การสร้างภาพ 3 มิติจากการมองภาพ 3 มุมมอง

นางสาว ฑัมพร วยาจุต 37054114
นางสาว บุศรินทร์ ชนะพันธ์ภากร 37054125
นางสาว สุนีย์ วลีธรชีพสวัสดิ์ 37054148

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2540

Create Three-Dimension From 3 views

Dhumporn	Vayachuta	37054114
Boosarin	Chanapanpakorn	37054125
Sunee	Waleetorncheepsawat	37054148

A Special Project Sumitted in Partial Fulfilment 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 Chaokhuntaharn Ladkrabang

1997

ปัญหาพิเศษเรื่อง การสร้างภาพ 3 มิติ จากการมองภาพ 3 มุมมอง

ชื่อนักศึกษา นางสาว ฑัมพร วยาจุต 37054114

นางสาว บุศรินทร์ ชนะพันธ์ภากร 37054125

นางสาว สุนีย์ วลีธรชีพสวัสดิ์ 37054148

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา ผศ. สุนทร สุชาติเวชภูมิ

อ. วิรัตน์ ศิริมังคลานุรักษ์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาประจำปีการศึกษา 2540 หลักสูตรวิทยาศาสตรบัณฑิต สาขาคณิตศาสตร์ประยุกต์

รศ. ภัคคินี ชิตสกุล

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะกรรมการโครงการพิเศษ

รศ.ดร. ไมตรี โพธิ์สุข

ประธานกรรมการ

อ. กฤษฎา ไตรสุรัตน์

กรรมการ

อ. วิรัตน์ ศิริมังคลานุรักษ์

กรรมการและอาจารย์ที่ปรึกษา

ผศ. สุนทร สุชาติเวชภูมิ

กรรมการและอาจารย์ที่ปรึกษา

หัวข้อปัญหาพิเศษ	การสร้างภาพ 3 มิติ จากการมองภาพ 3 มุมมอง		
นักศึกษา	ทัมพร	วยาจุต	
	บุศรินทร์	ชนะพันธุ์ภากร	
	สุนีย์	วลีธรชีพสวัสดิ์	
อาจารย์ที่ปรึกษา	ผศ.	สุนทร	สุชาติเวชภูมิ
	อ.	วิรัตน์	ศิริมังคลานุรักษ์
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์		
ปีการศึกษา	2540		

บทคัดย่อ

ปัญหาพิเศษฉบับนี้ จัดทำขึ้นโดยนำความรู้ทางด้านคณิตศาสตร์มาประยุกต์ใช้ทางคอมพิวเตอร์กราฟฟิค โดยใช้ในการสร้างซอฟแวร์สำหรับสร้างภาพ 3 มิติ จากการมองภาพ 3 มุมมอง โดยแบ่งการทำงานของโปรแกรมออกเป็น 4 ส่วน คือ การหาข้อมูลเบื้องต้นจากการมองภาพใน 2 มิติ และนำข้อมูลที่ได้มาใช้สร้างภาพ 3 มิติ หลังจากนั้นก็นำเสนอทางจอภาพ และส่วนสุดท้ายคือการบันทึกข้อมูลของภาพ 2 มิติจาก 3 มุมมองลงในแฟ้มข้อมูล

การสร้างซอฟแวร์นี้ ใช้ C++ Builder ในการเขียนโปรแกรมส่วนประมวลผล ในการพัฒนาซอฟแวร์เริ่มจากการศึกษาพื้นฐานทางด้าน Drawing เพื่อออกแบบซอฟแวร์ให้เหมาะสม ง่ายต่อความเข้าใจและการใช้งาน

Speacial Topic	Create Three Dimension From 3 Views
Student	Dhumporn Vayachuta
	Boosarin Chanapanpakorn
	Sunee Waleecheepsawat
Adviser	Mr. Sunthorn Suchatvejapoom
	Mr. Wirat Sirimangklanurak
Department	Mathematics and Computer Science
Year	1997

Abstract

This special project is a collection of the works on knowledge from computer graphic to apply for making software in creating three-dimensions from three views. By device algorithm of program to be 4 parts is basic entering form seeing picture in 3-D, create and present 3-D picture and save data of 2-D picture from seeing three view in file

In creating this software , use C++ BUILDER in writing program processing of development software , In the beginning learn basic in drawing for design software to suitable , easy understanding and using

กิตติกรรมประกาศ

ปัญหาพิเศษฉบับนี้ สำเร็จลงได้ด้วยดีก็เพราะหลายเหตุปัจจัย โดยเฉพาะอย่างยิ่ง

ผศ. สุนทร สุชาติเวชภูมิ
อ. วิรัตน์ ศิริมังคลานุรักษ์

ที่ได้ให้แนวทางในการดำเนินการ ตลอดจนคำปรึกษาอันก่อให้เกิดแนวความคิดที่สามารถแก้ไขปัญหาต่าง ๆ ที่เกิดขึ้นในระหว่างการทำปัญหาพิเศษฉบับนี้ นอกจากนี้ยังช่วยแนะแนวทางในการดำเนินงาน และตรวจทานแก้ไขด้วยความเอาใจใส่เป็นอย่างดียิ่ง

ขอขอบคุณเจ้าหน้าที่ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ ที่ให้ความสะดวกในการใช้ห้องปฏิบัติการคอมพิวเตอร์ และให้ความสะดวกในการเบิกอุปกรณ์ต่างๆ ที่ใช้ในการจัดทำปัญหาพิเศษ

คณะผู้จัดทำขอกราบขอบพระคุณอาจารย์ทุกท่านที่ประศาสน์วิชาความรู้ ทั้งในภาคทฤษฎีและภาคปฏิบัติแก่ผู้จัดทำ จนกระทั่งปัญหาพิเศษฉบับนี้สัมฤทธิ์ผลได้ด้วยดีทุกประการ

สารบัญ

สารบัญรูป

บทที่1

ความสำคัญและที่มาของปัญหาพิเศษ

เนื่องด้วยการศึกษาวิชา Drawing เป็นไปได้ยาก การนำ Computer เข้ามาเป็นสื่อในการเรียนการสอน จะเป็นการช่วยแก้ปัญหา และ เพิ่มประสิทธิภาพในการเรียนการสอนวิชา Drawing นี้ แต่เนื่องจากระบบที่มีอยู่มีราคาสูง และการใช้งานที่ค่อนข้างยุ่งยากซับซ้อน จึงไม่สามารถนำไปปฏิบัติได้อย่างเหมาะสม ดังนั้น จึงได้เห็นถึงความสำคัญในการพัฒนาระบบงานนี้ขึ้นมา เพื่อช่วยให้การเรียนการสอนมีประสิทธิภาพมากขึ้น

วัตถุประสงค์ของปัญหาพิเศษ

1. เพื่อพัฒนาระบบที่ช่วยในการเรียนการสอนวิชา Drawing เบื้องต้น
2. เพื่อให้เกิดความรู้ความเข้าใจในการใช้งาน และการออกแบบระบบบนระบบปฏิบัติการ Windows
3. เพื่อเป็นระบบเบื้องต้นสำหรับการพัฒนางานในด้านคอมพิวเตอร์กราฟฟิกต่อไป

ขอบเขตของปัญหาพิเศษ

1. ได้เครื่องมือที่สามารถใช้งาน เพื่อเพิ่มประสิทธิภาพในการเรียนการสอนในวิชา Drawing
2. สามารถสร้างภาพ 3 มิติจากข้อมูลของภาพจาก 3 มุมมอง

ประโยชน์ที่ได้รับ

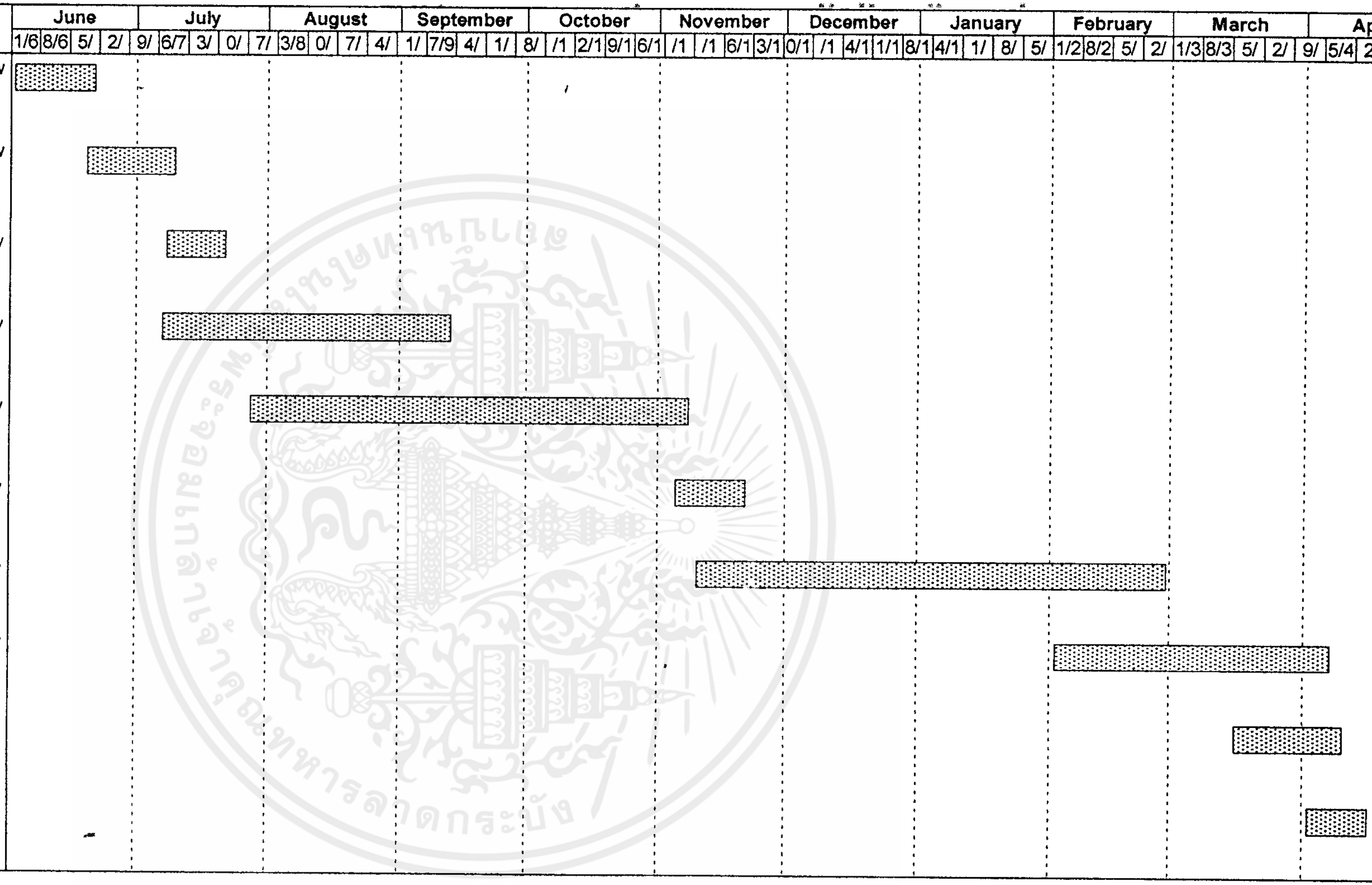
1. สามารถนำระบบไปช่วยในการเรียนการสอนวิชา Drawing ได้
2. นำความรู้ทางคณิตศาสตร์ไปประยุกต์ใช้ในงาน Computer Graphics

ขั้นตอนการดำเนินงาน

ศึกษาความรู้พื้นฐานเกี่ยวกับ Computer Graphic และ ภาษา C

- ออกแบบระบบงานและส่วนประกอบต่าง ๆ ของระบบ
- การพัฒนาระบบ
- ขั้นตอนทดสอบและแก้ไข
- สรุปประสิทธิภาพของระบบ และปัญหาที่เกิดขึ้น รวมถึงข้อเสนอแนะสำหรับการพัฒนาระบบต่อไป

ID	Task Name	Duration
1	ศึกษาปัญหาและที่มาของปัญหา	3w
2	ศึกษาเครื่องมือ Software ขอบเขตและความเป็นไปได้	3w
3	จัดทำแบบขออนุมัติทำปัญหาพิเ	2w
4	ศึกษา Software ที่ใช้	10w
5	ศึกษาทฤษฎีและหลักการที่เกี่ยว	15w
6	ออกแบบอินเตอร์เฟซ	2.5w
7	เขียนโปรแกรมโดยใช้โปรแกรม C ++ Builder	16w
8	ทดสอบและแก้ไขโปรแกรม	9.2w
9	สรุปโครงงานปัญหาพิเศษ	3.8w
10	จัดทำเอกสารประกอบโครงงานปัญหาพิเศษ	2w



Project: Project_3D.MPP
Date: Tue 12/5/98

Task

บทที่ 2
การแปลงภาพเรขาคณิต 2 มิติ

บทนำ

การแปลงภาพเรขาคณิต (Geometric transformations) หมายถึง ภาพที่วาดขึ้นมานั้น สามารถที่จะนำมาเปลี่ยนแปลง แก้ไข ตำแหน่งหรือขนาดได้ ซึ่งการแปลงที่จะกล่าวถึงในหัวข้อใหญ่ ๆ ได้แก่ การย้ายภาพ (Translation) , การหมุนภาพ (Rotation) , การย่อ - ขยายภาพ (Scaling) , การบิดภาพ (Shearing)

- **การแทนจุดในสองมิติ**

แทนด้วยเมตริกซ์ $\begin{bmatrix} x & y \end{bmatrix}$ และสมาชิก x และ y ในเมตริกซ์นั้น คือคู่ลำดับของจุดบนรูปใน 2 มิติ

- **การแปลงรูปในสองมิติ**

การแปลงภาพใน 2 มิติ จะกระทำได้โดยการคูณเมตริกซ์ใด ๆ เข้าไปในสมการ

$$[A]\ [T] = [B]$$

[T] คือ เมตริกซ์ที่ใช้คูณเพื่อทำการแปลงเมตริกซ์

- **การแปลงจุด**

แทนจุดใด ๆ ด้วย $\begin{bmatrix} x & y \end{bmatrix}$ เราจะได้

$$[X]\ [T] = \begin{bmatrix} x & y \end{bmatrix}\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$= \begin{bmatrix} (ax+cy) & (bx+dy) \end{bmatrix}$$

$$= \begin{bmatrix} x^{*} & y^{*} \end{bmatrix}$$

– ในกรณีที่ $a = d = 1$ และ $c = b = 0$ จะได้

$$[X]\ [T] = \begin{bmatrix} x & y \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x & y \end{bmatrix}$$

$$= \begin{bmatrix} x^{*} & y^{*} \end{bmatrix}$$

จะเห็นได้ว่า ไม่มีการเปลี่ยนตำแหน่งของจุด

– ในกรณีที่ $d = 1, b = c = 0$ จะได้

$$[X]\,[T] = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} ax & y \end{bmatrix}$$

$$= \begin{bmatrix} x^* & y^* \end{bmatrix}$$

จะเห็นได้ว่า ค่า x จะเปลี่ยน ค่า y คงที่

– ในกรณีที่ $b = c = 0$ จะได้

$$[X]\,[T] = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix}$$

$$= \begin{bmatrix} ax & dy \end{bmatrix}$$

$$= \begin{bmatrix} x^* & y^* \end{bmatrix}$$

◊ ถ้า $a \neq d$ การย่อ - ขยาย จะผิดเพี้ยนไปจากเดิม แต่ถ้า $a = d > 1$ ภาพที่ถูกขยายแล้ว จะถูกเคลื่อนย้ายห่างออกไปจากจุดประจำที่ แต่ถ้า $a = d < 1$ ภาพที่ถูกย่อแล้ว จะถูกเคลื่อนย้ายเข้ามาใกล้กับจุดประจำที่มากขึ้น

– กรณีที่ $b = c = 0, d = 1$ และ $a = -1$ แล้วจะได้

$$[X]\,[T] = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -x & y \end{bmatrix}$$

$$= \begin{bmatrix} x^* & y^* \end{bmatrix}$$

ภาพที่ปรากฏออกมาจะสะท้อนกับแกน y

– ถ้า $b = c = 0, a = 1$ และ $d = -1$ ภาพที่ปรากฏออกมาจะสะท้อนกับแกน x

– ถ้า $b = c = 0, a = d < 0$ ภาพที่ปรากฏออกมาจะสะท้อนกับจุดกำเนิด

– ในกรณีที่ $a = d = 0$ แล้วจะได้

$$[X]\,[T] = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x & (bx+y) \end{bmatrix}$$

$$= \begin{bmatrix} x^* & y^* \end{bmatrix}$$

จากสมการจะเห็นได้ว่า ค่า x ไม่เปลี่ยน แต่ค่า y ขึ้นอยู่กับสมการเส้นตรงที่ผ่านจุดกำเนิด ในทางเดียวกัน เมื่อ $a = d = 1$, $b = 0$ ค่า y จะไม่เปลี่ยน แต่ค่า x ขึ้นอยู่กับสมการเส้นตรงที่ผ่านจุดกำเนิดเช่นกัน

พิจารณาว่าถ้าต้องการย้ายตำแหน่งของจุด (0 , 0) จะได้

$$\begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix} = \begin{bmatrix} x^* & y^* \end{bmatrix}$$

จะเห็นได้ว่า ไม่สามารถย้ายตำแหน่งของจุด (0 , 0) ได้ ดังนั้นจะต้องเปลี่ยนเป็น Homogeneous Coordinate

- **การแปลงรูปของเส้นตรง**

สามารถแปลงรูปของเส้นตรงได้โดยทำการคูณด้วยเมตริกซ์

$$[T] = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$ โดยที่ a, b, c, d เป็นจำนวนใด ๆ

