w] \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_Z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= [S_x x_1 \quad S_y y_1 \quad S_Z z_1 \quad w]$$

ดังนั้น

$$x_2 = S_x x$$

$$y_2 = S_y y_1$$

$$z_2 = S_Z z_1$$

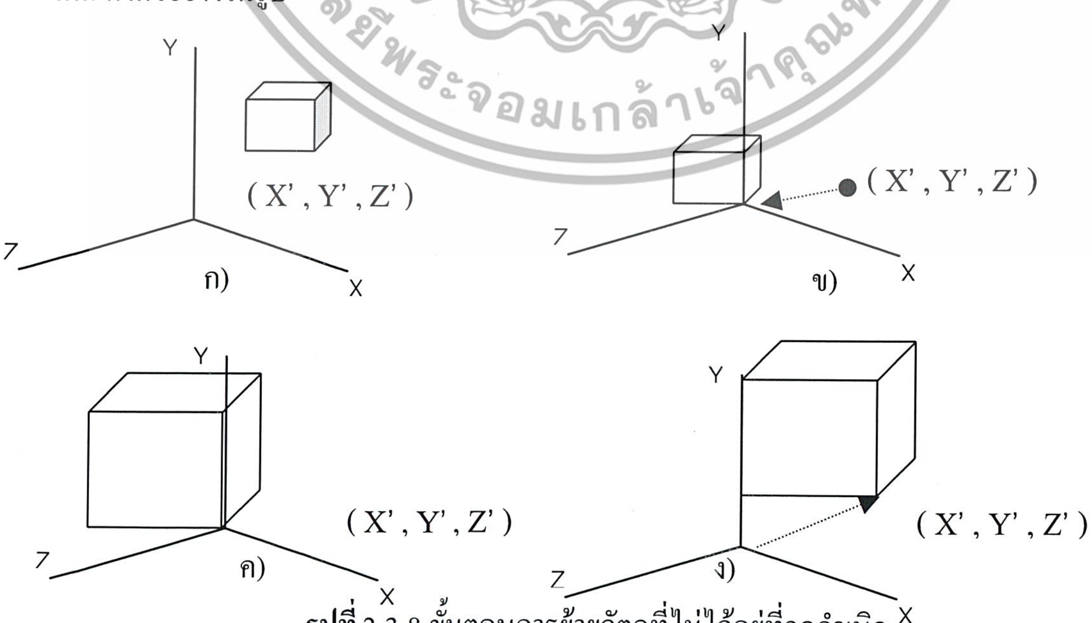
2.3.4.2 การแปลงแบบย้าย

เมตริกซ์การแปลงคือ

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_X & T_Y & T_Z & 1 \end{bmatrix}$$

เมื่อ T_X T_Y T_Z คือระยะที่ต้องเลื่อนไปตาม โคออร์ดิเนต X Y และ Z ตามลำดับ

ในทำนองเดียวกับระบบ2 มิติ การสเกลวัตถุใด ๆ ที่ไม่ได้อยู่ที่จุดกำเนิดต้องย้ายในตำแหน่งเดิม ดังตัวอย่างในรูป



รูปที่ 2.3-8 ขั้นตอนการย้ายวัตถุที่ไม่ได้อยู่ที่จุดกำเนิด

ดังนั้นเมตริกซ์การแปลง คือ

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -X & -Y & -Z & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_Z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ X & Y & Z & 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_X & 0 & 0 & 0 \\ 0 & S_Y & 0 & 0 \\ 0 & 0 & S_Z & 0 \\ (1 - S_X)x & (1 - S_Y)y & (1 - S_Z)z & 1 \end{bmatrix}$$

เมื่อ S_X S_Y S_Z คือ สเกลแฟคเตอร์สำหรับโคออร์ดิเนต X Y และ Z ตามลำดับ และ (x , y , z) คือจุดเริ่มต้นของวัตถุ

2.3.4.3 การแปลงแบบหมุน

ในระบบ 2 มิติ การหมุนรอบจุดกำเนิด (0 , 0) มีเมตริกซ์การแปลง คือ

$$R = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

แต่ในระบบ 3 มิติ การแปลงแบบหมุนมิได้เป็นการหมุนรอบจุดๆหนึ่ง แต่ เป็นการหมุนรอบแกนๆหนึ่ง ถ้าเราเปลี่ยนเมตริกซ์การแปลงแบบหมุนในสมการข้างต้น ให้ใช้ได้ในระบบ 3 มิติ เมตริกซ์ที่ได้จะเป็นเมตริกซ์การแปลงแบบหมุน Z เพราะการหมุนนี้ ค่าของโคออร์ดิเนต Z คงที่ไม่เปลี่ยนแปลง เราเขียนเมตริกซ์นี้ให้อยู่ในรูปของโฮโมจีเนียสโคออร์ดิเนตเมตริกซ์คือ

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

การหมุนเช่นนี้ เราอาจคิดว่าเป็นการหมุนวัตถุไปรอบ ๆ แกนที่หยุดนิ่ง หรืออาจคิดว่าแกนหมุนเคลื่อนที่ไปรอบๆ วัตถุที่อยู่นิ่งกับที่ก็ได้ ความแตกต่างของการแปลความหมายของการหมุนก็คือทิศทางของการหมุนเท่านั้น การหมุนวัตถุไปรอบ ๆ แกนที่อยู่กับที่ในทิศทางทวนเข็มนาฬิกา ก็คือการหมุนแกนไปรอบๆ วัตถุที่หยุดนิ่งทิศทางตามเข็มนาฬิกา

ถ้าหมุนจุด $P_1(x_1, y_1, z_1)$ รอบแกน Z เป็นมุม θ จะได้จุด $P_2(x_2, y_2, z_2)$ คือ

$$x_2 = x_1\cos(\theta) + y_1\sin(\theta)$$

$$y_2 = -x_1\sin(\theta) + y_1\cos(\theta)$$

$$z_2 = z_1$$

สำหรับการหมุนรอบแกน X และ Y ก็มีสมการคล้าย กับสมการ ข้างบน เมตริกซ์การแปลงแบบหมุนรอบแกน X คือ

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

ดังนั้นจุด $P_2(x_2, y_2, z_2)$ คือ

$$x_2 = x_1$$

$$y_2 = y_1\cos(\theta) + z_1\sin(\theta)$$

$$z_2 = -y_1\sin(\theta) + z_1\cos(\theta)$$

เมตริกซ์การแปลงแบบหมุนรอบแกน Y คือ

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

ดังนั้นจุด $P_2(x_2, y_2, z_2)$ คือ

$$x_2 = z_1\sin(\theta) - x_1\cos(\theta)$$

$$y_2 = y_1$$

$$z_2 = z_1\cos(\theta) + x_1\sin(\theta)$$

2.3.4.4 การหมุนรอบแกนใด ๆ

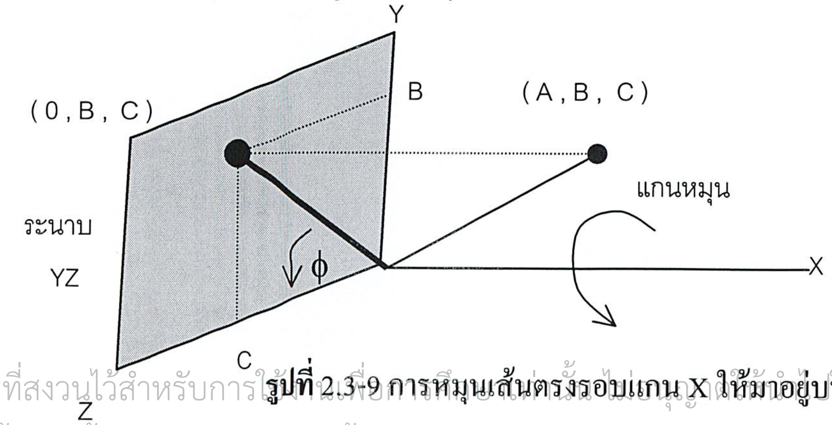
การแปลงแบบหมุนที่กล่าวไว้ในหัวข้อที่แล้ว เป็นการหมุนรอบแกนโคออร์ดิเนตทั้ง 3 เท่านั้น อย่างไรก็ตามเราสามารถหมุนวัตถุไปรอบแกนอื่นๆ ในระบบนอกเหนือจากแกนทั้ง 3 ได้ด้วยเช่นกัน โดยปกติแล้วเส้นตรงในระบบสามารถทำหน้าที่เป็นแกนหมุนได้ ซึ่งต่อไปเราจะหาเมตริกซ์การแปลงแบบหมุนรอบแกน (เส้นตรง) ใดๆ ในระบบ โดยเริ่มจากการย้ายจุดกำเนิดไปอยู่บนเส้นตรงหรือแกนหมุนนั้น จากนั้นเราจะหมุนไปรอบแกน X และแกน Y เพื่อจัดให้ แกนหมุนนั้นมีทิศทางเดียวกับแกน Z ดังนั้นการหมุนรอบแกน Z จึงเป็นการหมุนไปรอบแกนหมุนนี้ด้วย และท้ายสุดทำการแปลงกลับ เพื่อให้แกนหมุนกลับไปอยู่ในแนวเดิม และย้ายจุดกำเนิดกลับไปอยู่ในตำแหน่งเดิม การกำหนดเส้นตรงในระบบโคออร์ดิเนต สามารถกำหนดได้ด้วยจุดหนึ่งจุดบนเส้นตรงนั้นกับเวคเตอร์ที่มี

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ X_1 & Y_1 & Z_1 & 1 \end{bmatrix}$$

หลังจากการแปลงด้วยเมตริกซ์ T แล้ว จุด P_1 จะไปอยู่ที่จุดกำเนิดหลังจากการแปลงแบบหมุนเสร็จสิ้น เราต้องการอินเวอร์สเมตริกซ์ของ T เพื่อแปลงจุด P_1 กลับไปยังตำแหน่งเดิมก่อนการย้าย เมตริกซ์นี้คือ

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -X_1 & -Y_1 & -Z_1 & 1 \end{bmatrix}$$

ลำดับต่อไปคือการหมุนเส้นตรงรอบแกน X จนกระทั่งเส้นตรงอยู่บนระนาบ XY เพื่อให้เข้าใจการหามุมของการหมุน เราจะฉายเงาของเส้นตรงให้เกิดขึ้นบนระนาบ YZ สมมติว่าเรามีส่วนของเส้นตรงเส้นนี้ซึ่งอยู่ระหว่างจุด (0 , 0 ,0) และจุด (A , B , C) เราจะได้เงาของส่วนของเส้นตรงนี้ อยู่บนระนาบ YZ เป็นเส้นตรงที่อยู่ระหว่าง จุด (0 , 0 , 0) และ (0 , B , C)



รูปที่ 2.3-9 การหมุนเส้นตรงรอบแกน X ให้มาอยู่บนระนาบ XY

การหมุนเส้นตรงนี้ให้มาอยู่บนระนาบ XZ เงาของเส้นตรงจะเคลื่อนมาทับแกน Z พอดี นั่นคือมุมที่ใช้ในการหมุน คือ ϕ ดังแสดงในรูปข้างต้น ถ้า V คือความยาวของเงาบนระนาบ YZ ดังนั้น

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

และ

$$\sin(\phi) = B / V \qquad \text{............(b)}$$

$$\cos(\phi) = C / V \qquad \text{............(c)}$$

เมตริกซ์การแปลงแบบหมุนรอบแกน X คือ

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

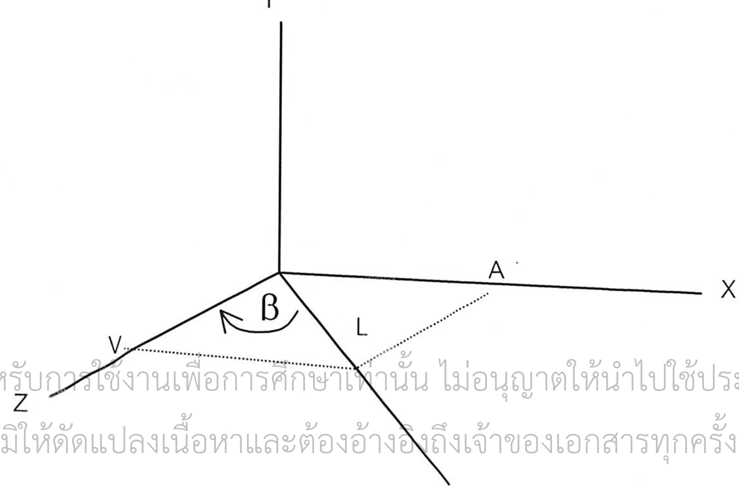
เมื่อแทนค่าในสมการ (b) และ (c) ลงในสมการของ R_X จะได้

$$R_X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C/V & B/V & 0 \\ 0 & -B/V & C/V & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

สำหรับการแปลงกลับของเมตริกซในสมการนี้คือการหมุนด้วยมุมเท่าเดิมแต่ในทิศทางตรงกันข้าม ทำให้ได้ อินเวอร์สเมตริกซ์คือ

$$R_X^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C/V & -B/V & 0 \\ 0 & B/V & C/V & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

หลังจากที่หมุนเส้นตรงไปบนระนาบ XZแล้ว เราจะหมุนเส้นตรงนี้รอบแกน Y จนกระทั่งเส้นตรงนี้ทับกับแกนเส้นพอดีดังแสดงในรูป



ซึ่งมุมที่ใช้ในการหมุนคือ β ถ้าความยาวของส่วนของเส้นตรงนี้คือ L ดังนั้น

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

$$= (A^2 + B^2 + C^2)^{1/2} \quad \text{............(d)}$$

และ $\sin(\beta) = A / L$ ....(e)

$\cos(\beta) = V / L$ ....(f)

เมตริกซ์การแปลงแบบหมุนรอบแกน Y คือ

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

$$= \begin{bmatrix} V/L & 0 & -A/L & 0 \\ 0 & 1 & 0 & 0 \\ A/L & 0 & V/L & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

และทำนองเดียวกันการแปลงกลับต้องหมุนรอบแกน Y ในทิศทางตรงกันข้ามเป็นมุม β ดังนั้นอินเวอร์สเมตริกซ์ คือ

$$R_Y^{-1} = \begin{bmatrix} V/L & 0 & -A/L & 0 \\ 0 & 1 & 0 & 0 \\ A/L & 0 & V/L & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ต่อไปเราจะหมุนวัตถุไปรอบแกน Z เป็นมุม θ โดยที่มุม θ นี้คือมุมที่เราต้องการจะหมุนวัตถุไปรอบๆ แกนหรือเส้นตรงที่เรากำลังกล่าวถึงอยู่ดังนั้นเมตริกซ์การแปลงแบบหมุนรอบแกน Z คือ

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

ดังนั้นเมตริกซ์การแปลงแบบหมุนรอบแกนใดๆ R_0 คือ

$$R_0 = T R_X R_Y R_Z R_Y^{-1} R_X^{-1} T^{-1}$$

2.4 A Simple Illumination Model

สมการเบื้องต้นที่ใช้

$$I=I_l k_d \cos\theta \quad (0<k_d<1) \quad 0\leq\theta\leq\pi/2$$

I คือ (Intensity) ความเข้มของแสงสะท้อนซึ่งมีผลต่อการลงสีให้วัตถุ

ซึ่งนอกจากนี้ยังมีตัวแปรอื่น ๆ ที่มีความสำคัญซึ่งมีผลต่อค่าของ I ดังสมการ

$$I = I_a k_a + I_e/d + k[k_d(n.L) + k_s(RS)^n]$$

- เช่นความยาวของคลื่นแสง
- ความสามารถในการสะท้อนแสงของวัตถุ
- เวกเตอร์ข้างเคียง (S)
- เนื่องจาก shade สีมีค่าตั้งแต่ 0-255

ดังนั้นจึงกำหนดค่า I ให้อยู่ในช่วง 0-255 เพื่อจะได้นำค่า I ที่ได้ไปกำหนดสีที่จะลงบน polygon ของวัตถุซึ่งการลง shade สีแบบนี้เป็นวิธี constant shadding จะได้สมการเป็น

$$I = 255\cos\theta$$

แต่เนื่องจาก $\cos\theta = n.L/|n||L|$

n คือ normal เวกเตอร์

L คือ Reflox เวกเตอร์

Ie คือ Reflex Ray

บทที่ 3
การออกแบบระบบและขั้นตอนการดำเนินงาน

3.1 กระบวนการและการพัฒนาระบบ

3.1.1 การศึกษาค้นคว้า

1. ศึกษาหลักการทางคณิตศาสตร์และหลักการ Computer Graphics ในต่อไปนี้
    - การแปลงภาพเรขาคณิต 2 มิติ และ 3 มิติ
    - สมการ Quadratic Surface และ Torus
    - การซ่อนเส้นของรูป
    - การใส่สีของรูป
    - การติดต่อกับจอภาพเพื่อแสดงผล
2. ศึกษาโปรแกรม Visual C++ เวอร์ชัน 6.0 เพื่อนำมาใช้ในการเขียนโปรแกรม

3.1.2 ขั้นตอนการดำเนินการ

1. ศึกษาหลักการทางคณิตศาสตร์ และความรู้ทาง Computer Graphic ที่ต้องนำมาใช้ในการวางแผน ออกแบบ และพัฒนาระบบ

2. ศึกษาระบบ และวิธีการใช้งาน Application Development Tools เพื่อให้เหมาะสม และมีความสามารถที่ใช้พัฒนาระบบได้ โดยเลือก Visual C++ เวอร์ชัน 6.0

3. ศึกษา Software ที่เกี่ยวกับ Computer Graphic ที่มีอยู่ในปัจจุบัน เพื่อศึกษา และนำมาประยุกต์ใช้ในพัฒนาโปรแกรม

4. ทำการพัฒนาระบบงานจริง โดยแบ่งเป็นขั้นตอนดังนี้
    - ออกแบบ Flow Chart ของระบบงาน
    - กำหนด Input , Output ของระบบ
    - ออกแบบ User Interface
    - ดำเนินงานพัฒนาโปรแกรม
    - ให้ผู้ใช้งานทดลองใช้ เพื่อหาข้อผิดพลาดและข้อเสนอแนะแล้วนำมาปรับปรุงโปรแกรม ให้ได้ตรงตามความต้องการ ใช้งานง่ายมากที่สุด
    - จัดทำเอกสารประกอบการใช้งาน