จะได้ว่า ให้ $[X] = [x \quad y]\,[X]$

$$[T] = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$= \begin{bmatrix} (ax+cy) & (bx+dy) \end{bmatrix}$$

- **การแปลงรูปของจุดกึ่งกลางของเส้นตรง**

เมื่อให้ $[A] = [x_1 \quad y_1]$ $[B] = [x_2 \quad y_2]$ และ $[T] = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ จะได้

$$\begin{bmatrix} A \\ B \end{bmatrix} [T] = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$= \begin{bmatrix} ax_1 + cy_1 & bx_1 + dy_1 \\ ax_2 + cy_2 & bx_2 + dy_2 \end{bmatrix} = \begin{bmatrix} A^* \\ B^* \end{bmatrix}$$

จะได้ว่าจุดปลายของเส้นตรง A^*B^* คือ

$$[A^*] = [ax_1 + cy_1 \quad bx_1 + dy_1]$$

$$[B^*] = [ax_2 + cy_2 \quad bx_2 + dy_2]$$

การหาจุดกึ่งกลางของเส้นตรง A^*B^* ได้ดังนี้

$$[x_m^* \quad y_m^*] = \begin{bmatrix} \dfrac{x_1^* + x_2^*}{2} & \dfrac{y_1^* + y_2^*}{2} \end{bmatrix}$$
$$= \begin{bmatrix} \dfrac{a(x_1 + x_2)}{2} + \dfrac{c(y_1 + y_2)}{2} & \dfrac{b(x_1 + x_2)}{2} + \dfrac{d(y_1 + y_2)}{2} \end{bmatrix}$$

พิจารณาที่เส้นตรง AB เดิม มีจุดกึ่งกลาง คือ

$$[x_m \quad y_m] = \begin{bmatrix} \dfrac{x_1 + x_2}{2} & \dfrac{y_1 + y_2}{2} \end{bmatrix}$$

แล้วนำเมตริกซ์ [T] มาคูณกับเมตริกซ์จุดกึ่งกลางของเส้นตรง AB จะได้จุดกึ่งกลางของเส้นตรงที่ย้ายไป คือ

$$[x_m \quad y_m]\,[T] = \begin{bmatrix} \dfrac{a(x_1 + x_2)}{2} + \dfrac{c(y_1 + y_2)}{2} & \dfrac{b(x_1 + x_2)}{2} + \dfrac{d(y_1 + y_2)}{2} \end{bmatrix}$$

ตัวอย่าง พิจารณาเส้นตรง AB ที่จุด A อยู่ที่จุด (0,1) และ B อยู่ที่จุด (2,3) นั่นคือ

$$[A] = [0 \quad 1] \qquad\qquad [B] = [2 \quad 3]$$

เมตริกซ์ที่ใช้สำหรับคูณเพื่อทำการแปลงเมตริกซ์เดิม คือ

$$[T] = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}$$

เมื่อทำการแปลงจะได้เส้นตรง A^*B^* โดยที่

$$\begin{bmatrix} A^* \\ B^* \end{bmatrix} = \begin{bmatrix} A \\ B \end{bmatrix} [T] = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 3 & 1 \\ 11 & 7 \end{bmatrix}$$

จุดกึ่งกลางของ A^*B^* คือ

$$\begin{bmatrix} x_m^* & y_m^* \end{bmatrix} = \begin{bmatrix} \frac{3+11}{2} & \frac{1+7}{2} \end{bmatrix}$$

$$= \begin{bmatrix} 7 & 4 \end{bmatrix}$$

จุดกึ่งกลางของเส้น AB เดิม คือ

$$\begin{bmatrix} x_m & y_m \end{bmatrix} = \begin{bmatrix} \frac{0+2}{2} & \frac{1+3}{2} \end{bmatrix}$$

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

จะได้จุดกึ่งกลางใหม่ที่ได้จากการแปลง คือ

$$\begin{bmatrix} x_m^* & y_m^* \end{bmatrix} = \begin{bmatrix} x_m & y_m \end{bmatrix} [T] = \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 7 & 4 \end{bmatrix}$$

ซึ่งจะเห็นว่าได้ผลลัพธ์เท่ากัน

- **การแปลงรูปของเส้นขนาน**

ให้ $[A] = \begin{bmatrix} x_1 & y_1 \end{bmatrix}$ และ $[B] = \begin{bmatrix} x_2 & y_2 \end{bmatrix}$ พิจารณาเส้นตรงที่เกิดจาก $[A]$ และ $[B]$ ซึ่งขนานกับ $[E]$ และ $[F]$ ดังนั้นความชันของทั้งสองเส้นจะเท่ากับ

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

ขั้นต่อไป ทำการแปลงจุดปลายของเส้น AB โดยใช้เมตริกซ์ $[T]$ ขนาด 2 x 2

$$\begin{bmatrix} A \\ B \end{bmatrix} [T] = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$=\begin{bmatrix} ax_1+cy_1 & bx_1+dy_1 \\ ax_2+cy_2 & bx_2+dy_2 \end{bmatrix}$$

$$=\begin{bmatrix} x_1^* & y_1^* \\ x_2^* & y_2^* \end{bmatrix}$$

$$=\begin{bmatrix} A^* \\ B^* \end{bmatrix}$$

หาความชันของ A^*B^* จะได้

$$m^* = \frac{(bx_2+dy_2)-(bx_1+dy_1)}{(ax_2+cy_2)-(ax_1+cy_1)} = \frac{b(x_2-x_1)+d(y_2-y_1)}{a(x_2-x_1)+c(y_2-y_1)}$$

หรือ $$m^* = \frac{b+d\dfrac{(y_2-y_1)}{(x_2-x_1)}}{a+c\dfrac{(y_2-y_1)}{(x_2-x_1)_1}} = \frac{b+dm}{a+cm}$$

จะเห็นได้ว่า ความชันของ m^* ขึ้นอยู่กับ x_1, x_2, y_1 และ y_2 แต่เพราะว่า m,a,b,c,d มีค่าเท่าเดิมสำหรับเส้น EF และ AB ดังนั้น สามารถสรุปได้ว่า เมื่อทำการแปลงเส้นคู่ขนานแล้วนั้นก็จะยังคงขนานกันอยู่

- **การหมุนภาพ**

การหมุนภาพเป็นการแปลงอีกแบบหนึ่ง สำหรับการหมุนภาพนี้จะต้องกำหนดว่าจุดใดเป็นจุดหมุนเสมอ หลังจากที่ภาพถูกหมุนไปแล้ว ระยะห่างระหว่างจุดหมุนกับภาพจะยังคงมีค่าเท่าเดิม รูปร่าง ลักษณะของภาพก็ยังคงเดิม แต่ภาพจะมีการจัดวางที่ต่างไปจากเดิมอันเนื่องมาจากการหมุนนั่นเอง การหมุนภาพนี้อาจจะหมุนทีละหลาย ๆ ภาพก็ได้ จะหมุนแบบทวนเข็มนาฬิกาหรือตามเข็มนาฬิกาก็ได้ และจุดหมุนที่ใช้ก็อาจจะอยู่ภายในภาพหรืออยู่ภายนอกภาพก็ได้

สำหรับเมตริกซ์ที่จะใช้คูณเพื่อทำการหมุนโดยที่จุดหมุนคือจุดกำเนิด ในกรณีต่าง ๆ

$$[T] = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$ เมื่อต้องการหมุนเป็นมุม 90 °

$$[T] = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$ เมื่อต้องการหมุนเป็นมุม 180°

$$[T] = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \qquad \text{เมื่อต้องการหมุนเป็นมุม } 270^\circ$$

$$[T] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \text{เมื่อต้องการหมุนเป็นมุม } 360^\circ \text{ หรือ} 0^\circ$$

สำหรับการหมุน ที่มุม θ ใด ๆ

$$[T] = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

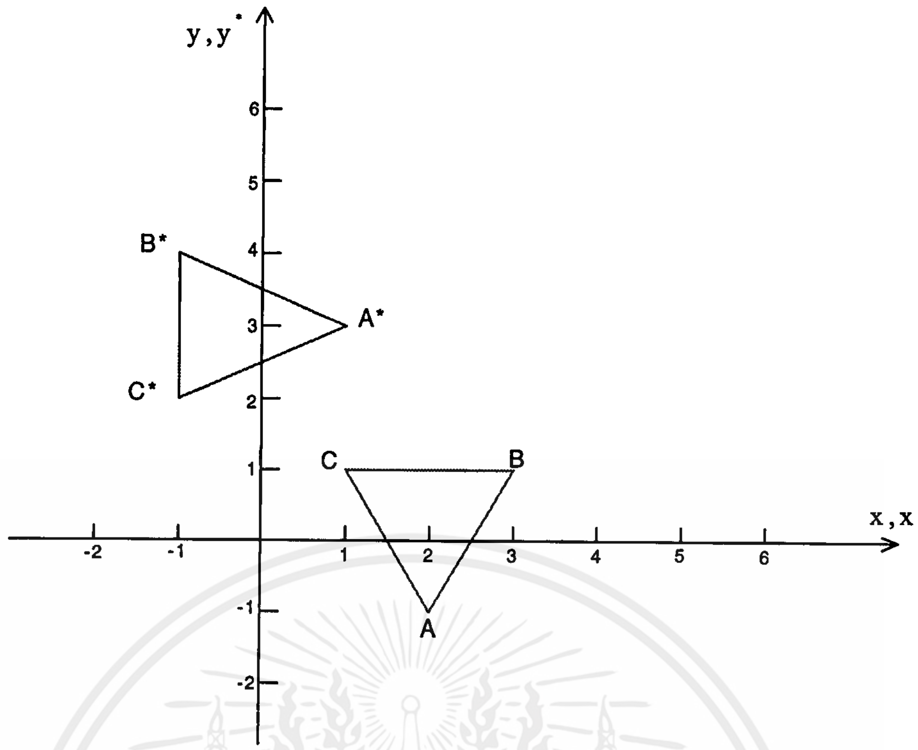
<u>ตัวอย่าง</u> ถ้าต้องการหมุนสามเหลี่ยม ABC เป็นมุม 90° ในทิศทวนเข็มนาฬิกากับจุดกำเนิด ดังนั้น เมตริกซ์ที่จะใช้คูณเพื่อทำการแปลง คือ

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

ถ้าเราใช้เมตริกซ์ 3 x 2 แทน Coordinate ของจุดยอดของรูปสามเหลี่ยม เราจะได้ว่า

$$\begin{bmatrix} 3 & -1 \\ 4 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ -1 & 4 \\ -1 & 2 \end{bmatrix}$$

เมื่อหมุนแล้ว จะได้สามเหลี่ยม $A^*B^*C^*$ ดังรูป



รูปที่ 2.1

- **การสะท้อน**

ในความจริงแล้วการสะท้อนของภาพก็เหมือนกับการหมุนภาพเป็นมุม 180° สำหรับการสะท้อนกับแกน $y = 0$ หรือ แกน x นั้นจะใช้เมตริกซ์

$$[T] = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$ คูณเพื่อทำการแปลง

และสำหรับแกน $x = 0$ หรือ แกน y นั้น จะใช้เมตริกซ์

$$[T] = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$ คูณเพื่อทำการแปลง

และสำหรับแกนที่เส้น $y = x$

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

และสำหรับเส้นที่ $y = -x$

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

<u>ตัวอย่าง</u> ต้องการภาพสะท้อนของสามเหลี่ยม ABC กับแกน X แล้วหาภาพสะท้อนกับเส้นตรง Y = - X

ขั้นแรกหาภาพที่สะท้อนกับแกนX ให้เมตริกซ์ของพิกัดจุดของภาพที่เกิดจากการสะท้อนของสามเหลี่ยม ABC กับแกน X คือ เมตริกซ์ $[X^*]$ ซึ่งหาได้จาก

$$[X^*] = [X]\,[T_1] = \begin{bmatrix}4 & 1\\5 & 2\\4 & 3\end{bmatrix}\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix} = \begin{bmatrix}4 & -1\\5 & -2\\4 & -3\end{bmatrix}$$

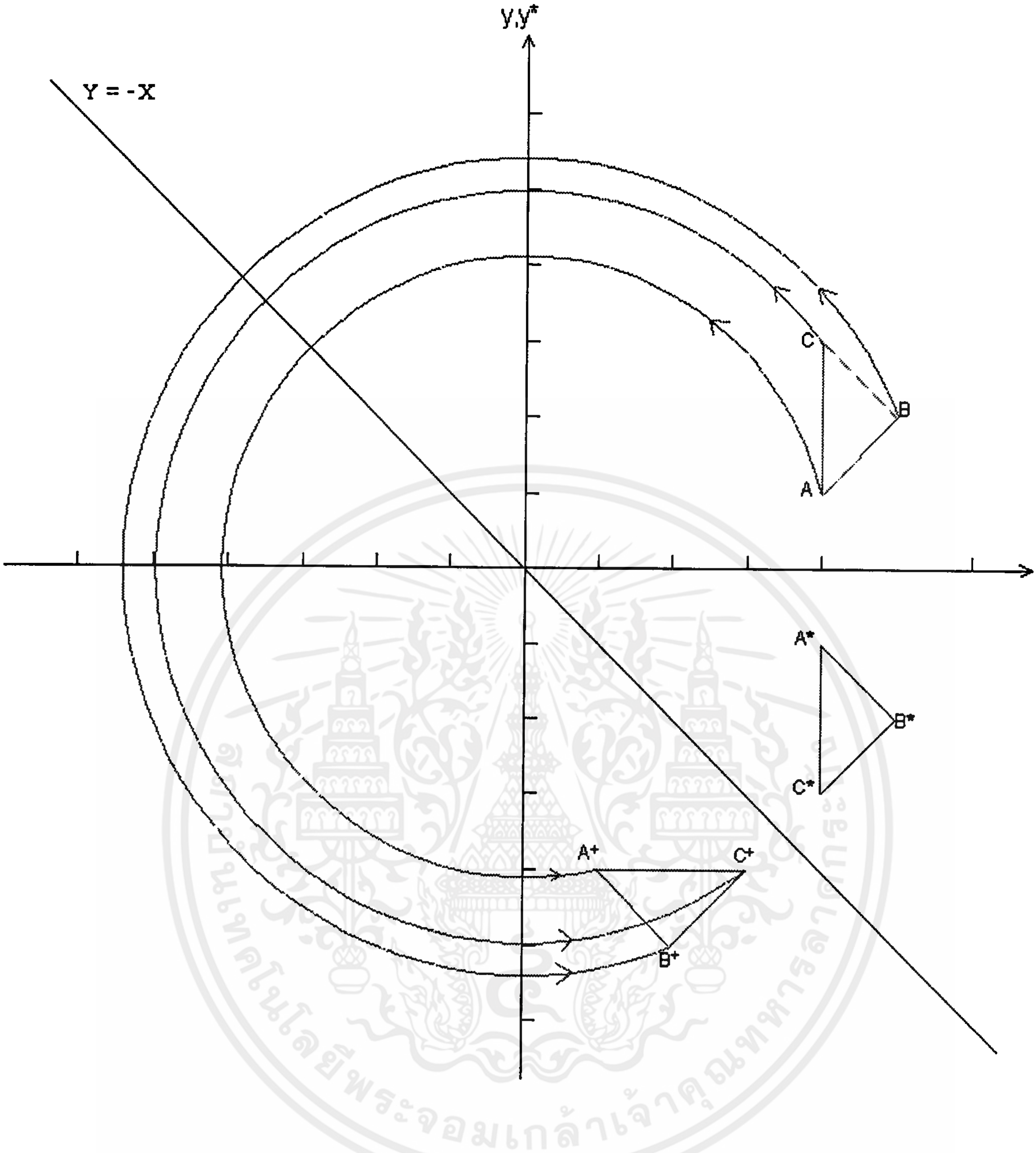
ขั้นตอนที่สอง หาภาพสะท้อนของ $[X^*]$ กับเส้น y = - x ให้เมตริกซ์ของพิกัดจุดของภาพที่เกิดจากการสะท้อน คือ เมตริกซ์ $[X^+]$ ซึ่งหาได้จาก

$$[X^+] = [X^*]\,[T_2] = \begin{bmatrix}4 & -1\\5 & -2\\4 & -3\end{bmatrix}\begin{bmatrix}0 & -1\\-1 & 0\end{bmatrix} = \begin{bmatrix}1 & -4\\2 & -5\\3 & -4\end{bmatrix}$$

หรืออีกทางหนึ่ง โดยการหมุนภาพสามเหลี่ยม ABC เป็นมุม $\theta = 270^\circ$

จะได้ผลตามที่ต้องการเช่นกัน

$$[X^+] = [X]\,[T_3] = \begin{bmatrix}4 & 1\\5 & 2\\4 & 3\end{bmatrix}\begin{bmatrix}0 & -1\\1 & 0\end{bmatrix} = \begin{bmatrix}1 & -4\\2 & -5\\3 & -4\end{bmatrix}$$



รูปที่ 2.2

- **การย่อขยายภาพ**

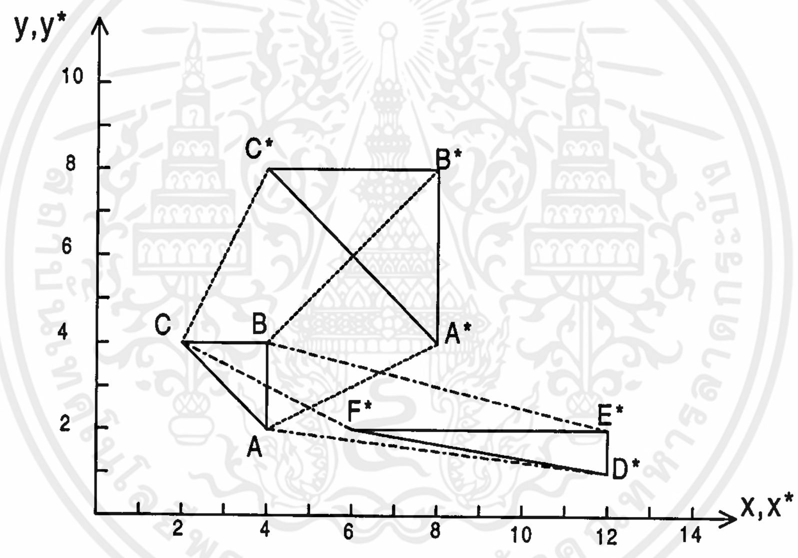
เราสามารถเปลี่ยนขนาดของภาพได้โดยการเปลี่ยนขนาดหน้าต่างแสดงภาพ หรือเปลี่ยนขนาดของช่องแสดงภาพ เทคนิคนี้ไม่สามารถใช้ได้ในบางกรณี เช่น ถ้าเราต้องการเปลี่ยนขนาดของภาพภาพหนึ่งในหน้าต่างเท่านั้น ถ้าเราขยายหรือย่อขนาดของหน้าต่าง ภาพทั้งหมดในหน้าต่างก็จะขยายหรือย่อไปด้วย ในบทนี้จะกล่าวถึงวิธีการเปลี่ยนขนาดของภาพเฉพาะภาพใดภาพหนึ่ง พิจารณาจาก

$$[X]\,[T] \quad = \quad \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

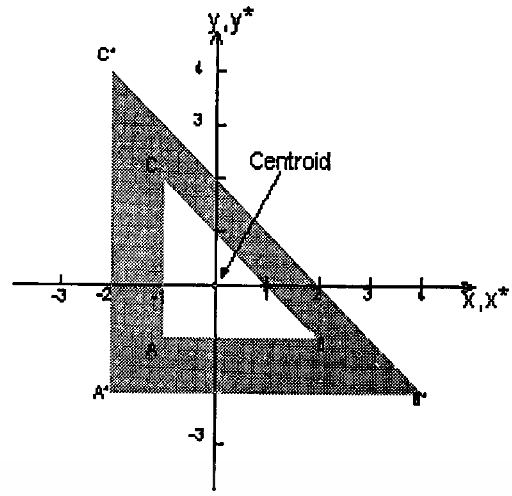
ในกรณีที่ a = d , b = c = 0 เป็นรูปแบบมาตรฐานของการย่อขยายภาพ แล้วจะได้รูปของการย่อขยาย จากสามเหลี่ยม ABC เป็นสามเหลี่ยม $A^*B^*C^*$ ดังรูป แต่ถ้า $a \neq d$, b = c = 0 แล้วจะทำให้รูปแบบการย่อขยายผิดพลาดไป คือ จะได้รูปของการย่อขยายจากสามเหลี่ยม ABC เป็นสามเหลี่ยม $D^*E^*F^*$ ดังรูป

ในกรณีที่ a = d > 1 ภาพที่ถูกขยายแล้วจะถูกเคลื่อนย้ายห่างออกไปจากจุดประจำที่ แต่ถ้ากรณีที่ a = d < 1 ภาพที่ถูกย่อแล้วจะถูกเคลื่อนย้ายเข้ามาใกล้กับจุดประจำที่มากขึ้น ยกตัวอย่างเช่น

$$[X^*] = [X]\,[T] = \begin{bmatrix} 4 & 2 \\ 4 & 4 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 2 & 12 \\ 1 & 12 \end{bmatrix}$$



รูปที่ 2.3



รูปที่ 2.4

- **การแปลงภาพหลาย ๆ แบบผสมกัน**

การแปลงภาพแบบต่าง ๆ ที่กล่าวมาแล้วข้างต้น สามารถนำมาผสมกันกลายเป็นการแปลงภาพแบบใหม่ได้ เช่น ทำการหมุนภาพพร้อม ๆ กับการย้ายภาพ สมมุติว่าเราทำให้เกิดเป็นภาพเคลื่อนไหวได้ จะได้ภาพนั้นหมุนไปพร้อม ๆ กับเคลื่อนที่ไปด้วยนั่นเอง และเมื่อทำการพิจารณาการหมุนและการสะท้อนของ $[X] = [x \quad y]$ โดยหมุนเป็นมุม 90° และตามด้วยการสะท้อนกับเส้น $y = -x$ จะเขียนได้ว่า

$$[X'] = [X][T_1] = [x \quad y]\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = [-y \quad -x]$$

และได้ว่า
$$[X^*] = [X'][T_2] = [-y \quad x]\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} = [-x \quad y]$$

ในทางตรงกันข้าม ถ้าเราสะท้อนแล้วตามด้วยการหมุน จะได้ผลเป็น

$$[X'] = [X][T_2] = [x \quad y]\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} = [-y \quad -x]$$

และ $$[X^*] = [X'][T_1] = [-y \quad -x]\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = [x \quad -y]$$

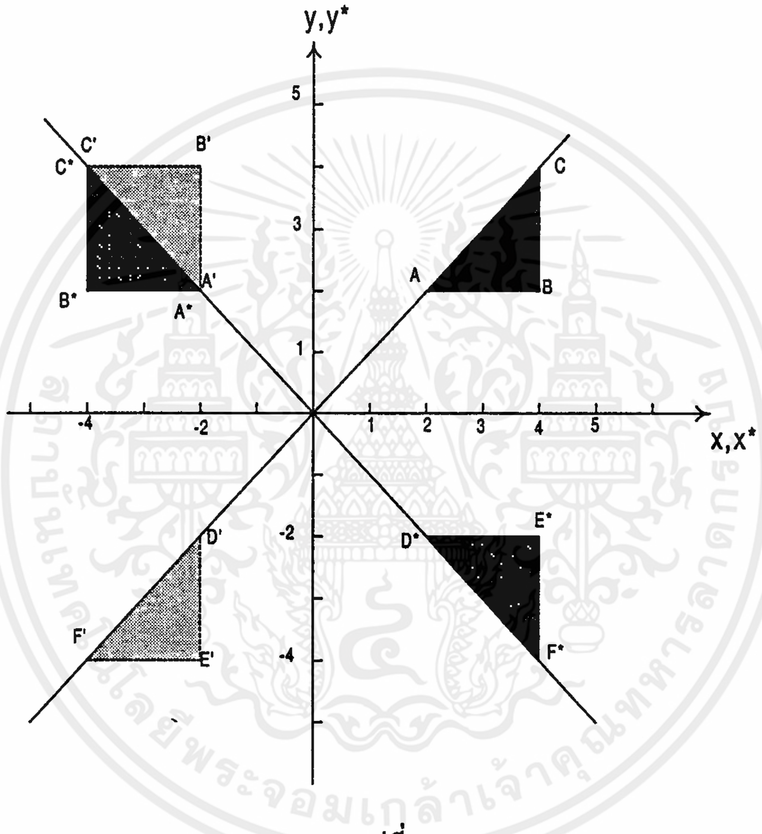
<u>**ตัวอย่าง**</u> พิจารณารูปสามเหลี่ยม ABC มีการหมุนเป็นมุม 90° (ได้รูปสามเหลี่ยม A′B′C′) และสะท้อนกับเส้น $y = -x$ (ได้รูปสามเหลี่ยม $A^*B^*C^*$) ดังนั้น

$$[T_1] = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \text{และ} \quad [T_2] = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

ให้ $[T_3] = [T_1][T_2]$ ดังนั้น การแปลงรูปสามเหลี่ยม ABC สามารถเขียนได้เป็น

$$[X^*] = [X][T_1][T_2] = [X][T_3]$$

จะได้ $\begin{bmatrix} 2 & 2 \\ 4 & 2 \\ 4 & 4 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 4 & 2 \\ 4 & 4 \end{bmatrix}\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -2 & 2 \\ -4 & 2 \\ -4 & 4 \end{bmatrix}$



รูปที่ 2.5

พิจารณาการทำสะท้อนก่อนแล้วทำการหมุน 90° จากรูปจะเห็นว่าถ้าทำการสะท้อนรูปสามเหลี่ยม ABC กับเส้น y = - x จะได้รูปสามเหลี่ยม D′E′F′ และเมื่อทำการหมุนไป 90° เราจะได้รูปสามเหลี่ยม $D^*E^*F^*$ ซึ่งไม่เหมือนรูปสามเหลี่ยม $A^*B^*C^*$ ซึ่งได้จากการหมุนและสะท้อน ตามลำดับ จะได้

$$[X^*] = [X][T_2][T_1] = [X][T_4]$$

$$\begin{bmatrix}2&2\\4&2\\4&4\end{bmatrix}\begin{bmatrix}0&-1\\-1&0\end{bmatrix}\begin{bmatrix}0&1\\-1&0\end{bmatrix}=\begin{bmatrix}2&2\\4&2\\4&4\end{bmatrix}\begin{bmatrix}1&0\\0&-1\end{bmatrix}=\begin{bmatrix}2&-2\\4&-2\\4&-4\end{bmatrix}$$

ดังนั้นจะสรุปได้ว่า ลำดับการทำสะท้อน และหมุนมีผลต่อการแปลงรูป

- **การย้ายภาพ และ Homogeneous Coordinate**

การเขียนคู่ลำดับ ปกตินั้นเรามีปัญหาในการย้ายจุด $\begin{bmatrix}0 & 0\end{bmatrix}$ เพราะว่าเมื่อคูณด้วยเมตริกซ์ใด ๆ แล้ว ค่าของตำแหน่งใหม่ที่ออกมาก็คงยังเป็นศูนย์เสมอ ดังนั้น เราแก้ปัญหาโดยวิธีการเขียนพิกัดให้อยู่ในรูปของ Homogeneous Coordinate เช่น $\begin{bmatrix}x & y\end{bmatrix}$ เขียนเป็น Homogeneous Coordinate คือ $\begin{bmatrix}x & y & 1\end{bmatrix}$ หรือ Homogeneous Coordinate เขียนได้เป็น $\begin{bmatrix}hx & hy & h\end{bmatrix}$ ดังนั้นจะเห็นได้ว่า สามารถเขียนให้อยู่ในรูปของ Homogeneous Coordinate ได้หลายรูปแบบ เช่น $\begin{bmatrix}6 & 4 & 2\end{bmatrix}$, $\begin{bmatrix}12 & 8 & 4\end{bmatrix}$, $\begin{bmatrix}3 & 2 & 1\end{bmatrix}$ ทั้งหมดล้วนแทนจุด (3,2) และเมตริกซ์ที่ใช้แปลงข้อมูลจะอยู่ในรูป $[T] = \begin{bmatrix}a&b&0\\c&d&0\\m&n&1\end{bmatrix}$

จะได้
$$\begin{bmatrix}x^* & y^* & 1\end{bmatrix} = \begin{bmatrix}x & y & 1\end{bmatrix}\begin{bmatrix}1&0&0\\0&1&0\\m&n&1\end{bmatrix}$$
$$= \begin{bmatrix}x+m & y+n & 1\end{bmatrix}$$

จะเห็นได้ว่า สามารถทำการแปลง (0,0) ได้แล้ว

- **การหมุนรอบจุดอิสระใด ๆ**

จากการศึกษาที่ผ่านมา การหมุนจะเกิดจากการหมุนรอบจุดกำเนิด การเปลี่ยนพิกัดให้อยู่ในรูปของHomogeneous Coordinate นั้น ทำให้สามารถที่จะหมุนรอบจุดใด ๆ ที่ไม่ใช่จุดกำเนิดได้ ดังนั้นถ้าเราต้องการหมุนตำแหน่ง $\begin{bmatrix}x & y & 1\end{bmatrix}$ รอบจุด m, n เป็นมุมใด ๆ สามารถเขียนได้เป็น

$$\begin{bmatrix}x^* & y^* & 1\end{bmatrix}=\begin{bmatrix}x & y & 1\end{bmatrix}\begin{bmatrix}1&0&0\\0&1&0\\-m&-n&1\end{bmatrix}\begin{bmatrix}\cos\theta&\sin\theta&0\\-\sin\theta&\cos\theta&0\\0&0&1\end{bmatrix}\begin{bmatrix}1&0&0\\0&1&0\\m&n&1\end{bmatrix}$$

และเขียนได้เป็น

$$\begin{bmatrix} x^* & y^* & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix} = \begin{bmatrix} x+m & y+n & 1 \end{bmatrix}$$

<u>ตัวอย่าง</u> สมมุติให้จุดศูนย์กลางวัตถุอยู่ที่ $\begin{bmatrix} 4 & 3 \end{bmatrix}$ และต้องการที่จะหมุนวัตถุเป็นมุม 90° ทวนเข็มนาฬิกา ต้องใช้เมตริกซ์ $\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ เพราะที่จุดกำเนิดไม่ใช่จุดศูนย์กลางของวัตถุ จึงจำเป็นต้องย้ายรูปให้จุดศูนย์กลางของวัตถุเป็นจุดกำเนิดก่อน โดยใช้เมตริกซ์คูณเพื่อทำการย้าย เมตริกซ์ที่ใช้คือ $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -4 & -3 & 1 \end{bmatrix}$ และเมื่อทำการหมุนแล้วก็ทำการย้ายตำแหน่งกลับไปที่เดิม ซึ่งก็คือต้องคูณด้วย "เมตริกซ์ผกผัน" ทั้งหมดสามารถเขียนได้เป็น

$$\begin{bmatrix} x^* & y^* & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -4 & -3 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 4 & 3 & 1 \end{bmatrix}$$

และเมื่อทำการคูณแล้ว จะเขียนได้เป็น

$$\begin{bmatrix} x^* & y^* & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 7 & -1 & 1 \end{bmatrix}$$

- **การสะท้อนกับเส้นตรงอิสระใด ๆ**

ในข้างต้น การสะท้อนที่ศึกษานั้นจะสะท้อนกับเส้นที่ผ่านจุดกำเนิด ในส่วนนี้จะทำการศึกษาการสะท้อนวัตถุกับเส้นตรงใด ๆ ที่อาจไม่ผ่านจุดกำเนิดบ้าง มีขั้นตอนดังนี้

— ย้ายเส้นและวัตถุ ดังนั้น เส้นจะผ่านจุดกำเนิด

— หมุนแกน หรือ เส้น และวัตถุรอบจุดกำเนิด จนกระทั่งเส้นหรือแกนทับกันสนิทกับแกน Coordinate แกนใดแกนหนึ่ง

— สะท้อนวัตถุกับแกน Coordinate

— คูณด้วยเมตริกซ์ผกผัน กลับไปยังจุดเดิม

— คูณด้วยเมตริกซ์ผกผัน ย้ายตำแหน่งไปยังตำแหน่งเดิมของมัน

สรุป เขียนอยู่ในรูปของผลคูณเมตริกซ์ คือ

$$[T] = [T'][R][R'][R^{-1}][T']^{-1}$$

โดยที่ $[T']$ คือ เมตริกซ์ที่ใช้คูณเพื่อทำการย้ายตำแหน่ง

$[R]$ คือ เมตริกซ์ที่ใช้คูณเพื่อทำการหมุนรอบจุดกำเนิด

$[R']$ คือ เมตริกซ์ที่ใช้คูณเพื่อทำการสะท้อน

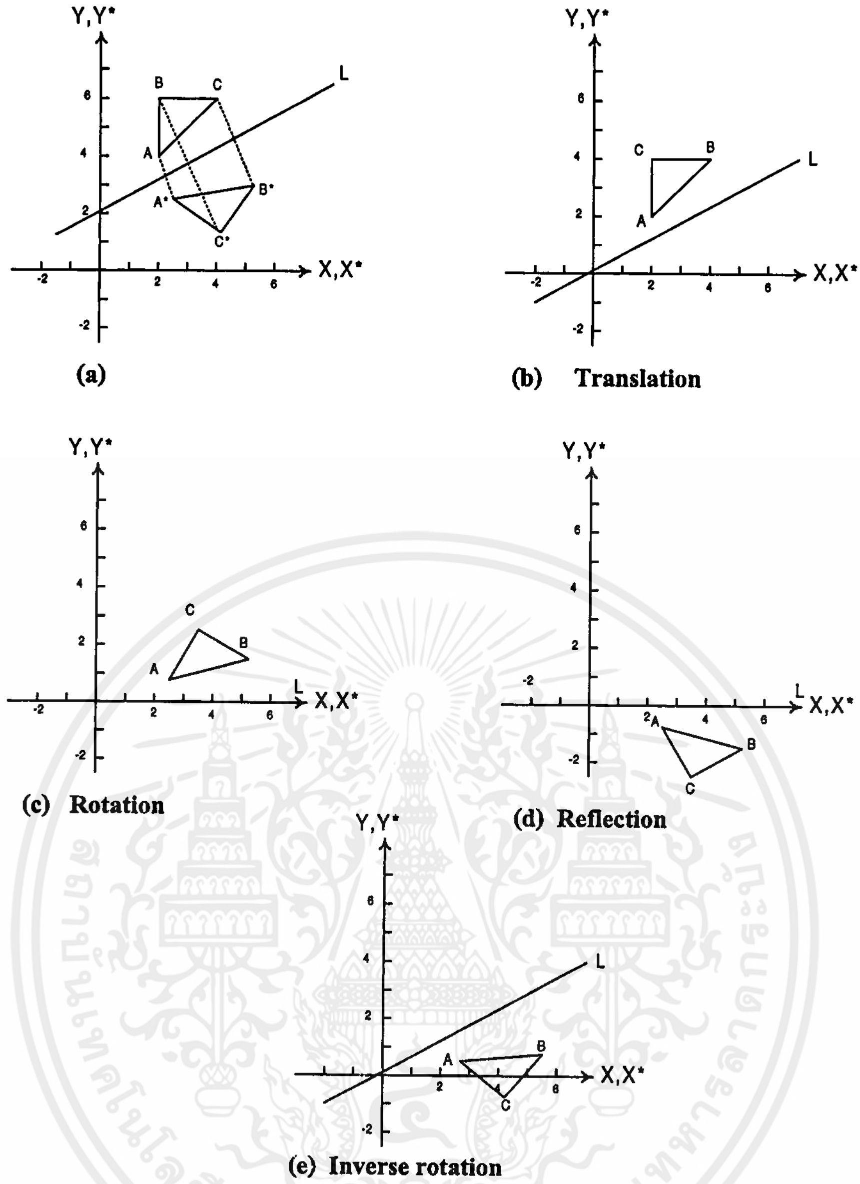
ตัวอย่าง พิจารณาเส้นตรง L และสามเหลี่ยม ABC สมการเส้นตรง คือ $y = \frac{1}{2}(x+4)$ และจุดยอดของวัตถุ คือ $\begin{bmatrix}2 & 4 & 1\end{bmatrix}$, $\begin{bmatrix}4 & 6 & 1\end{bmatrix}$, $\begin{bmatrix}2 & 6 & 1\end{bmatrix}$ เส้นตรง L ผ่านจุด ถ้าแปลงโดยการย้ายมา -2 หน่วยในแกน y และต้องหมุนด้วย $\theta = -26.57°$ รอบจุดกำเนิดเพื่อให้ทับสนิทกับแกน x จากนั้นก็แปลงตำแหน่งของสามเหลี่ยมกลับโดยการหมุนและย้ายตำแหน่งกลับไปที่จุดเดิม ดังแสดงให้เห็นใน รูปที่ 2.6 สามารถเขียนได้เป็น

$$[T] = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -2 & 1\end{bmatrix}\begin{bmatrix}\frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0\\ \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1\end{bmatrix} \mathbf{x} \begin{bmatrix}\frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & 0\\ \frac{-1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 2 & 1\end{bmatrix}$$

$$[T] = \begin{bmatrix}\frac{3}{5} & \frac{4}{5} & 0\\ \frac{4}{5} & \frac{-3}{5} & 0\\ \frac{-8}{5} & \frac{16}{5} & 1\end{bmatrix}$$