3.1.3 ความรู้เบื้องต้นเกี่ยวกับ Visual C++ เวอร์ชัน 6.0

โดยทั่ว ๆ ไปแล้วภาษาคอมพิวเตอร์ ประกอบด้วย 2 แนวความคิดคือ ข้อมูล(Data) และ อัลกอริทึม (Algorithm) ข้อมูลประกอบด้วยข่าวสารต่าง ๆ ที่โปรแกรมใช้ และทำกระบวนการ หรือโพรเซส(Process) อัลกอริทึม เป็นวิธีการ(Method) ที่โปรแกรมใช้ภาษา C เป็นภาษา procedure ดังนั้น จึงเน้นหนักไปทางด้าน อัลกอริทึม มากกว่า ส่วนทางด้านข้อมูลนั้นภาษา C ทำได้ไม่ดี เมื่อนำมาใช้ในการพัฒนาโปรแกรมขนาดใหญ่ ๆ ด้วยข้อจำกัดนี้ จึงได้มีการพัฒนา ภาษา C เป็นภาษาใหม่ นั่นก็คือ C++ ซึ่งเป็นภาษาเชิงวัตถุ (Object Oriented Language) โดยการนำเอาภาษา C บวกกับ การโปรแกรมเชิงวัตถุ (OOP : Object Oriented Programming)

คุณสมบัติของ OOP มีการเน้นไปทางด้านข้อมูลมากกว่าที่จะเน้นทางด้านภาษาแบบ procedure ดังนั้นภาษาแบบ OOP จะไม่ค่อยมีการเขียน Flowchart แสดงการทำงานกันนัก ยกเว้นในส่วน อัลกอริทึม ที่ยังใช้กันอยู่

เมื่อ ภาษา C++ ได้รับความนิยมมากขึ้น คุณสมบัติของ OOP ได้รวมกับการ Interface กับ Windows API ท้ายสุดทาง Microsoft ได้พัฒนา การ Interface นี้ จนกลายเป็น MFC (Microsoft Foundation Class) ซึ่งปรากฏ อยู่ในทุกวันนี้

ใน MFC จะประกอบด้วย Library MFC ซึ่งเป็นการรวมคลาสของC++ และ AFX จำนวนมาก ที่ถูกออกแบบมาเพื่อนำมาสร้าง Application บน Windows การรวม คลาส ต่างๆนี้ เพื่อขยายขอบเขต ของ C++ ออกไป โดยการรวมส่วนโครงสร้าง พื้นฐานต่าง ๆ ไว้เป็นหลัก ซึ่งส่วนโครงสร้างพื้นฐานนี้ จะถูกนำมาใช้สร้าง Application

MFC ถูกออกแบบให้ Interface Object Oriented แก่ระบบปฏิบัติการ Windows ที่สนับสนุนการนำโค้ดกลับมาใช้ได้ นอกจากนี้ MFC ยังมี คลาสต่าง ๆ จำนวนมากที่ห่อหุ้ม Windows Dialog box และ อื่นๆ มีผลให้การใช้ MFCมาพัฒนา GUI Application บน Windowsได้อย่างง่ายดายและรวดเร็ว

3.2 การออกแบบระบบ

ประกอบด้วย 3 ส่วนต่อไปนี้

1. ส่วนนำข้อมูลเข้า

เป็นข้อมูลที่มาจากผู้ใช้ โดยที่ผู้ใช้ใส่ข้อมูลที่ต้องใช้ในการคำนวณ ตามที่ได้กำหนดไว้ เช่น เลือกรูปที่ต้องการ เลือกสีของรูป เป็นต้น

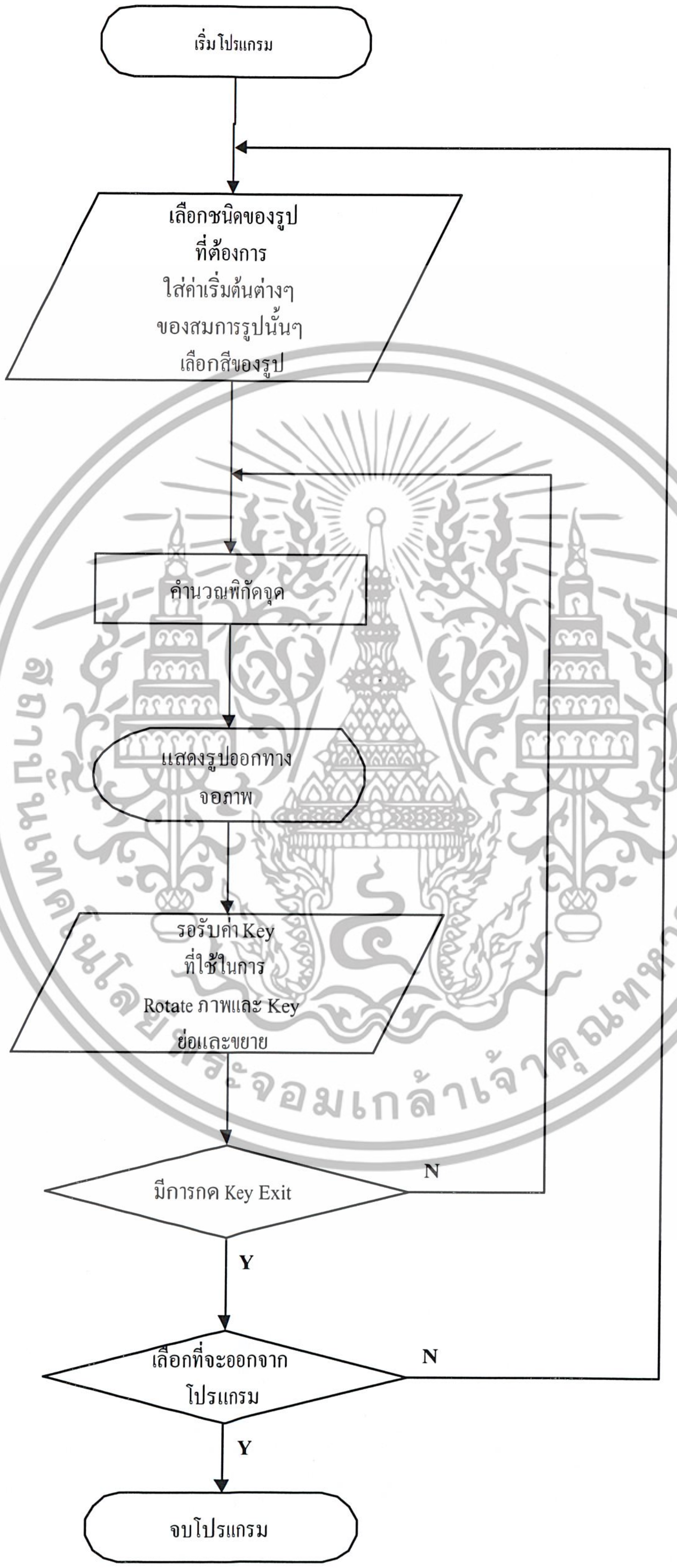
2. ส่วนวิเคราะห์

นำข้อมูลที่รับเข้ามา ทำการประมวลผล คำนวณข้อมูลภาพ และใส่สีรูป เก็บข้อมูลลงหน่วยความจำ

3. ส่วนแสดงผล

นำค่าข้อมูลจากข้อ 2 มาแสดงออกทางจอภาพ ในลักษณะของรูป 3 มิติ ซึ่งจะสามารถหมุนหรือย่อขยายภาพที่แสดงอยู่นั้นได้

3.3 ขั้นตอนการทำงานของโปรแกรม

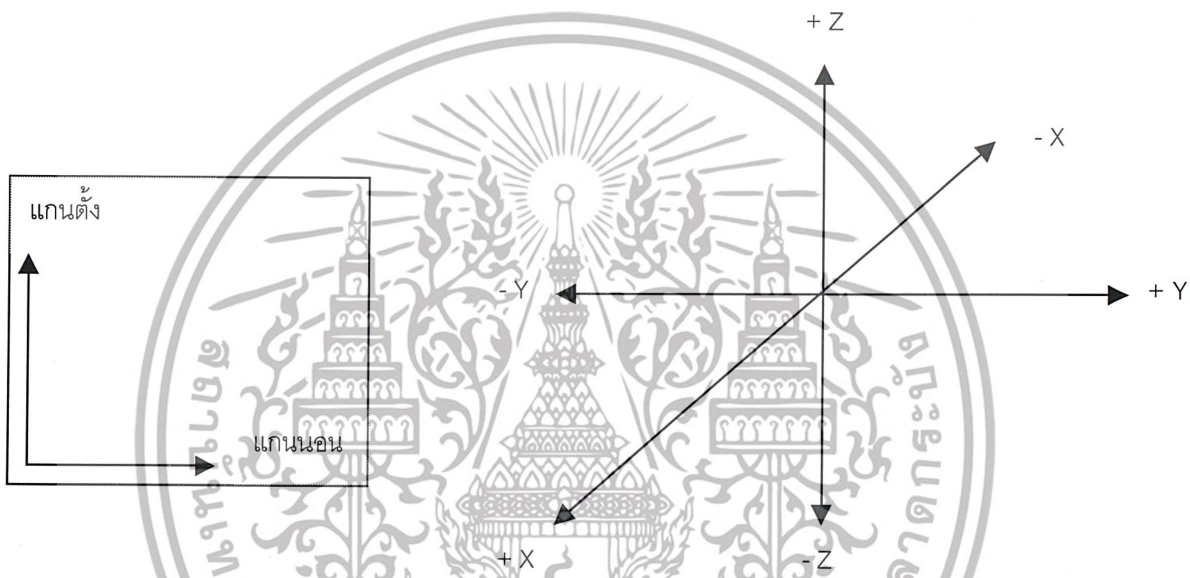


3.4 ความรู้ทางคณิตศาสตร์ที่นำมาประยุกต์ใช้

3.4.1 การสร้างแกนของระบบภาพ 3 มิติ

การสร้างภาพ 3 มิติบนหน้าจอ 2มิตินั้นเป็นไปได้ยาก เพราะ ในระบบภาพ 3 มิตินั้นมี 3 แกน คือ แกน X แกนY และ แกน Z ซึ่งบนหน้าจอนั้นจะมีแกนได้แค่ 2 แกน ดังนั้นเราจะสร้างแกนที่ 3 บนหน้าจอได้อย่างไร