และจะแปลงตำแหน่งเป็นสามเหลี่ยม $A^*B^*C^*$ คือ

$$\begin{bmatrix}2 & 4 & 1\\ 4 & 6 & 1\\ 2 & 6 & 1\end{bmatrix}\begin{bmatrix}\frac{3}{5} & \frac{4}{5} & 0\\ \frac{4}{5} & \frac{-3}{5} & 0\\ \frac{-8}{5} & \frac{16}{5} & 1\end{bmatrix} = \begin{bmatrix}\frac{14}{5} & \frac{12}{5} & 1\\ \frac{28}{5} & \frac{14}{5} & 1\\ \frac{22}{5} & \frac{6}{5} & 1\end{bmatrix}$$


Y,Y*
X,X*
L
A
B
C
A*
B*
C*
(a)
(b) Translation
(c) Rotation
(d) Reflection
(e) Inverse rotation

รูปที่ 2.6

การแปลงภาพเรขาคณิต 3 มิติ

เนื่องจากเราต้องการใช้การคำนวณทางเมตริกซ์มาใช้ในการคำนวณการแปลงภาพ เราจึงต้องแปลงพิกัดปกติของเราให้เป็น " Homogeneous Coordinate " ก่อน ดังนั้นถ้าเป็นพิกัด 3 มิติ จะต้องเขียนในรูป vector 1 x 4 เช่น $\begin{bmatrix} x & y & z \end{bmatrix}$ จะต้องทำการเขียนให้อยู่ในรูปของ $\begin{bmatrix} x & y & z & 1 \end{bmatrix}$ สำหรับจุดใหม่ที่แปลงแล้วคือ $\begin{bmatrix} x' & y' & z' \end{bmatrix}$ เขียนให้อยู่ในรูปของ vector จะเป็น $\begin{bmatrix} x' & y' & z' & h \end{bmatrix}$

$$\begin{bmatrix} x' & y' & z' & h \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} [T]$$

ซึ่ง [T] คือ เมตริกซ์ที่ใช้ทำการแปลง และถ้าเราจะทำการแปลงจาก " Homogeneous Coordinate " กลับมาเป็น " Physical Coordinate" เราสามารถทำได้โดย

$$\begin{bmatrix} x^* & y^* & z^* & 1 \end{bmatrix} = \begin{bmatrix} \frac{x'}{h} & \frac{y'}{h} & \frac{z'}{h} & 1 \end{bmatrix}$$

สำหรับเมตริกซ์ที่ใช้ในการคูณ เพื่อทำการแปลงภาพใน 3 มิตินั้น มีรูปแบบเป็น

$$[T] = \begin{bmatrix} a & b & c & p \\ d & e & f & q \\ g & i & j & r \\ l & m & n & s \end{bmatrix}$$

<u>Three Dimensional Scaling</u> (การย่อ - ขยายภาพ)

เมตริกซ์ที่ใช้ในการเปลี่ยน Scale ของวัตถุใด ๆ นั้น จะมีรูปแบบของเมตริกซ์เป็น

$$[X]\,[T] = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & e & 0 & 0 \\ 0 & 0 & j & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} ax & ey & jz & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x^* & y^* & z^* & 1 \end{bmatrix}$$

รวมทั้งหมดแล้ว ในการแปลง Scale ของภาพนั้น เมตริกซ์ที่นำมาคูณเพื่อทำการ แปลงจะอยู่ในรูปของ

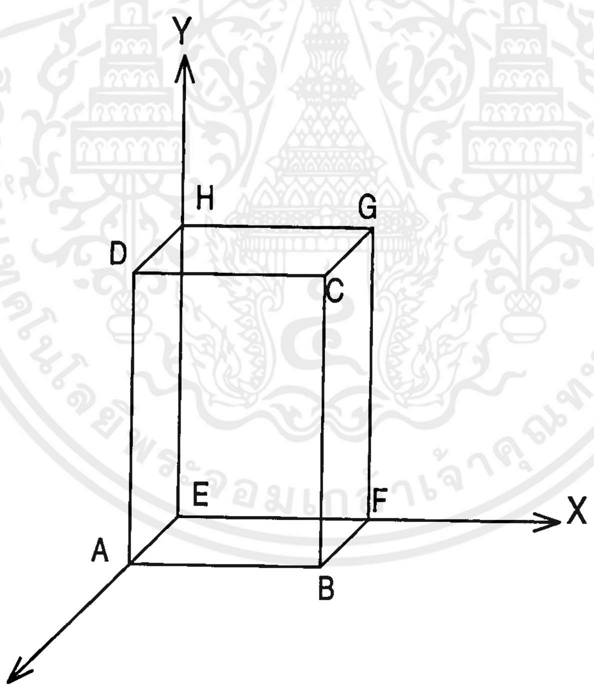
$$[X]\ [T] = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & s \end{bmatrix}$$

$$= \begin{bmatrix} x' & y' & z' & s \end{bmatrix}$$

และเปลี่ยนเป็น" Physical Coordinate " โดย

$$[X]\ [T] = \begin{bmatrix} x^* & y^* & z^* & 1 \end{bmatrix} = \begin{bmatrix} \frac{x'}{s} & \frac{y'}{s} & \frac{z'}{s} & 1 \end{bmatrix}$$

หรือ $$[T] = \begin{bmatrix} \frac{1}{s} & 0 & 0 & 0 \\ 0 & \frac{1}{s} & 0 & 0 \\ 0 & 0 & \frac{1}{s} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ เพื่อทำให้ค่า $h = 1$



รูปที่ 2.7

พิจารณาจากรูปที่ 2.7 เขียน Homogeneous Coordinate ได้เป็น

$$[X] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 2 & 0 & 1 & 1 \\ 2 & 3 & 1 & 1 \\ 0 & 3 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 3 & 0 & 1 \\ 0 & 3 & 0 & 1 \end{bmatrix}$$

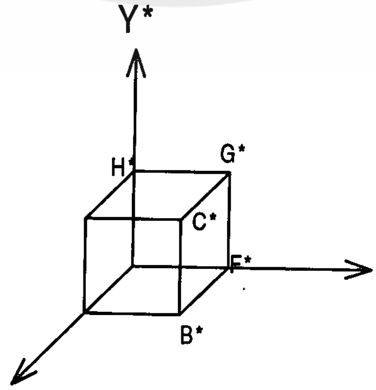
ทำการย่อ - ขยายวัตถุ โดยการคูณด้วยเมตริกซ์

$$[T] = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

และผลลัพธ์ของ Homogeneous Coordinate คือ

$$[X^*] = [X]\,[T] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 2 & 0 & 1 & 1 \\ 2 & 3 & 1 & 1 \\ 0 & 3 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 3 & 0 & 1 \\ 0 & 3 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

ผลที่ได้ดังจะเห็นในรูป 2.8



รูปที่ 2.8

แต่ถ้าต้องการทำให้รูปมีขนาดเพิ่มขึ้นเป็น 2 เท่า เมตริกซ์ [T] ที่นำมาคูณจะอยู่ในรูป

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}$$

ผลของHomogeneous Coordinate ที่เปลี่ยนไป แสดงให้เห็นในด้านล่าง

$$[X'] = [X^*][T] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0.5 \\ 1 & 0 & 1 & 0.5 \\ 1 & 1 & 1 & 0.5 \\ 0 & 1 & 1 & 0.5 \\ 0 & 0 & 0 & 0.5 \\ 1 & 0 & 0 & 0.5 \\ 1 & 1 & 0 & 0.5 \\ 0 & 1 & 0 & 0.5 \end{bmatrix}$$

จะเห็นว่า $[X']$ ที่หาได้อยู่ในรูปของ Homogeneous Coordinate แต่ไม่สามารถแปลงในรูป " Physical Coordinate" ได้ จึงต้องแปลง $[X']$ ก่อน โดยการคูณ 2 แล้วจะได้

$$[X^*] = \begin{bmatrix} 0 & 0 & 2 & 1 \\ 2 & 0 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 0 & 2 & 2 & 1 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix}$$

จะสังเกตุได้ว่า ถ้าต้องการทำให้รูปมีขนาดเพิ่มขึ้นเป็น 2 เท่า จะคูณด้วย $[T]$ แล้วหา $[X']$ ได้จากนั้นก็คูณ 2 ก็จะได้ $[X^*]$ ดังนั้น หากเราต้องการทำให้มีขนาดเพิ่มขึ้นเป็น n เท่า จะคูณด้วย

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{n} \end{bmatrix}$$

แล้วหา $[X']$ ได้จากนั้นก็คูณ n ก็จะได้ $[X^*]$ ซึ่งมีขนาดเป็น n เท่าของ $[X]$ เดิม

<u>การบิดภาพ</u>

เมตริกซ์ที่ใช้คูณเพื่อทำการบิดภาพนั้นจะมีรูปแบบของเมตริกซ์ดังนี้

$$[X]\,[T] = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & b & c & 0 \\ d & 1 & f & 0 \\ g & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x+yd+gz & bx+y+iz & cx+fy+z & 1 \end{bmatrix}$$

<u>การหมุน</u>

การหมุนภาพจะต้องกำหนดว่าจุดใดเป็นจุดหมุนเสมอ หรือเมตริกซ์ที่ใช้ทำการคูณเพื่อหมุนภาพนั้นจะอยู่ในรูปแบบดังนี้

หมุนตามแกน x $$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

หมุนตามแกน z $$[T] = \begin{bmatrix} \cos\omega & \sin\omega & 0 & 0 \\ -\sin\omega & \cos\omega & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

หมุนตามแกน y $$[T] = \begin{bmatrix} \cos\phi & 0 & -\sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

<u>ตัวอย่าง</u> พิจารณารูปที่ 2 .10 เมตริกซ์ของตำแหน่ง คือ

$$[X] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 3 & 0 & 1 & 1 \\ 3 & 2 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 1 \\ 3 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix}$$

ต่อไป จะทำการหมุนรอบแกน x โดย $\theta = 90°$ จะได้เมตริกซ์สำหรับ การคูณเพื่อแปลง คือ

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

โดยเมตริกซ์ที่ทำการแปลงแล้ว คือ

$$[X^*] = [X]\,[T] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 3 & 0 & 1 & 1 \\ 3 & 2 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 1 \\ 3 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 3 & 1 & 0 & 1 \\ 3 & 1 & -2 & 1 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 1 \\ 3 & 0 & -2 & 1 \\ 0 & 0 & -2 & 1 \end{bmatrix}$$

อีกกรณีหนึ่งคือการหมุนรอบแกน y โดยที่ $\phi = 90°$ จะได้เมตริกซ์สำหรับการคูณเพื่อแปลง คือ

$$[T'] = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

โดยเมตริกซ์ที่ทำการแปลงแล้ว คือ

$$\left[X^{*'}\right]=[X]\ [T']=\begin{bmatrix}0&0&1&1\\3&0&1&1\\3&2&1&1\\0&2&1&1\\0&0&0&1\\3&0&0&1\\3&2&0&1\\0&2&0&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&0&-1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}=\begin{bmatrix}1&0&0&1\\1&0&-3&1\\1&2&-3&1\\1&2&0&1\\0&0&0&1\\0&0&-3&1\\0&2&-3&1\\0&2&0&1\end{bmatrix}$$

ต่อไปลองพิจารณาการหมุนตามแกน x แล้วต่อด้วยการหมุนตามแกน y จะได้เมตริกซ์ใช้คูณ เพื่อทำการหมุนดังต่อไปนี้

$$[T]=\begin{bmatrix}1&0&0&0\\0&\cos\theta&\sin\theta&0\\0&-\sin\theta&\cos\theta&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}\cos\theta&0&-\sin\theta&0\\0&1&0&0\\\sin\theta&0&\cos\theta&0\\0&0&0&1\end{bmatrix}$$

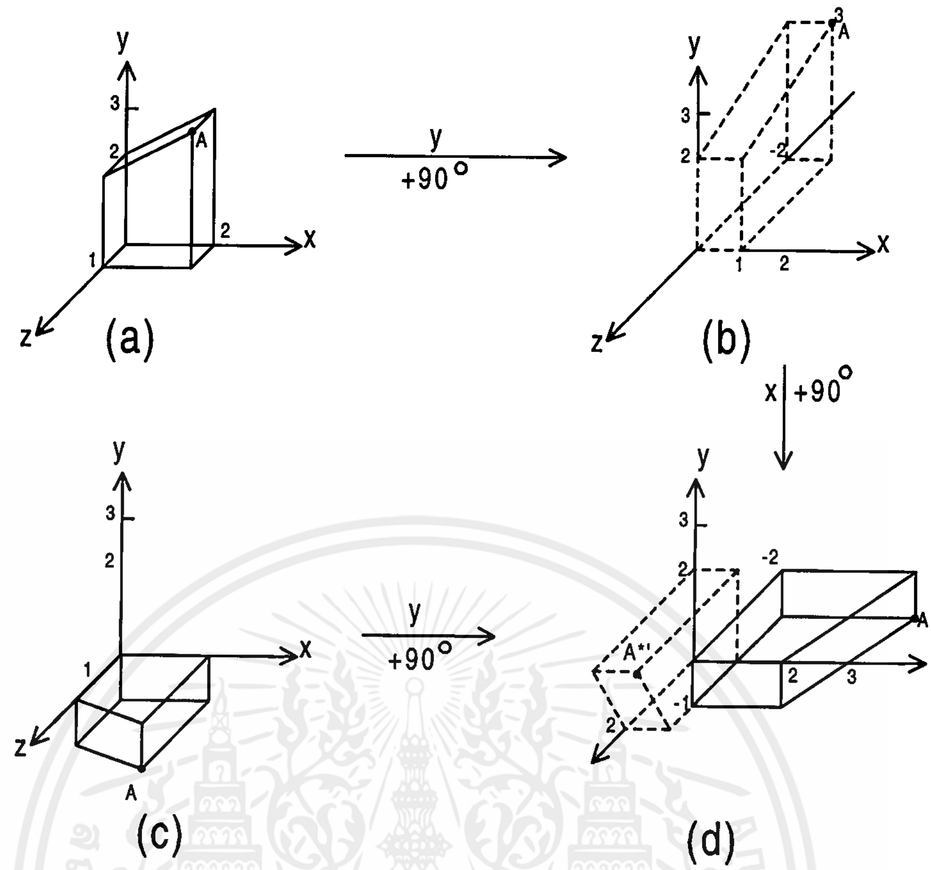
$$=\begin{bmatrix}\cos\theta&0&-\sin\theta&0\\\sin^2\theta&\cos\theta&\cos\theta\sin\theta&0\\\cos\theta\sin\theta&-\sin\theta&\cos^2\theta&0\\0&0&0&1\end{bmatrix}$$

และในทางกลับกัน ถ้าเราหมุนตามแกน y ก่อน แล้วค่อยหมุนตามแกน x เมตริกซ์ที่ใช้หมุนจะเป็น

$$[T]=\begin{bmatrix}\cos\theta&0&-\sin\theta&0\\0&1&0&0\\\sin\theta&0&\cos\theta&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&\cos\theta&\sin\theta&0\\0&-\sin\theta&\cos\theta&0\\0&0&0&1\end{bmatrix}$$

$$=\begin{bmatrix}\cos\theta&\sin^2\theta&-\cos\theta\sin\theta&0\\0&\cos\theta&\sin\theta&0\\\sin\theta&-\cos\theta\sin\theta&\cos^2\theta&0\\0&0&0&1\end{bmatrix}$$

ถ้าเราลองทำการเปรียบเทียบการหมุนทั้ง 2 ลักษณะ จะเห็นว่าเมื่อหมุนเสร็จแล้วภาพ 3 มิติที่ปรากฏจะต่างกัน



รูปที่ 2.9

การสะท้อนของภาพ 3 มิติ

ในการสะท้อนของรูป 3 มิตินั้นจะเสมือนกับการหมุนภาพต่อไปเป็นเมตริกซ์ที่จะใช้ในการคูณเพื่อให้เกิดภาพสะท้อน

ถ้าสะท้อนตาม plane x - y $$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

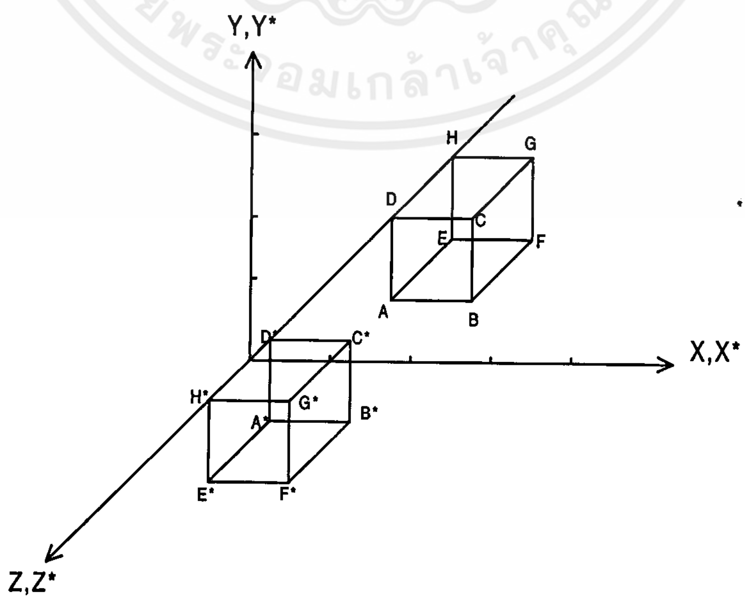
ถ้าสะท้อนตาม plane y - z $$[T] = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ถ้าสะท้อนตาม plane x - z $$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

<u>ตัวอย่าง</u> กล่อง ABCDEFGH แสดงดังรูปที่ 5 เขียนเมตริกซ์แสดงตำแหน่งได้

$$[X] = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 2 & 0 & -1 & 1 \\ 2 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 0 & -2 & 1 \\ 2 & 0 & -2 & 1 \\ 2 & 1 & -2 & 1 \\ 1 & 1 & -2 & 1 \end{bmatrix}$$

$$[X^*] = [X]\,[T] = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 2 & 0 & -1 & 1 \\ 2 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 0 & -2 & 1 \\ 2 & 0 & -2 & 1 \\ 2 & 1 & -2 & 1 \\ 1 & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 2 & 0 & 1 & 1 \\ 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 2 & 1 \\ 2 & 0 & 2 & 1 \\ 2 & 1 & 2 & 1 \\ 1 & 1 & 2 & 1 \end{bmatrix}$$



รูปที่ 2.10

การย้ายภาพ

เมตริกซ์ที่ใช้ในการย้ายภาพ คือ

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{bmatrix}$$

การหมุน " Homogeneous Coordinate " นั้น สามารถเขียนได้เป็น

$$\begin{bmatrix} x' & y' & z' & h \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{bmatrix}$$

จะได้ $\begin{bmatrix} x' & y' & z' & h \end{bmatrix} = \begin{bmatrix} (x+l) & (y+m) & (z+n) & 1 \end{bmatrix}$

หรือในการหมุน " Physical Coordinate" ก็จะสามารถเขียนได้เป็น

$$x^* = x + l$$
$$y^* = y + m$$
$$z^* = z + n$$

การแปลงภาพหลาย ๆ แบบผสมกัน

เราใช้รูปแบบการคูณในเมตริกซ์ ดังนี้

$$[X]\,[T] = [X]\,[T_1]\,[T_2]\,[T_3]\,[T_4]\,\ldots$$

เมื่อ $[T] = [T_1]\,[T_2]\,[T_3]\,[T_4]\,\ldots$

ดังนั้น เราสามารถแปลงภาพหลาย ๆ แบบผสมกันได้ ดังตัวอย่าง เราจะทำการย้ายตำแหน่งของภาพก่อน เสร็จแล้วนำการหมุนตามแกน x และหมุนตามแกน x ตามลำดับ

$$[T] = [T_r]\,[R_x]\,[R_y] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & 0 & -\sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ \sin\phi\sin\theta & \cos\theta & \cos\phi\sin\theta & 0 \\ \sin\phi\cos\theta & -\sin\theta & \cos\phi\cos\theta & 0 \\ l\cos\phi + m\sin\phi\sin\theta + n\sin\phi\cos\theta & m\cos\theta - n\sin\theta & -l\sin\phi + m\cos\phi\sin\theta + n\cos\phi\cos\theta & 1 \end{bmatrix}$$

<u>การหมุนรอบแกนของตัวเอง โดยขนานกับแกน Coordinate</u>

เมตริกซ์ที่ใช้ในการคูณเพื่อให้วัตถุ 3 มิติหมุนรอบแกนของตัวเอง คือ

$$[T] = [T_r]\,[R]\,[T_r]^{-1}$$

จะได้ว่า

$$[X^*] = [X]\,[T_r]\,[R]\,[T_r]^{-1}$$

โดย $[X^*]$ แทนเมตริกซ์ที่เกิดจากการหมุนแล้ว

$[X]$ แทนเมตริกซ์ก่อนการหมุน

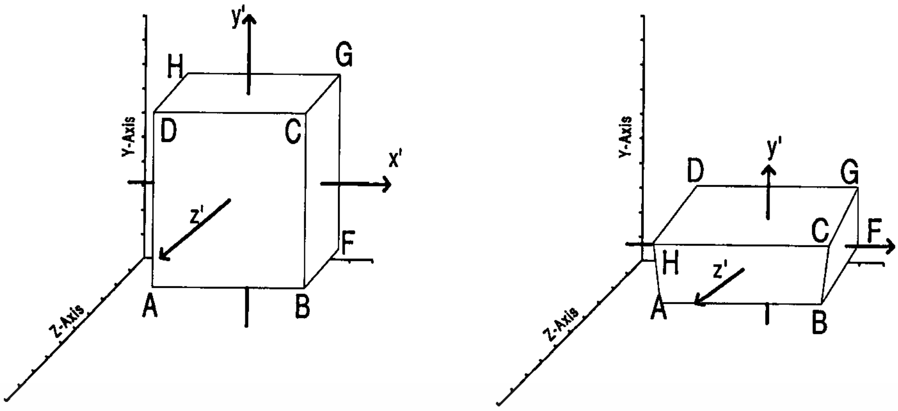
$[T_r]$ แทนเมตริกซ์ที่จะใช้คูณเพื่อทำการย้ายที่

$[R]$ แทนเมตริกซ์ที่จะใช้คูณเพื่อทำการหมุน

$[T_r]^{-1}$ แทนเมตริกซ์ผกผัน ของการหมุน

<u>ตัวอย่าง</u> พิจารณาจากรูปที่ 2 .12 จะได้เมตริกซ์แสดงตำแหน่ง คือ

$$[X] = \begin{array}{c} \begin{bmatrix} 1 & 1 & 2 & 1 \\ 2 & 1 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 2 & 1 & 1 \end{bmatrix} \end{array} \begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array}$$



รูปที่ 2 .11

ให้ทำการหมุน block โดยที่ $\theta = 30^{\circ}$ รอบแกน x′ ซึ่งผ่านจุดศูนย์ถ่วงของ block โดยจุดศูนย์ถ่วงของ block คือ $\begin{bmatrix} x_c & y_c & z_c & 1 \end{bmatrix} = \begin{bmatrix} \frac{3}{2} & \frac{3}{2} & \frac{3}{2} & 1 \end{bmatrix}$ โดยสามารถหาเมตริกซ์แสดงตำแหน่งที่หมุนแล้วโดยสมการต่อไปนี้

$$[X^{*}] = [X]\,[T_r]\,[R]\,[T_r]^{-1}$$

โดยที่

$$[T_r] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -y_c & -z_c & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{-3}{2} & \frac{-3}{2} & 1 \end{bmatrix}$$

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & \theta & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.866 & 0.5 & 0 \\ 0 & -0.5 & 0.866 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

และ

$$[T_r]^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & y_c & z_c & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{3}{2} & \frac{3}{2} & 1 \end{bmatrix}$$

เมื่อคูณเมตริกซ์ $[T_r]$ แล้วแกนของblock จะถูกย้ายตำแหน่งมาทับกับแกน x-coordinate พอดี จากนั้นจะทำการคูณด้วยเมตริกซ์ $[R_x]$ เพื่อทำการหมุนรอบแกน x-coordinate จากนั้นก็ทำการคูณเพื่อย้าย

ตำแหน่งของ block กลับสู่แกนเดิมของ block โดยการคูณด้วยเมตริกซ์ $\left[T_r\right]^{-1}$ ซึ่งแสดงดังสมการต่อไปนี้

$$[T] = \left[T_r\right]\,[R]\,\left[T_r\right]^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & y_c(1-\cos\theta)+z_c\sin\theta & z_c(1-\cos\theta)-y_c\sin\theta & 1 \end{bmatrix}$$

เมตริกซ์การแปลงที่ได้ คือ

$$[X'] = [X]\,[T] = \begin{bmatrix} 1 & 1 & 2 & 1 \\ 2 & 1 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.866 & 0.5 & 0 \\ 0 & -0.5 & 0.866 & 0 \\ 0 & 0.951 & -0.549 & 1 \end{bmatrix}$$

$$= \begin{array}{c} \begin{bmatrix} 1 & 0.817 & 1.683 & 1 \\ 2 & 0.817 & 1.683 & 1 \\ 2 & 1.683 & 2.183 & 1 \\ 1 & 1.683 & 2.183 & 1 \\ 1 & 1.317 & 0.817 & 1 \\ 2 & 1.317 & 0.817 & 1 \\ 2 & 2.183 & 1.317 & 1 \\ 1 & 2.183 & 1.317 & 1 \end{bmatrix} \end{array} \begin{array}{l} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array}$$

ถ้าจะทำการหมุนรอบแกนของตัวเอง ขนานกับแกน y-coordinate และตามด้วยการหมุนขนานกับแกน x-coordinate ทำได้โดยคูณกับเมตริกซ์ที่จะใช้หมุน คือ

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_c & -y_c & -z_c & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & 0 & -\sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_c & y_c & z_c & 1 \end{bmatrix}$$

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

การหมุนรอบแกนของตัวเองโดยอิสระ

เมตริกซ์ที่ใช้ในการคูณเพื่อให้วัตถุ 3 มิติหมุนรอบแกนตัวเองโดยอิสระ คือ

$$[M] = [T]\left[R_x\right]\left[R_y\right]\left[R_\delta\right]\left[R_y\right]^{-1}\left[R_x\right]^{-1}[T]^{-1}$$

เมื่อ $[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_0 & -y_0 & -z_0 & 1 \end{bmatrix}$ $\left[R_x\right] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{C_z}{d} & \frac{C_y}{d} & 0 \\ 0 & \frac{-C_y}{d} & \frac{C_z}{d} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$$\left[R_y\right] = \begin{bmatrix} d & 0 & C_x & 0 \\ 0 & 1 & 0 & 0 \\ -C_x & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[R_\delta\right] = \begin{bmatrix} \cos\delta & \sin\delta & 0 & 0 \\ -\sin\delta & \cos\delta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

โดยเราจะต้องหา direction cosine (C_x, C_y, C_z) ก่อน โดยที่

$$\left[C_x, C_y, C_z\right] = \frac{\left[(x_1 - x_0)(y_1 - y_0)(z_1 - z_0)\right]}{\left[(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2\right]^{\frac{1}{2}}}$$

การสะท้อนบน Arbitrary Plane

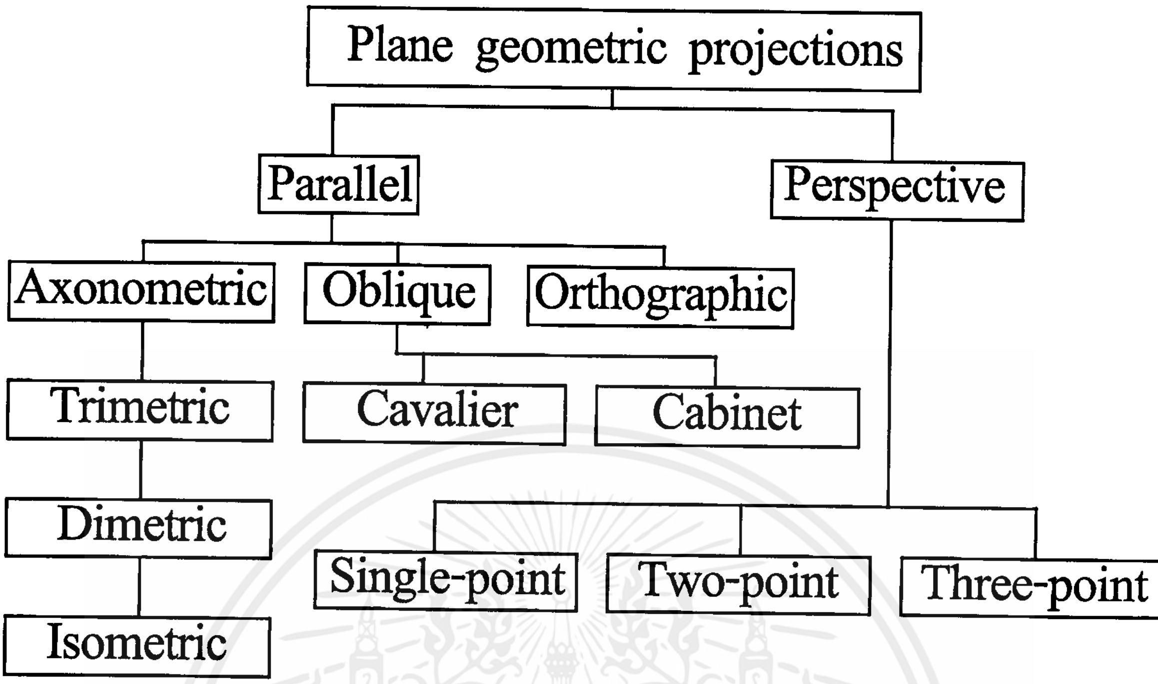
เป็นการสะท้อนกับระนาบอิสระที่เกิดจากแกนอิสระของรูปทรง และเมตริกซ์ที่ใช้ในการคูณเพื่อหมุนภาพนั้น มีรูปแบบดังนี้

$$[M] = [T]\left[R_x\right]\left[R_y\right][Rflt]\left[R_y\right]'\left[R_x\right]'[T]^{-1}$$

โดย $[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_c & -y_c & -z_c & 1 \end{bmatrix}$

และ [Rflt] เป็นเมตริกซ์ที่ใช้ทำการคูณเพื่อสะท้อนวัตถุกับระนาบอิสระ

Affine และ Perspective Geometry



รูปที่ 2.12

รูปที่ 2.12 ด้านบน คือ Hierarchy of plane geometric projections

Orthographic projections

เมตริกซ์สำหรับการหา projection บนระนาบที่ z = 0 คือ

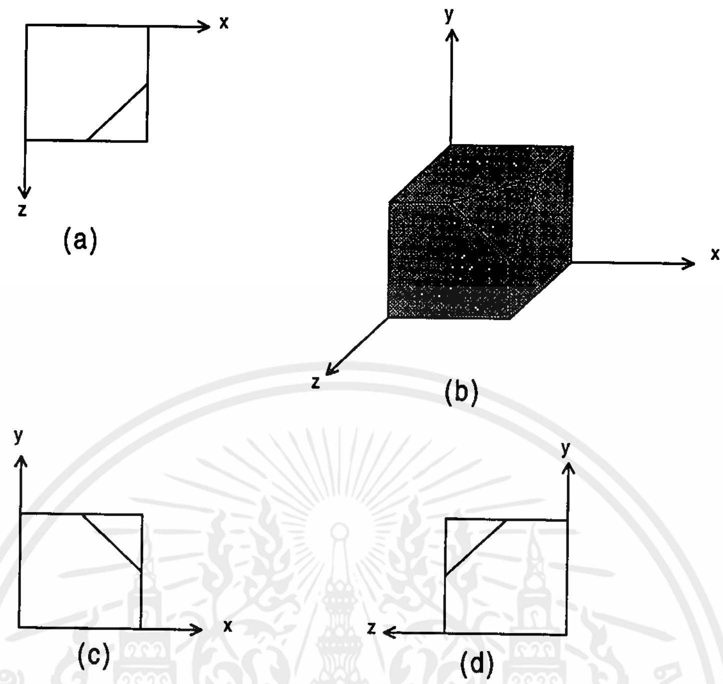
$$[P_z] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

เมื่อ $x = 0$ และ $y = 0$ เมตริกซ์สำหรับการหา projection คือ

$$[P_x] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[P_y] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

เป็นการหา projection จากการมองแค่ 3 มุมมอง สำหรับใน 6 มุมมอง เมื่อเรามองทางด้านหลัง เราก็หา projection บน z = 0 ทั้งด้านข้างซ้าย และ ด้านล่างก็เช่นกัน หา projection บนระนาบ x = 0 และ y = 0 ตามลำดับ