วิธีการสร้างแกนที่ 3 ของจอภาพนั้นเราใช้ความรู้ทางคณิตศาสตร์มาช่วยได้

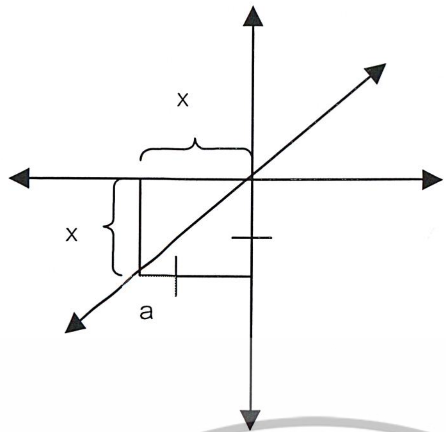


รูปที่ 3.1 การแปลงจอภาพ 2 มิติเป็นแกนในระบบ 3 มิติ

กำหนดให้แกนนอนของหน้าจอแทน แกน Y และแกนตั้งของหน้าจอแทน แกน Z โดยที่ แกน X จะเป็นแกนที่อยู่ในแนวขวาง กำหนดให้เป็นระบบโคออร์ดิเนตในระบบมือขวา

ในการแสดงจุดบนจอภาพเราทำได้แค่ระบุแกนนอนและแกนตั้ง แต่ไม่สามารถระบุค่าแกนในแนวขวางลงไปได้โดยตรง จึงจำเป็นต้องแปลงค่าของแกนในแนวขวางให้เป็นค่าในแกนนอนและแกนตั้งก่อน

ตัวอย่างเช่น กำหนด จุด (a , b , c) คือจุดที่จะแสดง ในการแสดงแกน Y ทำได้โดยลบค่าบนแกนไปเป็นระยะ b หน่วย เช่นเดียวกันกับแกน Z ทำได้โดยการบวกค่าบวกค่าเป็นระยะ c หน่วย สำหรับค่าของ แกน X เมื่อกำหนดให้แกน X ทำมุม 45 องศา กับแนวนอนบนจอภาพดังนั้น



เมื่อระยะทางบนแกน X เป็น a ดังนั้นเราจะหา x ได้จาก

$$a = \sqrt{\left(x^2 + x^2\right)}$$

$$a^2 = 2x^2$$

$$x = \sqrt{\left(a^2 / 2\right)}$$

ดังนั้นในการแสดงจุดบนแกน X ทำได้โดยการนำค่า x ไปลบบนแกนในแนวนอนและแกนในแนวตั้ง

3.4.2 การสร้างภาพให้เป็น 3 มิติ

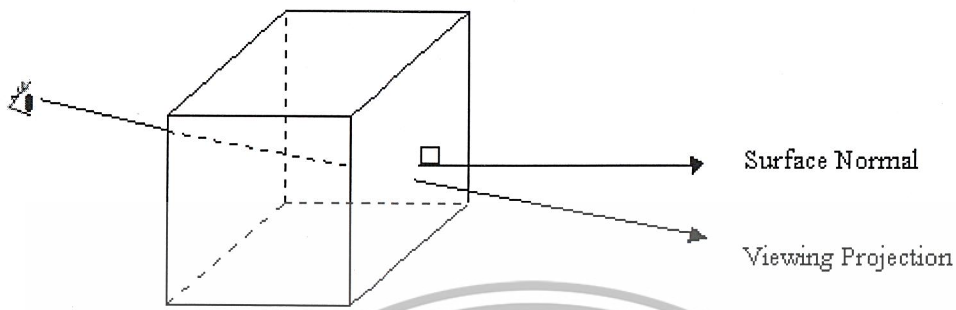
ในการสร้างภาพ 3 มิติของปัญหาพิเศษนี้เราจะสร้างภาพ โดยพิจารณาจากสมการที่อยู่ในรูปรูปของรัศมี มุมละติจูด และ มุมลองติจูด เพราะจะทำให้เราสร้างภาพได้ละเอียดยิ่งขึ้น คุณสมบัติที่ได้จากสมการคือ ค่ามุมระหว่างแกนซึ่งเรานำมาใช้ในการสร้างภาพด้วยวิธีหมุนรอบแกน

ในปัญหาพิเศษฉบับนี้เราจะใช้วิธีการสร้างภาพในระนาบ XZ ก่อน กำหนดให้ Y เป็นศูนย์ หลังจากนั้นนำจุดในระนาบ XZ มาหมุนรอบแกน Z ให้เกิดเป็นภาพ 3 มิติ วิธีการสร้างภาพนี้เป็นการสร้างแบบง่ายไม่ซับซ้อน ซึ่งจะมีข้อจำกัดเกิดขึ้นคือ ระยะบนแกน X จะเท่ากับแกน Y

ค่าของมุมที่ใช้หมุนเกิดจากการเอาค่า n ไปหาร 2π ทำให้เกิดค่าของมุมที่จะหมุน ซึ่งค่าของมุมนี้ ถ้ายิ่งมีค่าน้อยเท่าไหร่ก็จะทำให้ภาพละเอียดมากยิ่งขึ้น ซึ่งเป็นเรื่องที่ดี แต่เป็นข้อเสียคือจำนวนจุดที่จะแสดงจะมีค่ามากซึ่งจะเป็นการเปลืองทรัพยากร

3.4.3 การซ่อนเส้น

วิธีการซ่อนเส้นนั้นมีด้วยกันหลายวิธี เราเลือกใช้วิธีการ Backface Removal ซึ่งเป็นวิธีการที่ง่ายต่อความเข้าใจไม่ซับซ้อน และเป็นการนำเอาความรู้ทางคณิตศาสตร์มาประยุกต์ได้ดีที่สุด



รูปที่ 3.2 ภาพพื้นด้านหลังที่ถูกซ่อน

มีขั้นตอนการคำนวน ดังนี้

1. หา Surface Normal หาได้จากการเอาเวคเตอร์ของขอบของ surface ที่อยู่ติดกันมา Cross Productกัน สมมติเวคเตอร์ A =< x1 , y1 , z1 > และ เวคเตอร์ B=< x2 , y2 , z2 > ดังนั้น

$$A \times B = < y1*z2 - y2*z1 \ , \ x2*z1 - x1*z2 \ , \ x1*y2 - x2*y1 >$$

2. หา Viewing Direction เพื่อนำไป Dot Product กับ Surface Normal สมมติเวคเตอร์ A = < x1 , y1 , z1 > และ เวคเตอร์ B = < x2 , y2 , z2 > ดังนั้น

$$A.B = x1*x2 + y1*y2 + z1*z2$$

เนื่องจาก

$$A.B = |A||B|\cos(\theta)$$

- Cos (θ) มีค่ามากกว่า 0 ถ้าทำ θ มุมน้อยกว่า 90 องศา
- Cos (θ) มีค่าน้อยกว่า 0 ถ้าทำ θ มุมมากกว่า 90 องศา
- Cos (θ) มีค่าเท่ากับ 0 ถ้าทำ θ มุมเท่ากับ 90 องศา

เพราะฉะนั้น A.B จะ มากกว่า 0 ถ้า Cos (θ) > 0 นั่นคือ θ < 90 องศา

ดังนั้น เมื่อเรานำ Surface Normal ไป Dot Product กับ Viewing Direction แล้ว ถ้าค่าที่ได้น้อยกว่าศูนย์ นั่นคือจะเป็น Backface จะซ่อนไว้

3.5 ความต้องการด้านฮาร์ดแวร์และซอฟต์แวร์

ความต้องการด้านฮาร์ดแวร์

ลักษณะของอุปกรณ์คอมพิวเตอร์ที่นำมาใช้ในระบบนี้ ควรมีรายละเอียดดังนี้

1. เครื่องคอมพิวเตอร์
   - หน่วยประมวลกลาง CPU Pentium 100 MH หรือสูงกว่า
   - หน่วยความจำหลัก (Main Memory) ขนาดไม่ต่ำกว่า 16 เมกะไบต์
   - หน่วยความจำความเร็วสูง (cache Memory) ชนิดภายในหน่วยประมวลผลกลางขนาด ไม่น้อยกว่า 16 กิโลไบต์ และชนิดภายนอกหน่วยประมวลผลกลางขนาดไม่น้อยกว่า 256 กิโลไบต์
   - เครื่องขับจานแม่เหล็กอย่างอ่อน (Floppy Disk Drive) ขนาด 3.5 นิ้ว จำนวน 1 หน่วย
   - จอภาพสีชนิดรายละเอียดสูง ขนาดไม่ต่ำกว่า 14 นิ้วตามเส้นทะแยงมุม ซึ่งสามารถใช้แสดงภาพที่ได้รับจากวงจรแสดงผลกราฟฟิก รายละเอียดไม่น้อยกว่า 1,024*768 pixels แบบ Non-Interlace
   - คีย์บอร์ด (Keyboard) ที่มีอักษรภาษาไทย / ภาษาอังกฤษ ตัวเลขและเครื่องหมายสัญลักษณ์พิเศษอย่างน้อย 101 คีย์
   - มีอุปกรณ์ป้อนคำสั่งแบบเมาส์ (Mouse) ที่มีปุ่ม 2 ปุ่ม หรือ 3 ปุ่ม
2. เครื่องพิมพ์ ซึ่งมีลักษณะดังนี้
   - มีความละเอียดในการพิมพ์ไม่น้อยกว่า 600*600
   - มีหน่วยความจำไม่น้อยกว่า 16 เมกกะ ไบต์
   - มีความเร็วในการพิมพ์ 20 แผ่นต่อนาที