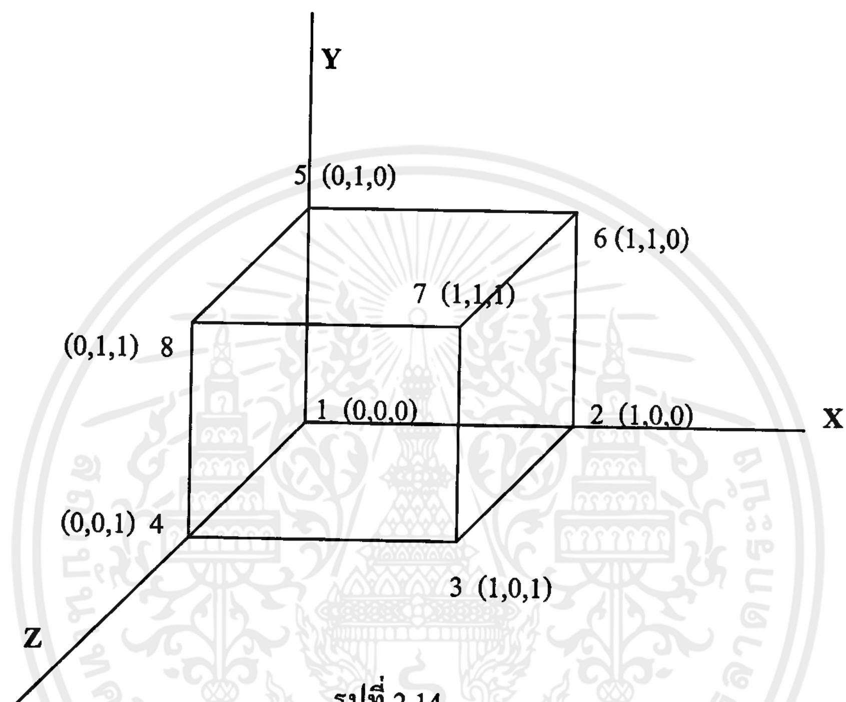
รูปที่ 2 .13

จากรูป (b) จะมี projection บน y = 0 เป็นรูป (a) , z = 0 เป็นรูป (c) , x=0 เป็นรูป (d)

ทฤษฎีพื้นฐานที่ใช้กับคอมพิวเตอร์กราฟฟิก

การออกแบบชิ้นงานต่างๆไม่ว่าจะเป็น เครื่องหัตถกรรม,ยานพาหนะ,เฟอร์นิเจอร์ และอื่นๆ จะต้องมีการกล่าวถึงพื้นผิวของวัตถุ ดังนั้นเราจึงต้องมีความเข้าใจในเรื่องพื้นผิว ซึ่งเป็นส่วนสำคัญในการแทนข้อมูลของวัตถุ ในที่นี้เราจะแทนพื้นผิวของวัตถุเป็นแบบ polygon

<u>ตัวอย่าง</u> การเก็บข้อมูลแบบ polygon



รูปที่ 2.14

ตารางที่ 1 จะเก็บจุดข้อมูลที่อยู่ในสามมิติจริง

Vertex	X	Y	Z
1	0	0	0
2	1	0	0
3	1	0	1
4	0	0	1
5	0	1	0
6	1	1	0
7	1	1	1
8	0	1	1

ตารางที่ 1

P	Vertex			
1	1	2	6	5
2	2	3	7	6
3	3	4	8	7
4	4	1	5	8
5	5	6	7	8
6	1	4	3	2

ตารางที่ 2

ตารางที่ 2 เก็บ polygon ซึ่งในแต่ละ polygon จะประกอบไปด้วยจุดยอดที่บอกตำแหน่งของ polygon ในที่นี้จะเก็บ 4 จุด เพราะเป็น polygon รูปสี่เหลี่ยมดังตารางที่ 2

บทที่ 3

การดำเนินงานและพัฒนาระบบ

3.1 ภาพรวมของระบบ

แผนงานและการพัฒนาระบบ

1 ศึกษาหลักและ Computer Graphic และหลักทางคณิตศาสตร์ที่ต้องนำมาใช้ในการวางแผนงานและพัฒนาระบบ

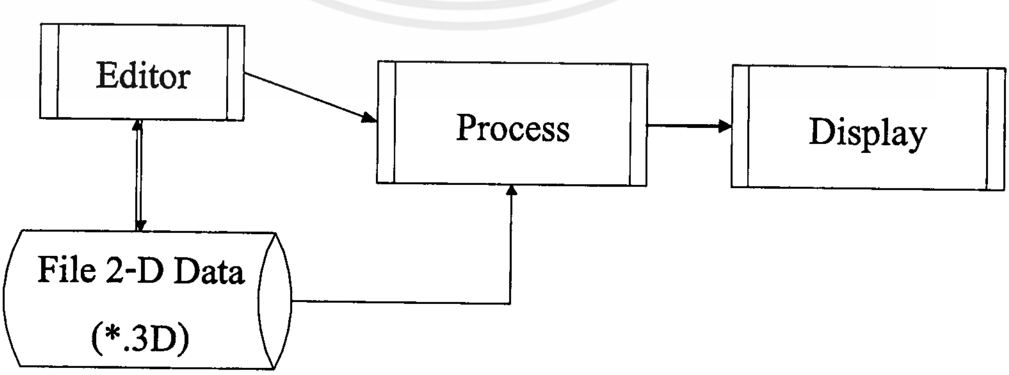
2. ศึกษาระบบและวิธีการใช้งาน Application Development Tools เพื่อให้เหมาะสม และมีความสามารถที่จะใช้พัฒนาระบบได้โดยเลือก C++ Builder

3. ศึกษาระบบ Software ที่เกี่ยวกับ graphic ที่มีในปัจจุบัน เพื่อศึกษาและนำมาประยุกต์ใช้ในการพัฒนาโปรแกรม

4. ทำการพัฒนาระบบงานจริง โดยแบ่งขั้นตอนดังนี้

- ออกแบบ flow chart ของระบบงาน
- กำหนด input output ของแต่ละ flow chart ในระบบ
- ออกแบบ user interface
- ดำเนินงานพัฒนาโปรแกรม
- หาข้อผิดพลาดของโปรแกรมโดยไปให้ผู้ใช้งานทดลองใช้ เพื่อสามารถนำมาปรับปรุงให้ได้ตรงความต้องการมากที่สุด
- จัดทำเอกสารประกอบการใช้งาน
- ติดตั้งระบบให้สามารถทำงานได้ตามวัตถุประสงค์

3.2. System Flow Diagram ของระบบ



รูปที่ 3.1

– ในส่วนการทำงานของ Editorจะรับ input เป็นจุด 2 จุด ซึ่งโปรแกรมจะทำการสร้างเส้นตรงเชื่อมจุด 2 จุดดังกล่าว แล้วจะแสดงภาพของทุกเส้นเชื่อมที่ user ป้อนข้อมูลเข้าไป โดย Editor จะรอรับข้อมูลจาก user จนครบทั้ง 3 view

– โปรแกรมจะทำการเก็บข้อมูลจากส่วนที่แล้วลงไฟล์ 2D และ โปรแกรมจะทำการตรวจเช็คข้อมูลของ user ว่าถูกต้องหรือไม่ด้วย

– นำข้อมูลจากไฟล์ 2D มาผ่าน Process (โดยใช้แนวความคิดจากบทที่ 2 และ 3) เพื่อทำการแปลงข้อมูลจาก 2 มิติเป็นข้อมูล 3 มิติ

– นำข้อมูล 3 มิติจากส่วนที่แล้วเก็บลง หน่วยความจำ เพื่อทำการแสดงออกจอภาพในส่วน Process นี้สามารถทำการหมุนภาพ ย้ายภาพตามความต้องการของผู้ใช้

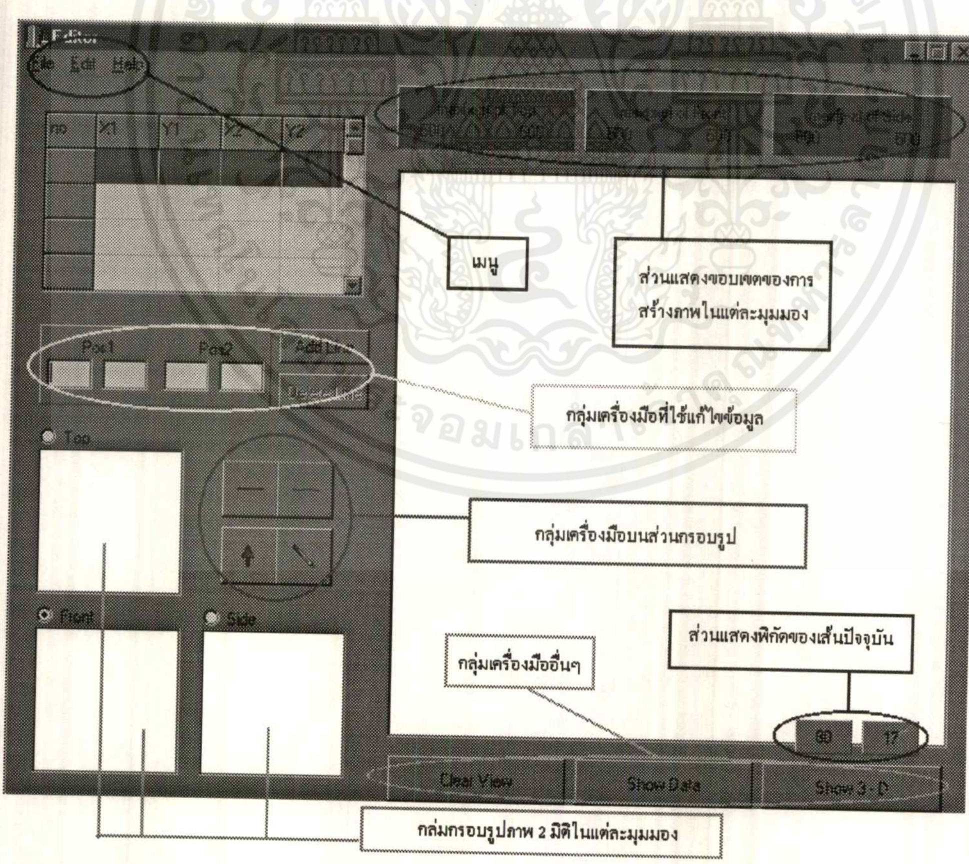
3.3. **<u>อัลกอริทึมของโปรแกรมส่วนติดต่อกับผู้ใช้(User InterFace)</u>**

ส่วนติดต่อกับผู้ใช้ ประกอบด้วย

1 หน้าจอหลัก

2 หน้าจอแสดงรูป 3 มิติ

หน้าจอหลัก ประกอบด้วย



รูปที่ 3.2

ส่วนที่ 1 เมนู

ส่วนที่ 2 ตารางแสดงข้อมูล 2 มิติ

ส่วนที่ 3 กรอบรูปแสดงข้อมูล 2 มิติ

ส่วนที่ 4 กลุ่มเครื่องมือบนส่วนกรอบรูป

ส่วนที่ 5 กลุ่มกรอบรูปภาพ 2 มิติในแต่ละมุมมอง

ส่วนที่ 6 ส่วนแสดงขอบเขตของการสร้างภาพในแต่ละมุมมอง

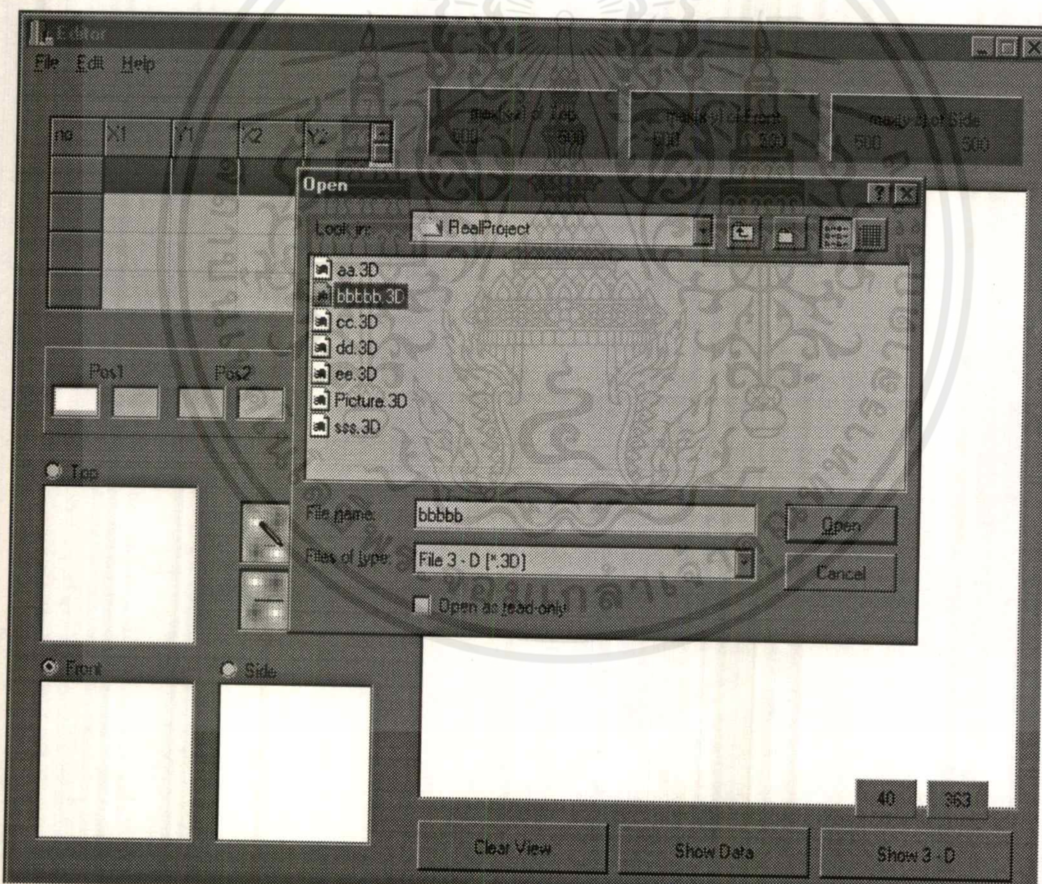
ส่วนที่ 7 ส่วนแสดงพิกัดของเส้นปัจจุบัน

ส่วนที่ 8 กลุ่มเครื่องมือที่ใช้แก้ไขข้อมูล

ส่วนที่ 9 กลุ่มเครื่องมืออื่น ๆ

เมนู

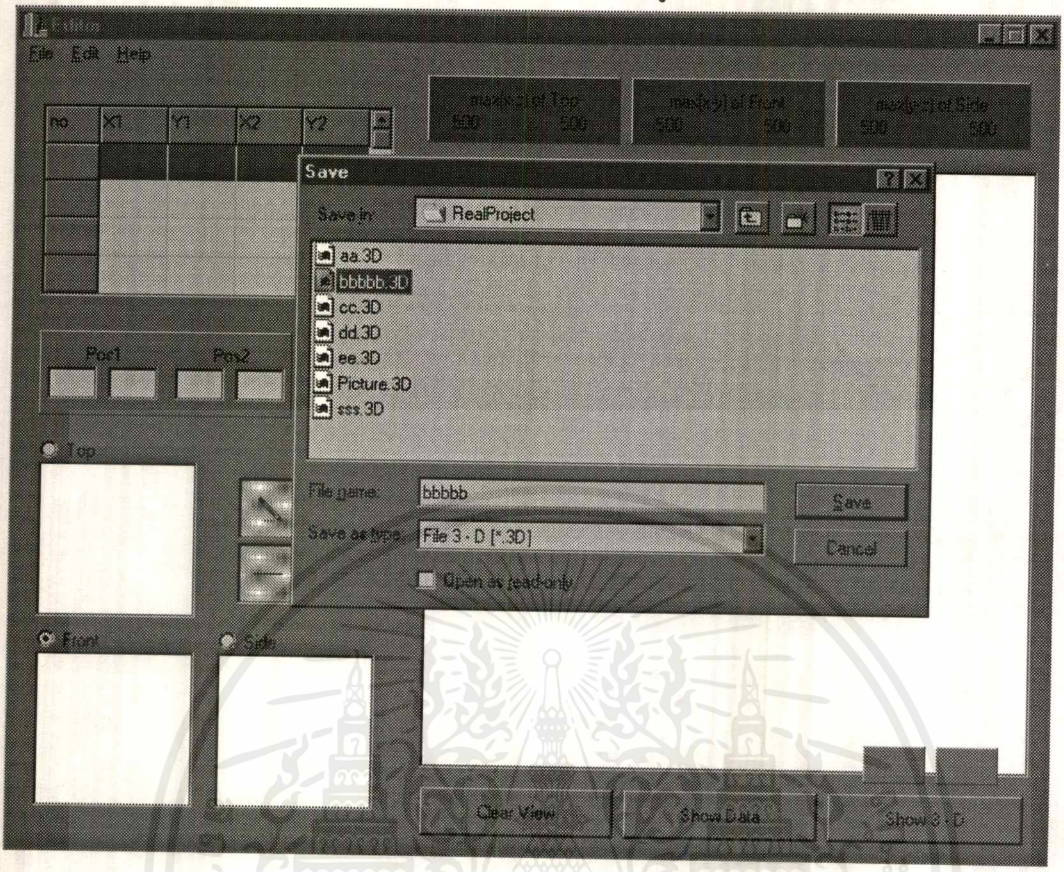
– เมื่อกดที่ File โปรแกรมจะทำการขึ้นทางเลือกต่าง ๆ ได้แก่