ความต้องการทางด้านซอฟต์แวร์

ซอฟต์แวร์ที่ใช้ในการพัฒนาโปรแกรมก็จะมี Windows 97 , Visual C++

บทที่ 4

การประเมินผล

ผลลัพธ์ที่ได้จากการพัฒนาโปรแกรมต้นแบบ " การจำลองภาพ 3 มิติ" สามารถประเมินผลในแต่ละด้านได้ดังนี้

4.1 ความเข้าใจในการสร้างซอฟท์แวร์จากภาษา Visual C++

ในการสร้างซอฟท์แวร์นั้น จะต้องมีการวางแผนขั้นตอนการทำงานที่เป็นระบบทั้งนี้ก็เพื่อให้กระบวนการสร้างซอฟ์แวร์สำเร็จลุล่วงทันเวลาที่กำหนดและบรรลุตามจุดประสงค์ที่วางไว้ ผู้สร้างซอฟท์แวร์จะต้องพิจารณาภาษาในการเขียนที่เหมาะสมเพื่อให้ซอฟท์แวร์มีประสิทธิภาพสูงสุด และจะต้องทำความเข้าใจในภาษานั้น ๆ เพื่อจะได้นำคุณสมบัติของภาษานั้นมาพัฒนาซอฟท์แวร์ได้อย่างมีประสิทธิภาพ

ซึ่งหลังจากได้ทดลองสร้างโปรแกรมต้นแบบ " การจำลองภาพ 3 มิติ" ผู้สร้างสามารถใช้งานภาษา Visual C++ 6.0 ได้ดียิ่งขึ้น และเข้าใจในการวางแผนงานในการพัฒนาโปรแกรม ซึ่งจะเป็นประโยชน์ในการพัฒนาโปรแกรมต่อไปในอนาคตเพื่อให้ได้ซอฟท์แวร์ ที่มีสิทธิภาพมากสูงสุด

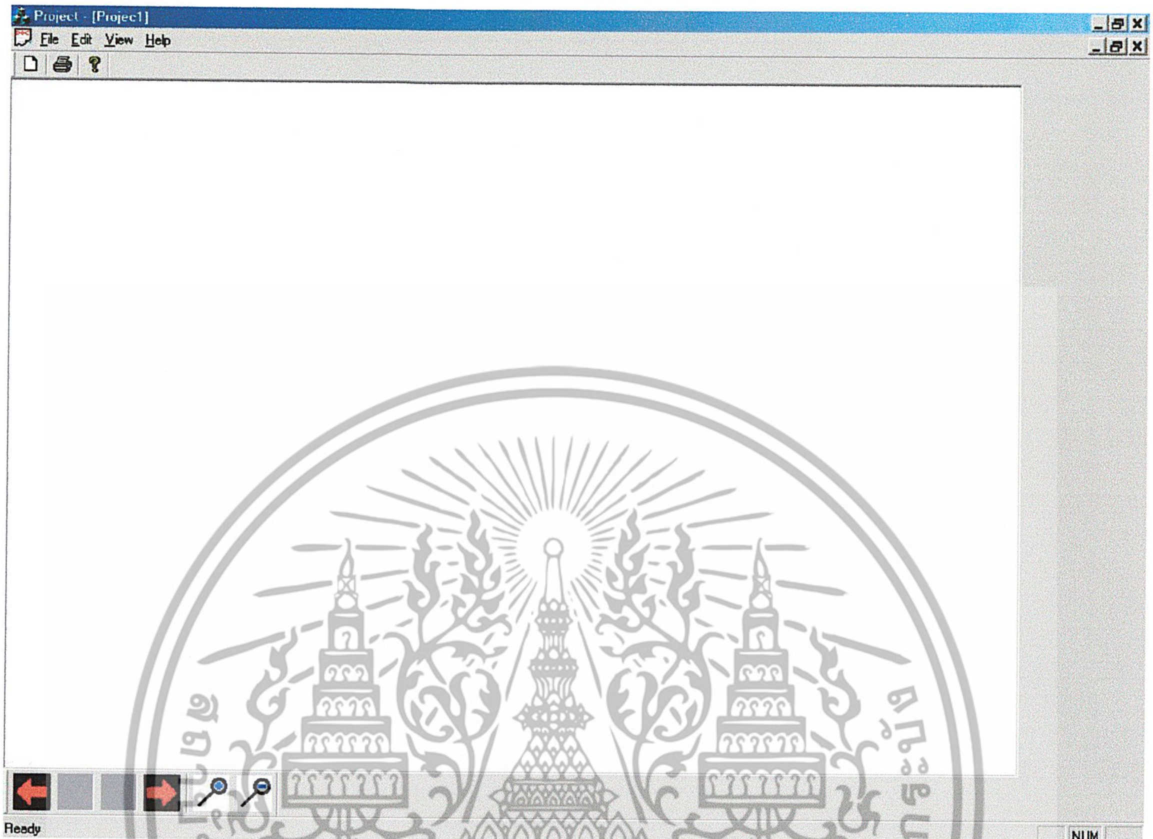
4.2 การนำความรู้ทางวิชาคณิตศาสตร์มาประยุกต์ใช้ในทางคอมพิวเตอร์

ความรู้ทางคณิตศาสตร์เป็นความรู้ที่สามารถนำมาประยุกต์ในงานได้หลากหลาย ไม่ว่าจะเป็นเทคโนโลยีการเกษตรหรือในทางวิศวกรรม สำหรับเทคโนโลยีทางด้านคอมพิวเตอร์ ก็มีการนำความรู้ทางคณิตศาสตร์มาใช้บ่อยครั้ง

ผู้สร้างได้ทดลองนำความรู้ทางคณิตศาสตร์มาประยุกต์ใช้ทางด้านคอมพิวเตอร์กราฟฟิกเพื่อเขียนalgorithm สำหรับการสร้างภาพจำลอง 3 มิติ และการหมุนภาพ ทำให้ทราบถึงความสามารถของวิชาคณิตศาสตร์และเข้าใจในวิธีการประยุกต์ที่เหมาะสม

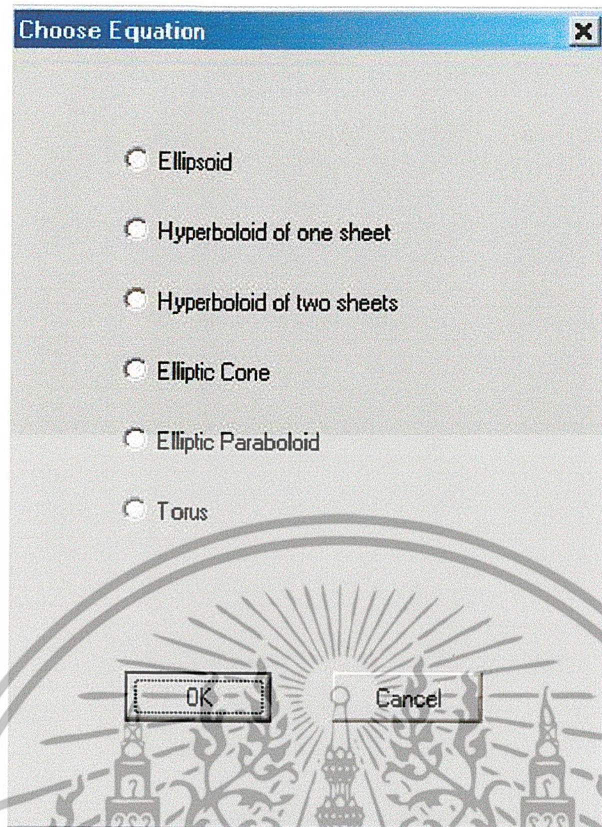
4.3 การทำงานของโปรแกรม

1. เมื่อทำการ Run โปรแกรมต้นแบบ จะปรากฏหน้าต่างดังรูป



รูปที่ 4.1 หน้าจอหลัก

2. จากนั้นให้ผู้ใช้ทำการเลือกภาพที่จะจำลอง โดยการกดเลือกเมนู File -> Equation หรือ Ctrl + E เมื่อทำการเลือกจะปรากฏหน้าต่างต่อไปนี้

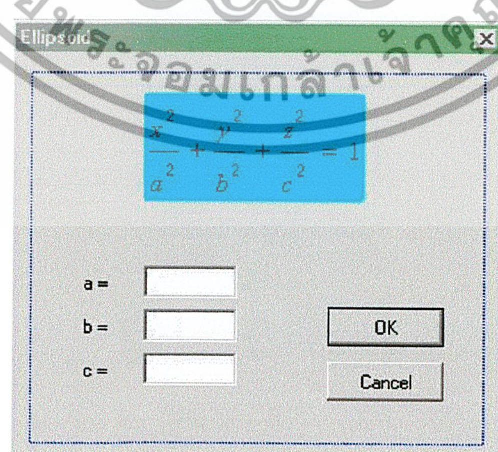


รูปที่ 4.2 หน้าจอเลือกสมการ

เมื่อปรากฏหน้าต่างนี้ต้องทำการเลือกสมการ ที่ต้องการ แล้วคลิกปุ่ม OK

3. หลังจากเลือกภาพที่จะจำลอง จะปรากฏหน้าต่าง property ของภาพที่จะจำลอง ผู้ใช้จำเป็นต้องใส่ข้อมูลของภาพ