รูปที่ 3.3

– เมื่อกดที่ New โปรแกรมจะทำการขึ้นหน้าจอให้ผู้ใช้เริ่มต้นการทำงานใหม่

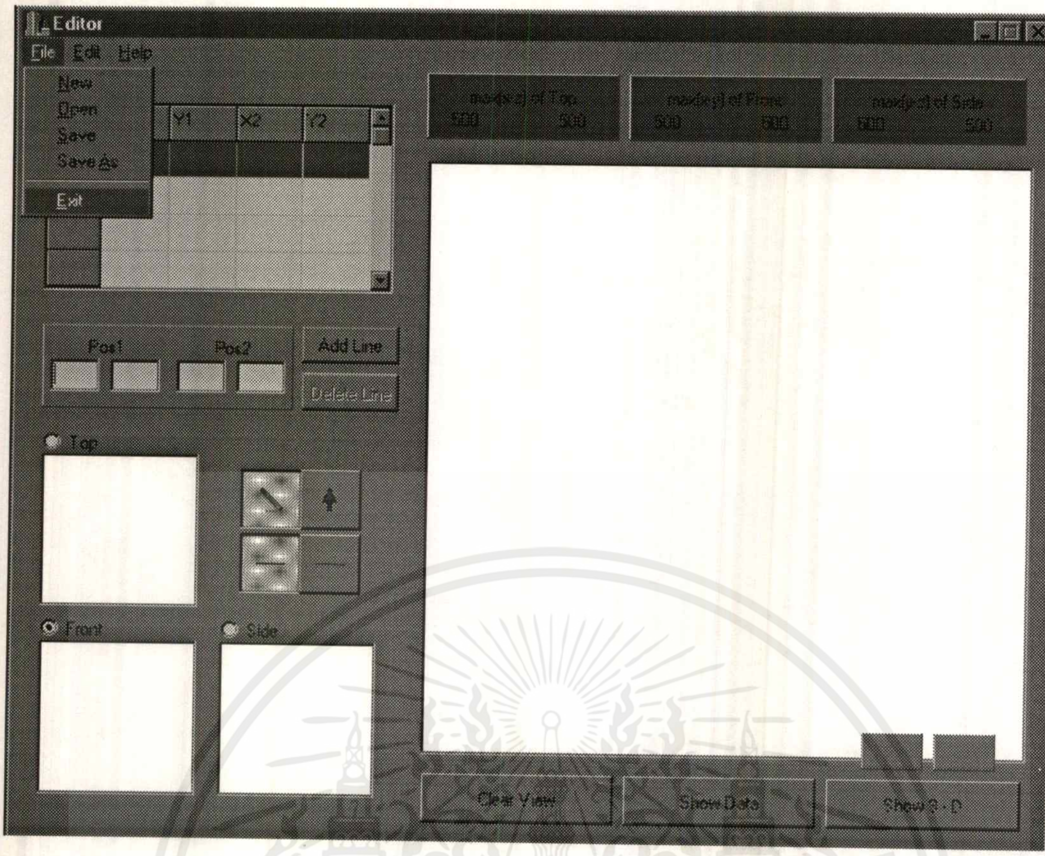
— เมื่อกดที่ **Open** โปรแกรมจะทำการขึ้นหน้าจอให้ผู้ใช้ใส่ชื่อไฟล์ที่ต้องการจะเปิด



รูปที่ 3.4

— เมื่อกดที่ **Save** โปรแกรมจะทำการตรวจเช็คว่า user เคยใส่ชื่อไฟล์หรือยัง ถ้ายังโปรแกรมจะทำการขึ้นหน้าจอเพื่อให้ user ใส่ชื่อไฟล์ แต่ถ้ามีชื่อไฟล์แล้ว(เช่นเปิดไฟล์แล้วแก้ไข) ก็จะทำการ save ทับ

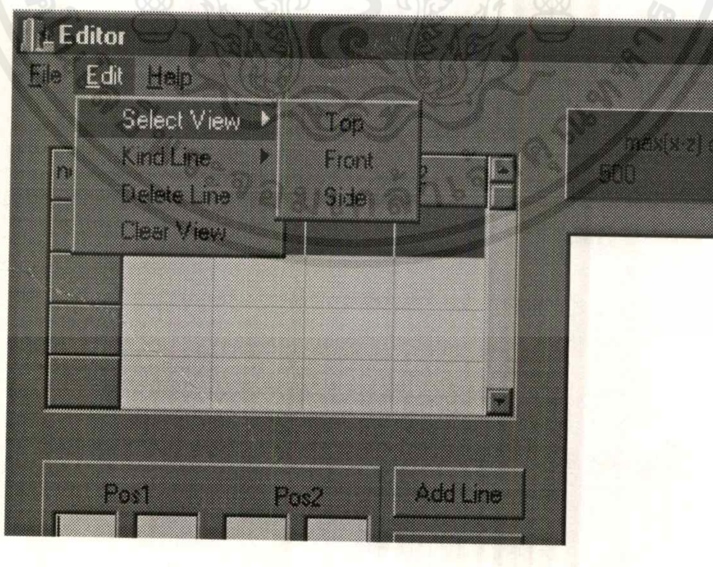
— เมื่อกดที่ **Save As** โปรแกรมจะทำการขึ้นหน้าจอให้ผู้ใช้ใส่ชื่อไฟล์ที่ผู้ใช้ต้องการจะบันทึกข้อมูลใหม่ ถ้ามีข้อมูลภายในไฟล์นั้น โปรแกรมก็จะทำการ save ทับ



รูปที่ 3.5

– เมื่อกดที่ **Exit** โปรแกรมจะจบการทำงาน

– **เมื่อกดที่ Edit โปรแกรมจะทำการขึ้นทางเลือกต่าง ๆ ได้แก่**



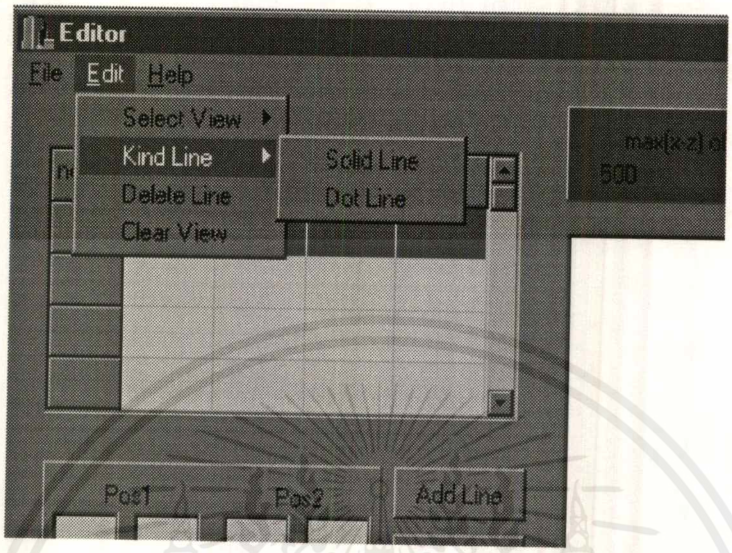
รูปที่ 3.6

– เมื่อเลื่อน Mouse ไปที่ **Select view** จะปรากฏเมนูย่อยที่ประกอบด้วย **Top , Front , Side** ซึ่ง

เมื่อกด Top โปรแกรมจะทำการเปลี่ยนระบบการทำงานไปที่มุมมองด้านบน

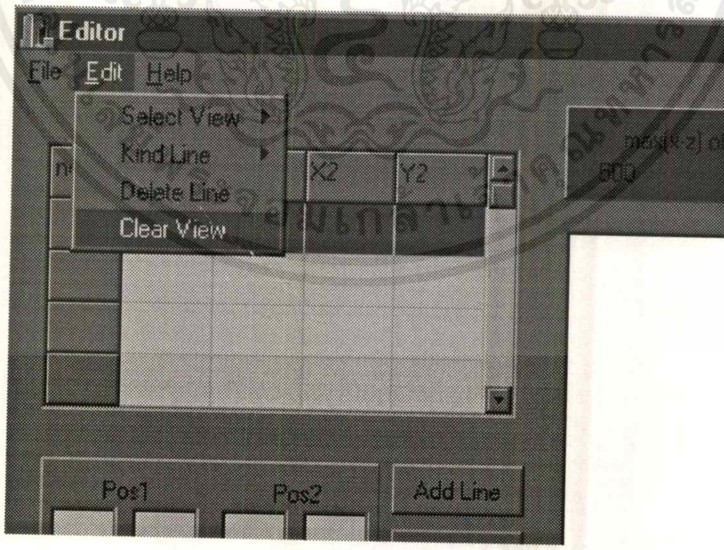
เมื่อกด Front โปรแกรมจะทำการเปลี่ยนระบบการทำงานไปที่มุมมองด้านหน้า

เมื่อกด Side โปรแกรมจะทำการเปลี่ยนระบบการทำงานไปที่มุมมองด้านข้าง



รูปที่ 3.7

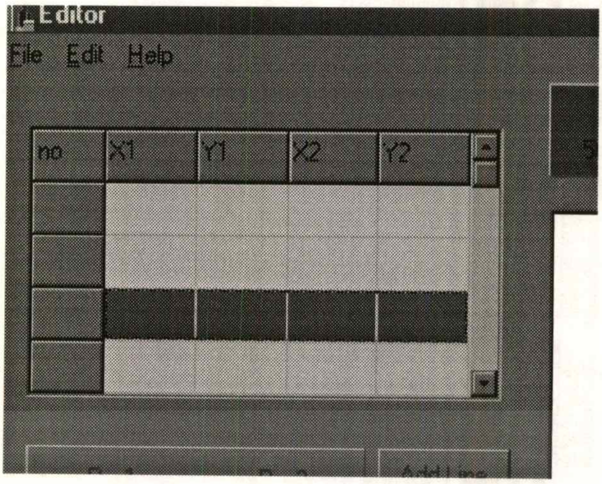
- เมื่อเลื่อน Mouse ไปที่ **Kind line** จะปรากฎเมนูย่อยที่ประกอบด้วย **Solid line** , **dot linee** ซึ่ง
- เมื่อกด Solid line โปรแกรมจะทำการสร้างเส้นในรูปแบบของเส้นทึบ
- เมื่อกด Dot line โปรแกรมจะทำการสร้างเส้นในรูปแบบของเส้นประ



รูปที่ 3.8

- เมื่อกดที่ **Clear view** โปรแกรมจะทำการลบข้อมูลของมุมมองปัจจุบัน
- **เมื่อกดที่ About โปรแกรมจะทำการแสดงข้อมูลเกี่ยวกับโปรแกรมและคณะผู้จัดทำ**

ตารางแสดงข้อมูล 2 มิติ



รูปที่ 3.9

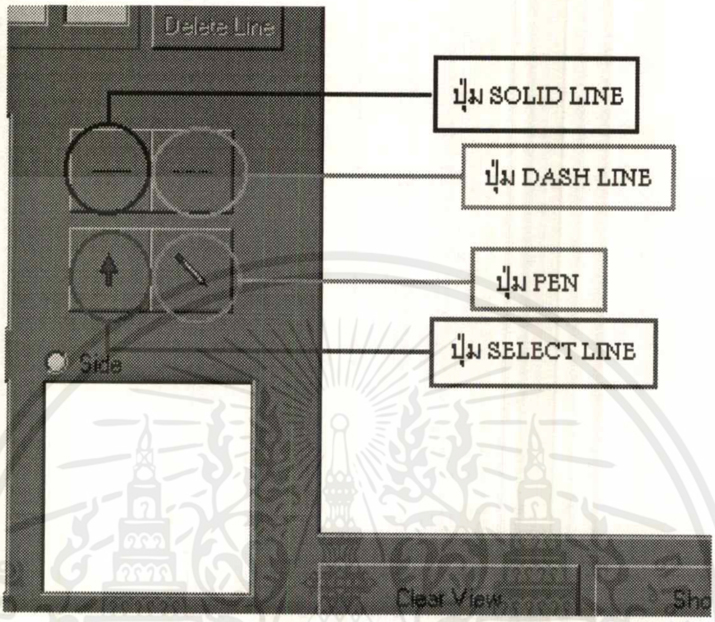
เมื่อผู้ใช้ทำการเลือกข้อมูลแถวใดแถวหนึ่งภายในส่วนนี้ จะปรากฏข้อมูลดังกล่าวบนส่วนแสดงพิกัดของเส้นปัจจุบัน (ส่วนที่ 7) และถ้าหากผู้ใช้ได้ทำการกดปุ่ม Select line ในกลุ่มเครื่องมือบนส่วนกรอบรูป (ส่วนที่ 4) ก่อนแล้ว ภาพจะปรากฏเส้นตรงสีแดงหนึ่งเส้น ซึ่งก็คือเส้นตรงที่สัมพันธ์กับข้อมูลเส้นที่ผู้ใช้ทำการเลือก

กรอบรูปแสดงข้อมูล 2 มิติ

รูปที่ 3.10

ก่อนที่ผู้ใช้จะทำงานที่ส่วนนี้ ผู้ใช้จะต้องทำการเลือกที่กลุ่มเครื่องมือบนส่วนกรอบรูป (ส่วนที่ 4) ก่อนว่าจะทำการวาดรูปหรือจะทำการเลือกเส้นที่มีอยู่แล้ว และหากผู้ใช้ต้องการวาดรูปก็ต้องเลือกอีกว่าต้องการเส้นแบบใด โดยเวลาผู้ใช้จะวาดรูปต้องทำการกดและลาก Mouse ค้างไปยังจุดที่ต้องการแล้วจึงปล่อย โดยพิกัดของกรอบรูปจะแสดงที่มุมขวาล่างของกรอบรูป

กลุ่มเครื่องมือบนส่วนกรอบรูป



รูปที่ 3.11

ประกอบด้วย

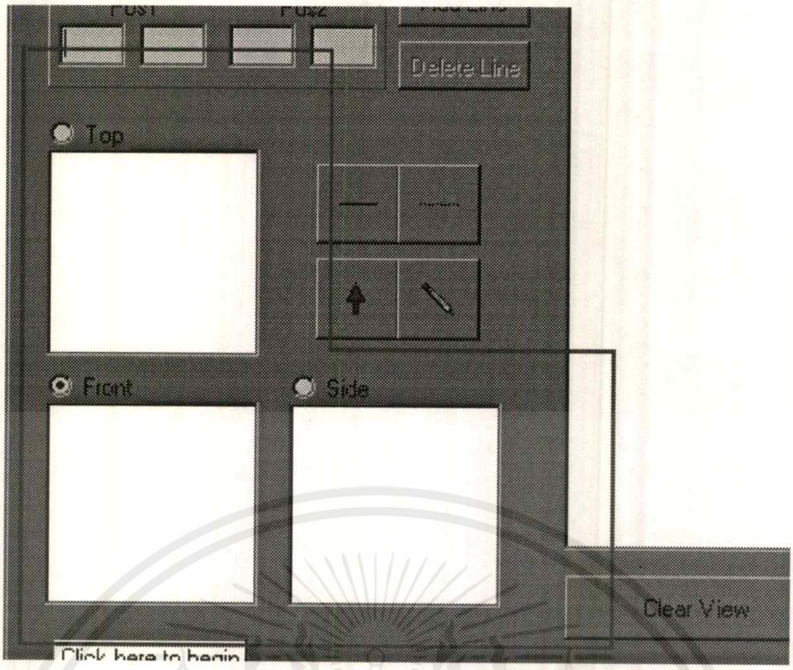
– ปุ่ม **Pen** เมื่อกดปุ่มนี้แล้วปุ่มยุบลง โปรแกรมจะรับทราบว่าที่กรอบรูปแสดงข้อมูล 2 มิติ (ส่วนที่ 3) จะทำการวาดรูป

– ปุ่ม **Select line** เมื่อกดปุ่มนี้แล้วปุ่มยุบลง โปรแกรมจะรับทราบว่าที่กรอบรูปแสดงข้อมูล 2 มิติ (ส่วนที่ 3) จะทำการเลือกเส้น

– ปุ่ม **Solid line** จะกดปุ่มนี้ได้ต่อเมื่อกดปุ่ม Pen และเมื่อกดปุ่มนี้ โปรแกรมจะทำการวาดเส้นทึบที่กรอบรูปแสดงข้อมูล 2 มิติ (ส่วนที่ 3) ทุกครั้งที่ผู้ใช้วาดเส้น

– ปุ่ม **Dash line** จะกดปุ่มนี้ได้ต่อเมื่อกดปุ่ม Pen และเมื่อกดปุ่มนี้ โปรแกรมจะทำการวาดเส้นประที่กรอบรูปแสดงข้อมูล 2 มิติ (ส่วนที่ 3) ทุกครั้งที่ผู้ใช้วาดเส้น

กลุ่มกรอบรูปภาพ 2 มิติในแต่ละมุมมอง



รูปที่ 3.12

มีหน้าที่หลัก คือ เป็นองค์ประกอบที่ผู้ใช้ใช้ในการเลือกว่าจะให้โปรแกรมทำงานที่มุมมองใด โดยถ้าหากผู้ใช้จะทำการเปลี่ยนมุมมอง จะกดที่ Check หรือ กดที่รูป เช่น เมื่อผู้ใช้ทำการกดที่รูป Side โปรแกรมจะทำงานที่ Side View คือ จะแสดงรูปของ Side ในกรอบรูปแสดงข้อมูล 2 มิติ และแสดงข้อมูลพิกัดแต่ละเส้นที่ตารางแสดงข้อมูล 2 มิติ โดยที่ทุกครั้งที่ผู้ใช้ทำการเปลี่ยน View โปรแกรมจะทำการตรวจสอบว่ารูปที่ผู้ใช้ป้อนไปนั้นถูกต้องหรือไม่

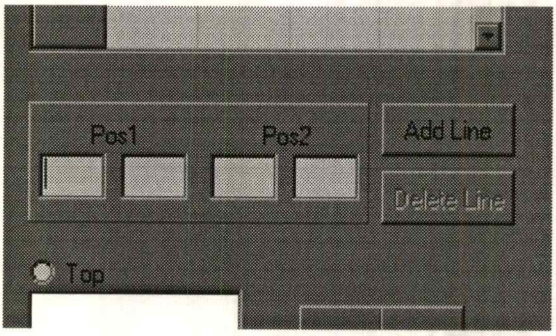
ส่วนแสดงขอบเขตของการสร้างภาพในแต่ละมุมมอง