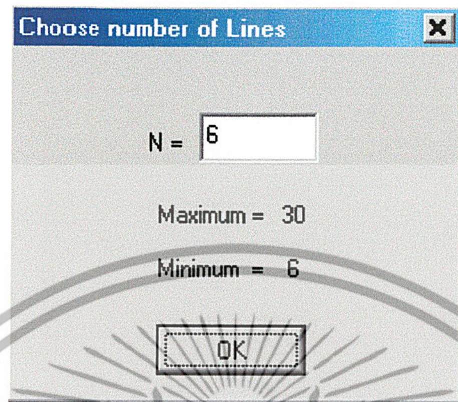
ตัวอย่างเช่น ถ้าผู้ใช้เลือก Ellipsoid จะปรากฏหน้าจอ ดังนี้



รูปที่ 4.3 หน้าจอ property ของ Ellipsoid

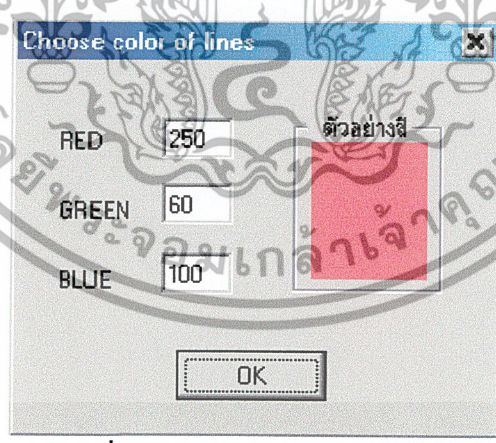
ในหน้าจอนี้จะให้ผู้ใช้ต้องใส่ค่า a , b และ c ซึ่งเป็นความยาวในแกนต่าง ๆ ของภาพ

4. เมื่อทำการกดปุ่ม OK จะปรากฏหน้าต่างใหม่ซ้อนขึ้นมาเพื่อรับค่าจำนวนเส้น โดยที่จะกำหนดค่าต่ำสุดคือ 6 และค่าสูงสุดคือ 30 เพื่อนำมาคำนวณหาจำนวนเส้นรอบรูป โดยค่าจำนวนเส้นนี้จะมีผลต่อความละเอียดของภาพที่จะถูกแสดง ถ้าค่าจำนวนเส้นน้อย จะทำให้ภาพที่ได้มีความละเอียดต่ำ แต่ถ้าค่าจำนวนเส้นมาก จะทำให้ภาพที่ได้มีความละเอียดสูงขึ้น



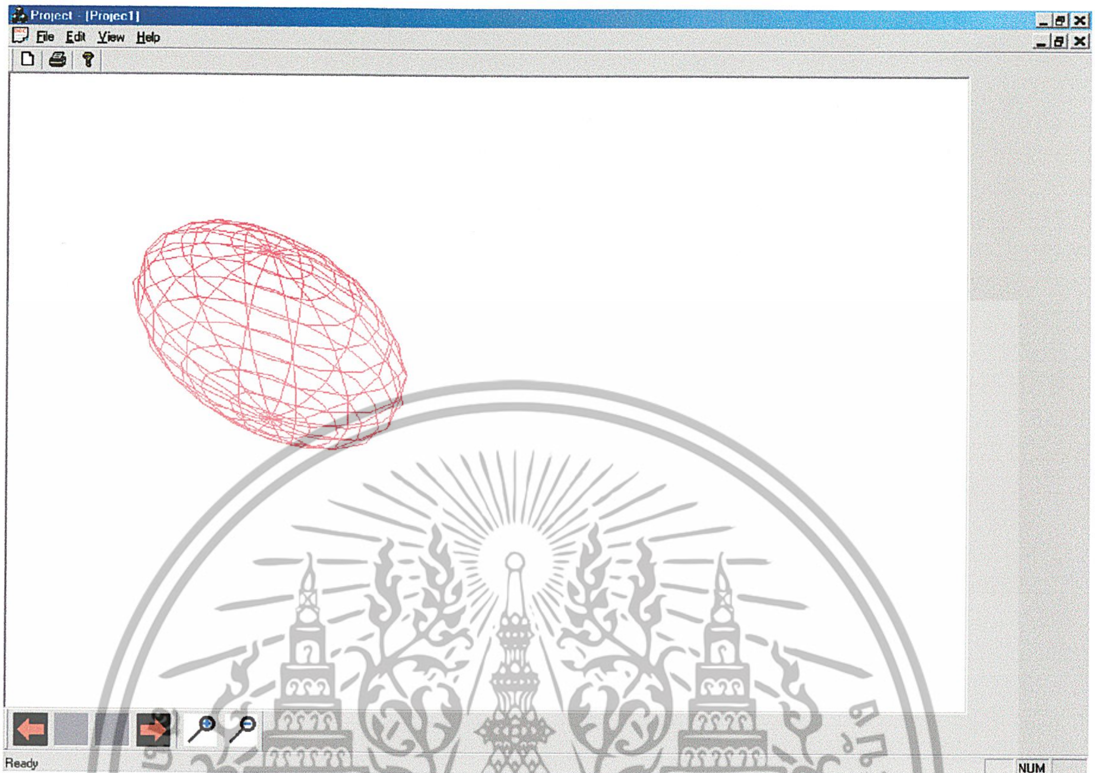
รูปที่ 4.4 หน้าจอ property รับค่าจำนวนเส้น

5. เมื่อทำการกดปุ่ม OK จะปรากฏหน้าต่าง ให้เลือกสี โดยที่ผู้ใช้จะต้องระบุในระบบ Macro RGB คือให้ใส่ระดับของแม่สี ได้แก่ สีแดง สีเขียว และสีน้ำเงิน โดยสีแต่ละสีจะมีค่าตั้งแต่ 0 ถึง 255 ดังตัวอย่างรูปที่ 4.5 เราทำการ ใส่ค่าของสีแดงเท่ากับ 250 สีเขียวเท่ากับ 60 และสีน้ำเงินเท่ากับ 100 เราสามารถดูสีที่ได้ทำการผสมสีแล้วในช่อง " ตัวอย่างสี "



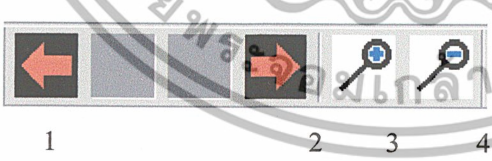
รูปที่ 4.5 หน้าต่างกำหนดสีของเส้น

6. เมื่อทำการกดปุ่ม OK จะปรากฏหน้าต่าง ดังรูป



รูปที่ 4.6 หน้าจอโปรแกรมเมื่อทำการใส่ข้อมูลครบแล้ว

ผลที่ได้จะเป็นภาพ 3 มิติ ผู้ใช้สามารถหมุนภาพได้ด้วย ทูลบาร์ต่อไปนี้



เมื่อ ปุ่ม 1 หมุนภาพไปทางซ้าย

ปุ่ม 2 หมุนภาพไปทางขวา

ปุ่ม 3 ขยายภาพ

ปุ่ม 4 ย่อภาพ

4.4 ปัญหาที่พบในการทำปัญหาพิเศษ

1. ในการเขียน algorithm จำลองการสร้างภาพนั้น สำหรับกรณีของ Hyperbolic Cylinder , Elliptic Cylinder และ Parabolic Cylinder เนื่องจากรูปสมการนั้นไม่มีค่า Z จึงไม่สามารถที่จะนำมาจำลองได้ และกรณีของ Hyperbolic Paraboloid นั้นเนื่องจากวิธีการในการสร้างภาพของปัญหาพิเศษฉบับนี้เป็นการสร้างโดยการหมุนภาพเป็น 3 มิติ แต่ Hyperbolic Paraboloid นั้นมีรูปร่างที่ซับซ้อนไม่สามารถจะนำมาหมุนได้โดยตรง จึงจำเป็นต้องตัดออกไปจากปัญหาพิเศษนี้
2. ในกระบวนการซ่อนเส้นนั้น มีวิธีการซ่อนเส้นหลายวิธีด้วยกัน ซึ่งในตอนแรกผู้จัดทำได้เลือกที่ใช้วิธีการ Backface Removal ซึ่งเป็นวิธีการที่ง่ายต่อความเข้าใจไม่ซับซ้อน ผู้ใช้จึงมุ่งศึกษาวิธีนี้ ปัญหาที่เกิดขึ้นในขั้นตอนนี้คือ เราไม่สามารถที่จะหา Viewing direction ที่ถูกต้องได้ ทำให้การซ่อนเส้นของภาพนั้นได้ผลลัพธ์ที่ไม่ถูกต้อง อีกทั้งยังมีผลให้กระบวนการ Render หรือการลงสีภาพนั้นไม่สามารถทำได้ จึงทำให้โปรแกรมนี้ขาดคุณสมบัติ การซ่อนเส้นและใส่สี
3. การเก็บข้อมูลทำได้ไม่รัดกุมจึงทำให้มีข้อมูลมากเกินไป ทรัพยากรคอมพิวเตอร์ที่มีอยู่จึงไม่เพียงพอ ทำให้เราไม่สามารถใช้โปรแกรมได้คุ้มค่า เต็มประสิทธิภาพได้
4. ยังไม่สามารถพิมพ์ออกทางเครื่องพิมพ์ได้ เนื่องจากไม่สามารถระบุตำแหน่งที่ถูกต้องของรูป กับตำแหน่งบนหน้าจอได้

บทที่ 5
สรุป วิจารณ์ และแนวทางในการพัฒนาระบบ

5.1 ความสามารถของโปรแกรม

1. โปรแกรมจะให้ผู้ใช้เลือกภาพที่จะทำการจำลองเป็น 3 มิติพร้อมทั้งใส่ข้อมูลที่สำคัญของภาพ แล้วนำไปประมวลผลจะเกิดรูปใน 3 มิติ
2. รูปใน 3 มิติ ที่สร้างมานั้น สามารถให้ผู้ใช้หมุนซ้าย – ขวา ดูได้รอบทิศทาง
3. สามารถ ย่อ ขยาย ขนาดของรูปได้(zoom in / zoom out)

5.2 ข้อจำกัดของโปรแกรม

1. ภาพที่จะสร้างต้องเลือกจากรายการในระบบเท่านั้น
2. เนื่องจากโปรแกรมจำเป็นต้องใช้ memory สูงจึงเกิดข้อจำกัดในการหมุน เช่น ถ้ากำหนดจำนวนเส้นของ n มากเกินไปก็จะทำให้เราหมุนภาพได้น้อยครั้ง
3. รูปที่สร้างได้ยังไม่สามารถทำให้คล้ายกับวัตถุจริงมากเท่าที่ควร เนื่องจากยังไม่ได้มีการปรับ curve และ surface ของ จุด (object ของ วัตถุ)

5.3 แนวทางในการพัฒนาต่อไป

1. ควรพัฒนาให้สามารถจำลองภาพได้หลากหลายขึ้น เพื่อสนับสนุนผู้ใช้ทุกประเภท
2. หากต้องการเพิ่มความละเอียดของรูปที่ได้ ให้มีความคล้ายของจริงมากยิ่งขึ้นต้องใช้ทฤษฎีปรับ curve และ surface ซึ่งต้องเพิ่มความละเอียดของจุให้มากกว่านี้ จึงต้องมี hard ware ที่มีประสิทธิภาพสูงด้วย
3. ควรนำความรู้ทางคอมพิวเตอร์ ในเรื่องของการจัดการทรัพยากรมาประยุกต์ใช้ในการออกแบบการเก็บข้อมูลภาพด้วยเพื่อที่จะได้ไม่เป็นการสิ้นเปลือง memory ทำให้เราสามารถประมวลผลได้หลากหลายขึ้น
4. ควรพัฒนาเพื่อให้ได้ภาพที่ถูกซ่อนเส้นและมีการลงสี ให้ได้ภาพที่มีความคล้ายของจริงยิ่งขึ้น
5. ความสามารถในการหมุนหรือการทำให้ภาพเคลื่อนที่ควรพัฒนาให้ทำได้หลายแบบ หลายวิธีมากยิ่งขึ้น

บรรณานุกรม

นิรุธ อำนวยศิลป์. 2521. **คู่มือการเขียนโปรแกรม Microsoft Visual C++ Version 6.0 ฉบับเพื่อการใช้งานจริง**. กรุงเทพฯ : ซัคเซส มีเดีย.

ยุทธนา ลีลาวัฒนกุล. 2544. **คู่มือการเขียนโปรแกรม และใช้งาน Visual C++ 6.0 ฉบับโปรแกรมเมอร์**. กรุงเทพฯ : อินโฟเพรส.

David F. Rogers , J.Alan Adam. **Mathematical Elements for Computer Graphics**. McGraw Hill. 1990.

David F. Rogers , McGraw_Hill Co. 1985. **Procedural Element for computer graphics**.

Foley , VanDam , Feiner , Hughe. 1992. **Computer Graphics principles and Practice.** Addison - Wesley Publishing company , Inc.

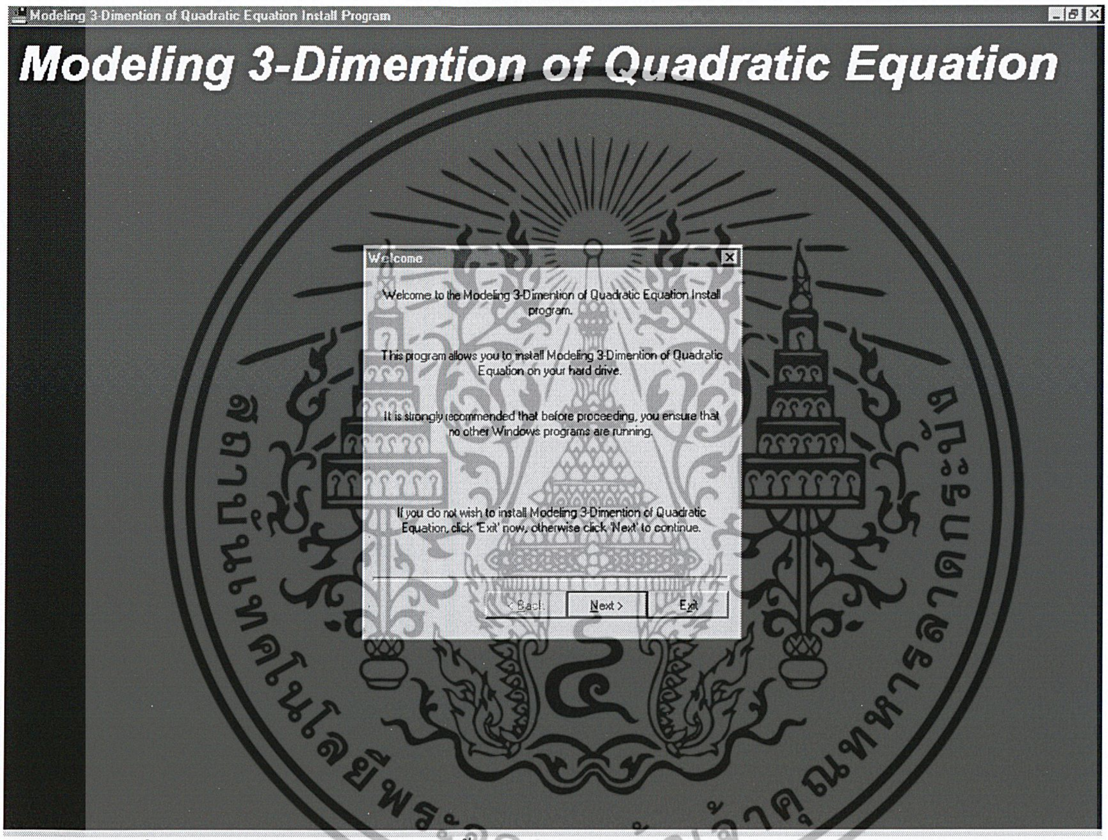
James D. Foley , Andries van Dam , Steven K.Feiner , John F. Hughes , Richard L.Phillips. 1994. **Introduction To Computer Graphics**. Addison-Wesley Publishing company , Inc.

Rogess , D.F. **Procedural Element for computer graphics**. McCraw Hill Co. 1995.

ภาคผนวก
ขั้นตอนการติดตั้งและการเรียกใช้งานโปรแกรม

ขั้นตอนการติดตั้งโปรแกรม

1. ดับเบิ้ลคลิกไฟล์ Setup 3D.exe เพื่อเข้าสู่ระบบการติดตั้งโปรแกรม จะได้ดังภาพ

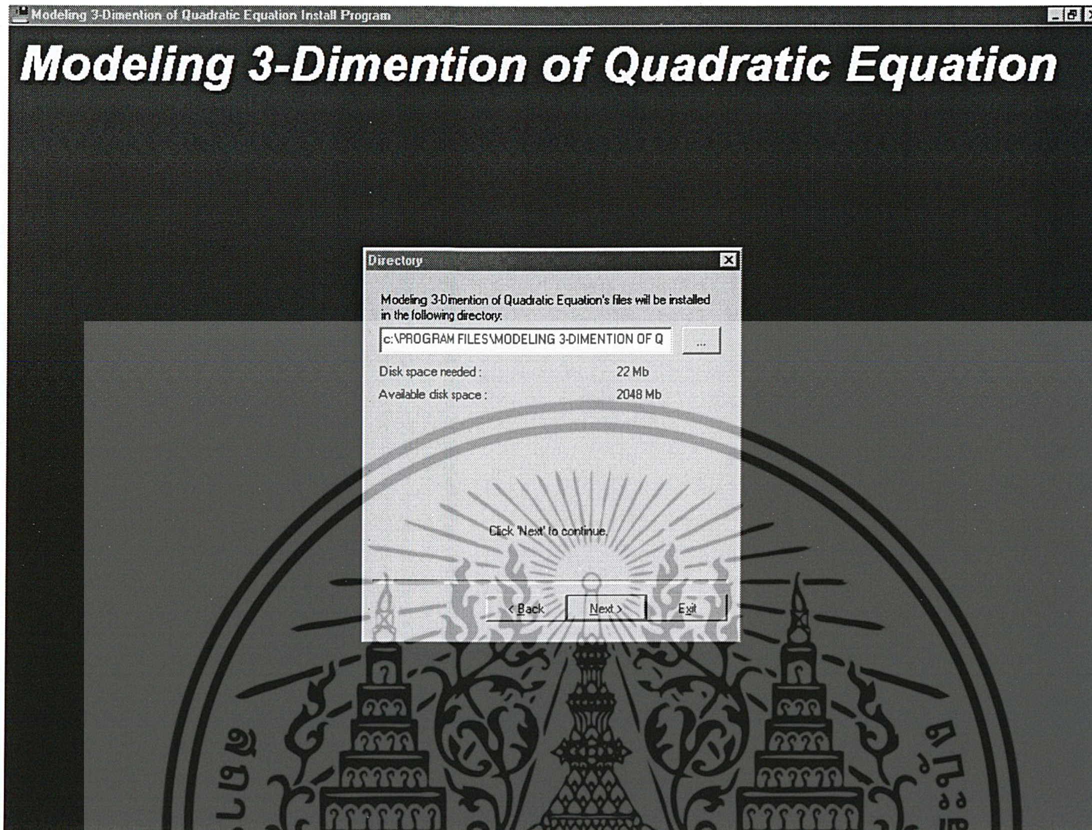


รูปที่ ผ. 1 เริ่มเข้าสู่ระบบการติดตั้งโปรแกรม

แล้วคลิก “ Next > “ เพื่อไปสู่หน้าถัดไป

2. เข้าสู่หน้า Information แล้วคลิก “ Next > “ เพื่อไปสู่หน้าถัดไป

3. เข้าสู่หน้า Directory ดังภาพ



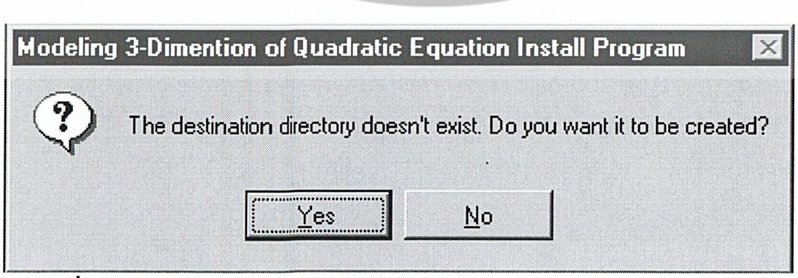
รูปที่ ผ. 2 เลือก Directory ของโปรแกรม

หน้านี้ผู้ใช้โปรแกรมต้องเลือก Directory ที่ต้องการจะติดตั้งตัวโปรแกรม โดยทางโปรแกรมจะตั้งอัตโนมัติ อยู่ที่ Directory

"C:\PROGRAM FILES\MODELING 3-DIMENTION OF QUADRATIC EQAUTION "

แต่ผู้ติดตั้งสามารถเปลี่ยนแปลงได้ตามต้องการ แล้วคลิก " Next > "

*** ถ้าโปรแกรมไม่พบ Directory ดังที่ระบุไว้ จะปรากฏ Dialog ดังภาพ



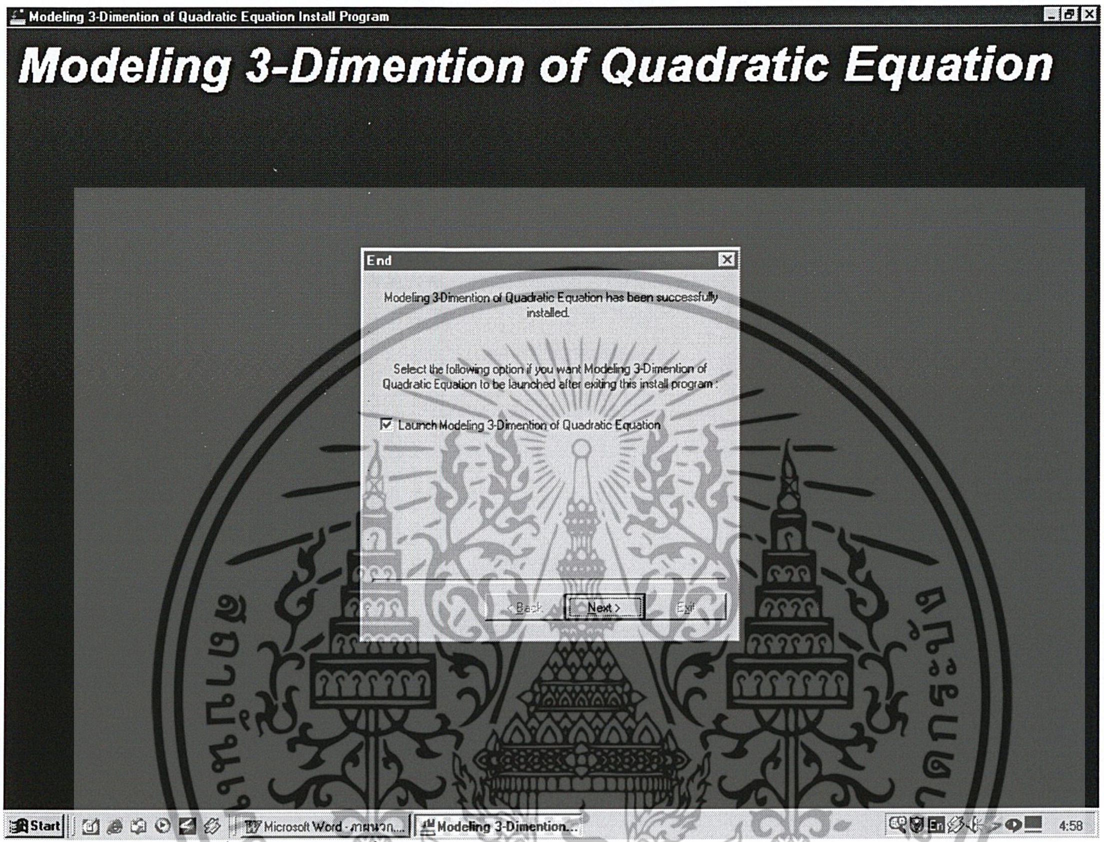
รูปที่ ผ. 3 เลือก Directory ของโปรแกรม

คลิก " Yes " เพื่อทำการสร้าง Directory ตามชื่อที่ได้ระบุขึ้นมาใหม่

คลิก " No " เพื่อยกเลิกการสร้าง Directory ใหม่ แล้วกลับไปเลือก Directory ใหม่

4. เข้าสู่หน้าเริ่มทำการติดตั้งโปรแกรม คลิก " Start " เพื่อเริ่มทำการติดตั้ง

5. โปรแกรมจะใช้เวลาในการติดตั้งช่วงเวลาหนึ่ง แล้วจะพบกับหน้า ดังรูป



รูปที่ ผ. 4 สิ้นสุดการติดตั้งโปรแกรม

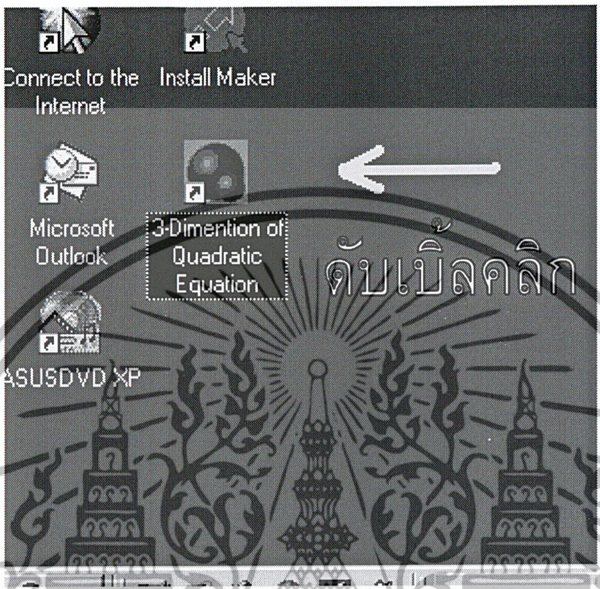
แล้วคลิก " Next > " เพื่อจบการติดตั้งโปรแกรม

5. แล้วคลิก " Exit " เพื่อออกจากการติดตั้งโปรแกรม

การเรียกใช้งานโปรแกรม

ผู้ใช้สามารถเรียกใช้งานโปรแกรมได้โดยง่าย คือ

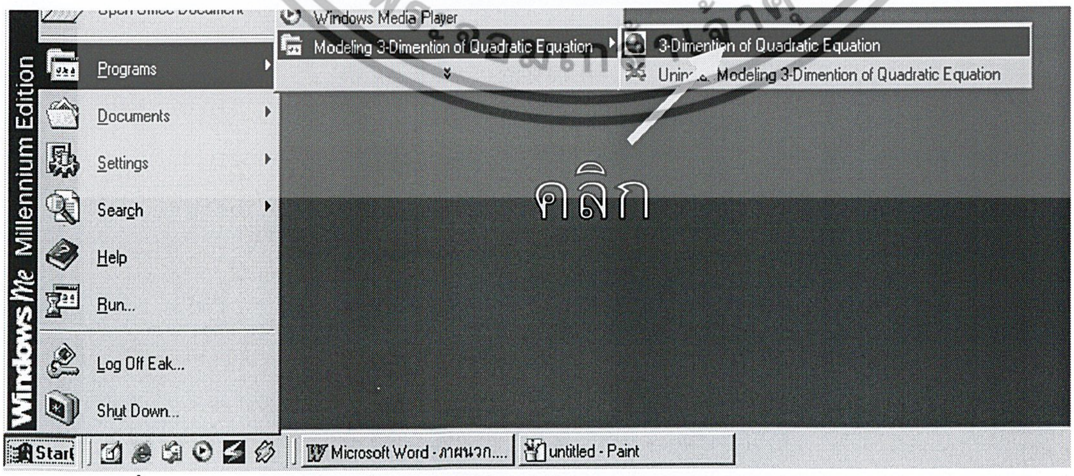
วิธีที่1 ดับเบิ้ลคลิกที่ Short Cut บน Desktop ของ Windows ที่ได้ทำการติดตั้งเรียบร้อยแล้ว ดังรูป



รูปที่ ผ. 5 Short Cut ของโปรแกรม

วิธีที่2 เลือกจาก เมนู Start ที่อยู่มุมด้านล่างซ้ายของ Windows โดยเลือก

คลิก Start -> Programs -> Modeling 3-Dimention of Quadratic Equation -> 3-Dimention of Quadratic Equation แล้วคลิก ดังรูป



รูปที่ ผ. 6 วิธีเปิดโปรแกรมโดยใช้ เมนู Start