เมื่อผู้ใช้ทำการใส่ข้อมูลลงใน View ใด View หนึ่งแล้ว โปรแกรมจะทำการหาขอบเขตใน View อื่น ๆ ที่สามารถเป็นไปได้ แสดงให้ผู้ใช้ได้รับทราบในส่วนนี้

ส่วนแสดงพิกัดของเส้นปัจจุบัน

จะแสดงข้อมูลพิกัดของเส้นตรงที่ผู้ใช้กำลังทำงานอยู่ เช่นเมื่อผู้ใช้ทำการเลือกเส้นตรงบนกรอบรูปแสดงข้อมูล 2 มิติ โปรแกรมจะแสดงพิกัดของจุดปลายทั้งสองด้านของเส้นตรงนั้น และหากผู้ใช้ต้องการจะวาดเส้นเพิ่มโดยการระบุพิกัดของจุดปลายทั้ง 2 ด้าน ก็ใส่ค่าพิกัดทั้ง 2 ด้าน และกดปุ่ม Add Line แล้วโปรแกรมจะทำการวาดเส้นที่กรอบรูปภาพ 2 มิติเพิ่มให้เอง โดยที่ส่วนนี้จะรับข้อมูลเฉพาะที่เป็นตัวเลขซึ่งมีค่าตั้งแต่ 0-400

กลุ่มเครื่องมือที่ใช้แก้ไขข้อมูล



รูปที่ 3.13

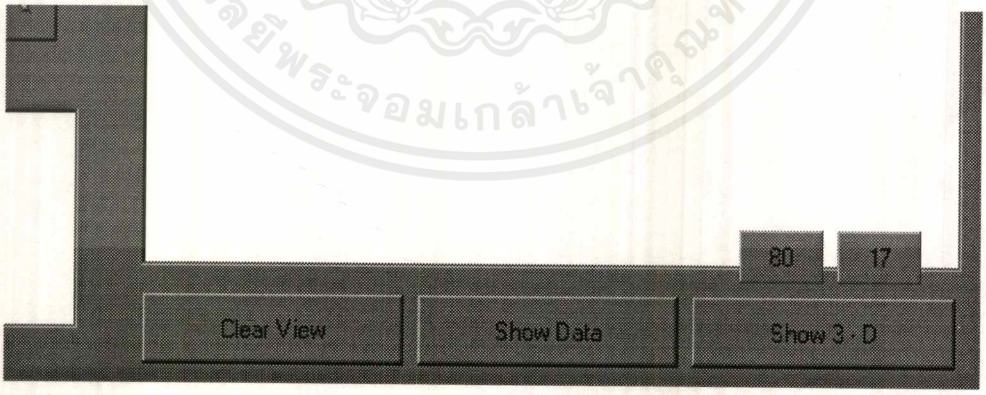
– **Add Line** เป็นปุ่มที่ใช้เพิ่มเส้น โดยจะทำการนำค่าในส่วนแสดงพิกัดของเส้นปัจจุบันไปวาดเป็นเส้นในกรอบรูปแสดงข้อมูล 2 มิติ

– **Delete Line** เป็นปุ่มที่ใช้ลบเส้น โดยจะทำงานได้เมื่อผู้ใช้ทำการเลือกเส้นแล้ว ซึ่งจะทำได้ 2 วิธี คือ

1. เลือกข้อมูลของเส้นนั้นที่ตาราง โดยอาจดูที่กรอบรูปแสดงข้อมูล 2 มิติว่าใช้เส้นที่ต้องการลบหรือไม่
2. เลือกเส้นที่กรอบรูปแสดงข้อมูล 2 มิติ

โดยทั้งสองวิธี โปรแกรมจะทำการแสดงค่าพิกัดของเส้นที่ผู้ใช้เลือกไว้ในส่วนแสดงพิกัดของเส้นปัจจุบัน และโปรแกรมจะทำการถามผู้ใช้ก่อนว่าต้องการจะลบหรือไม่

กลุ่มเครื่องมืออื่น ๆ



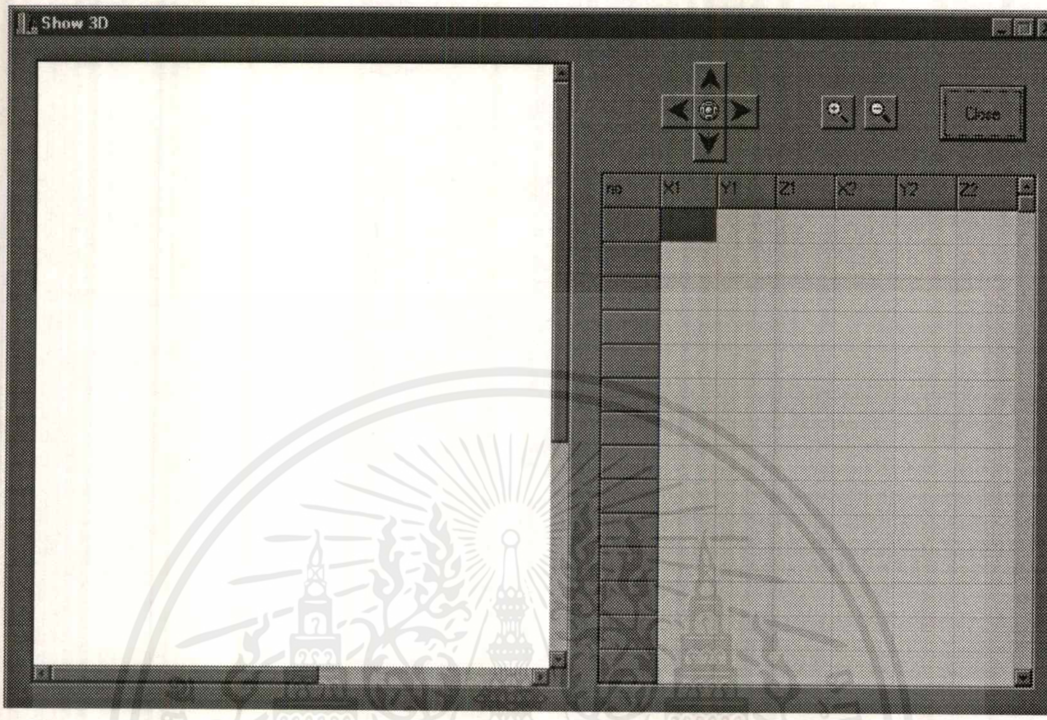
รูปที่ 3.14

– ปุ่ม **Clear View** จะทำการลบภาพและข้อมูลทั้งหมดของมุมมองปัจจุบัน

– ปุ่ม **Show Data** จะทำการแสดงค่าพิกัดของแต่ละเส้นในมุมมองปัจจุบันที่ตารางแสดงข้อมูล 2 มิติ

— ปุ่ม **Show 3 - D** จะทำการนำภาพของทั้ง 3 มุมมองไปประมวลผลเป็นภาพ 3 มิติ แล้วนำมาแปลงเพื่อแสดงภาพออกในส่วนของหน้าจอแสดงรูป 3 มิติ

หน้าจอแสดงรูป 3 มิติ ประกอบด้วย



รูปที่ 3.15

— ตารางแสดงค่าพิกัด x,y,z ของแต่ละเส้นว่ามีจุดปลายอยู่ที่ใด

— กรอบรูปแสดงภาพ 3 มิติที่ได้จากการประมวลภาพ 2 มิติ 3 มุมมองจากการหมุนและจากการขยายรูป

— หมุน ผู้ใช้จะทำการเลือกว่าจะหมุนไปเป็นมุมเท่าไร โดยจะคิดจากรูปที่ได้จากการประมวลผลภาพ 3 มิติ

— ขยาย ผู้ใช้จะทำการเลือกว่าจะขยายไปเป็นอัตราเท่าไร โดยจะคิดจากรูปที่ได้จากการประมวลผลภาพ 3 มิติ

— ปุ่ม Close เมื่อกดปุ่มนี้ โปรแกรมจะทำการคืนสู่หน้าจอหลัก

บทที่ 4

การทดลองและผลการทดลอง

4.1. Hardware ที่ใช้ในการทดลอง

<u>เครื่องคอมพิวเตอร์</u>

– หน่วยประมวลกลาง (CPU) Pentium 100 MH หรือสูงกว่า

– หน่วยความจำหลัก (Main Memory) ขนาดไม่ต่ำกว่า 16 เมกะไบต์

– หน่วยความจำความเร็วสูง (Cache Memory) ชนิดภายในหน่วยประมวลผลกลางขนาดไม่น้อยกว่า 16 กิโลไบต์ และชนิดภายนอกหน่วย

– ประมวลผลกลางขนาดไม่น้อยกว่า 256 กิโลไบต์

– เครื่องขับจานแม่เหล็กอย่างอ่อน (Floppy Disk Drive) ขนาด 3.5 นิ้ว จำนวน 1 หน่วย

– จอภาพสีชนิดรายละเอียดสูง ขนาดไม่ต่ำกว่า 14 นิ้วตามเส้นทะแยงมุมซึ่งสามารถใช้แสดงภาพที่ได้รับจากวงจรแสดงผลกราฟฟิค รายละเอียดไม่น้อยกว่า 1,024*768 pixels แบบ Non-Interlace

– คีย์บอร์ด (Keyborad) ที่มีอักษรภาษาไทย / ภาษาอังกฤษ ตัวเลขและเครื่องหมายสัญลักษณ์พิเศษอย่างน้อย 101 คีย์

– มีอุปกรณ์ป้อนคำสั่งแบบเมาส์ (Mouse) ที่มีปุ่ม 2 ปุ่ม หรือ 3 ปุ่ม

4.2. Software ที่ใช้ในการทดลอง

– ระบบปฏิบัติการไมโครซอฟวินโดว์95(Microsoft Window95)

– โปรแกรม Borland c++ Builder (Client-Server)

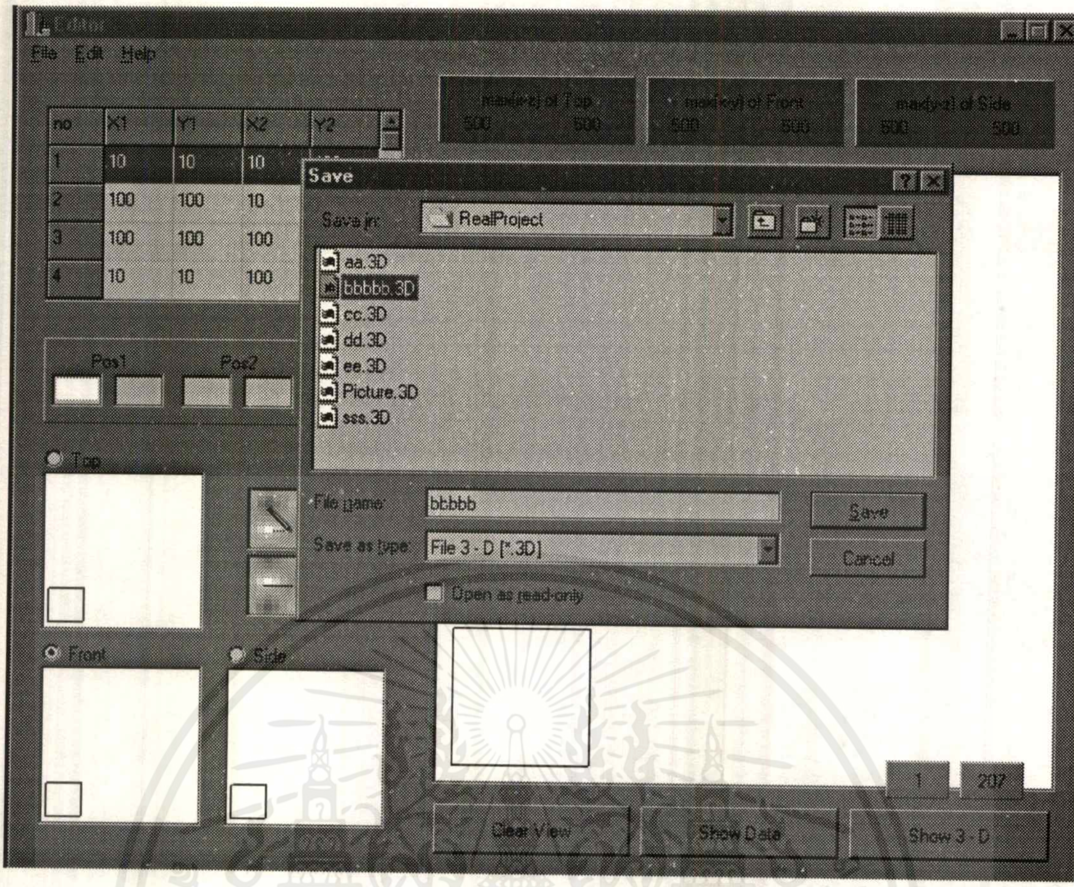
4.3. การทดลองและผลการทดลอง

<u>เหตุการณ์ที่ 1</u>

สมมุติว่าต้องการแสดงภาพ 3 มิติของวัตถุจริงชิ้นหนึ่งออกทางจอภาพ Computer user จะต้องทำการศึกษาวัตถุนั้นก่อนว่าแต่ละมุมมองวาดภาพได้อย่างไร (อาศัยความรู้ทางด้าน Drawing) จากนั้นก็ทำการเรียกใช้โปรแกรม โดยขั้นตอนการใช้โปรแกรมมีดังนี้

1. ณ. ขณะนี้ user ต้องการจะวาดรูปดังนั้น user จะต้องตรวจสอบว่า กดปุ่ม Pen หรือยัง และทำการเลือกชนิดของเส้นตามต้องการหรือยัง ดังรูปที่

2. ทำการวาดรูป โดยวิธีการวาดมี 2 วิธี

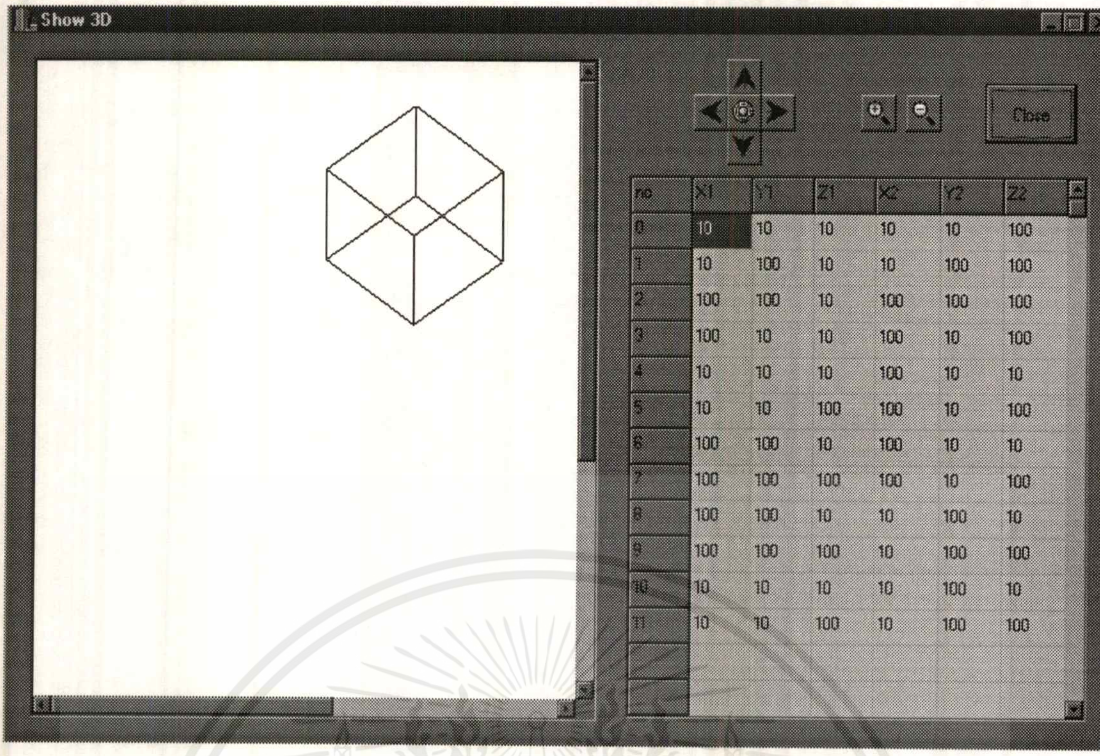


รูปที่ 4.1

– ใช้ Mouse วาดที่กรอบรูปแสดงภาพ 2 มิติ โดยใช้การกด Mouse แล้วกดค้าง และปล่อยที่จุดปลายที่ต้องการโดย user สามารถดูพิกัดปัจจุบันที่ Mouse ลากผ่านได้ที่ มุมกรอบขวาล่าง

– คีย์ข้อมูลลงในส่วนแสดงพิกัดของเส้นปัจจุบันแล้วกดปุ่ม Add Line

3. เมื่อวาดรูปของ View ปัจจุบันเสร็จแล้ว ทำการเปลี่ยน View โดยกดที่รูปของ View ที่ต้องการ (ในส่วนของกลุ่มกรอบรูปภาพ 2 มิติ) จากนั้นก็ทำการวาดรูปต่อจนครบ 3 View



รูปที่ 4.2

4. กดปุ่ม Show 3-D ก็จะปรากฏหน้าจอใหม่ (หน้าจอแสดงภาพ 3 มิติ) โดยที่หน้าจอนี้จะสามารถทำการหมุนภาพ (ในเชิง 2 มิติ) และย่อขยายภาพได้ เมื่อกดปุ่ม Close ก็จะทำการคืนการทำงานสู่หน้าจอหลัก

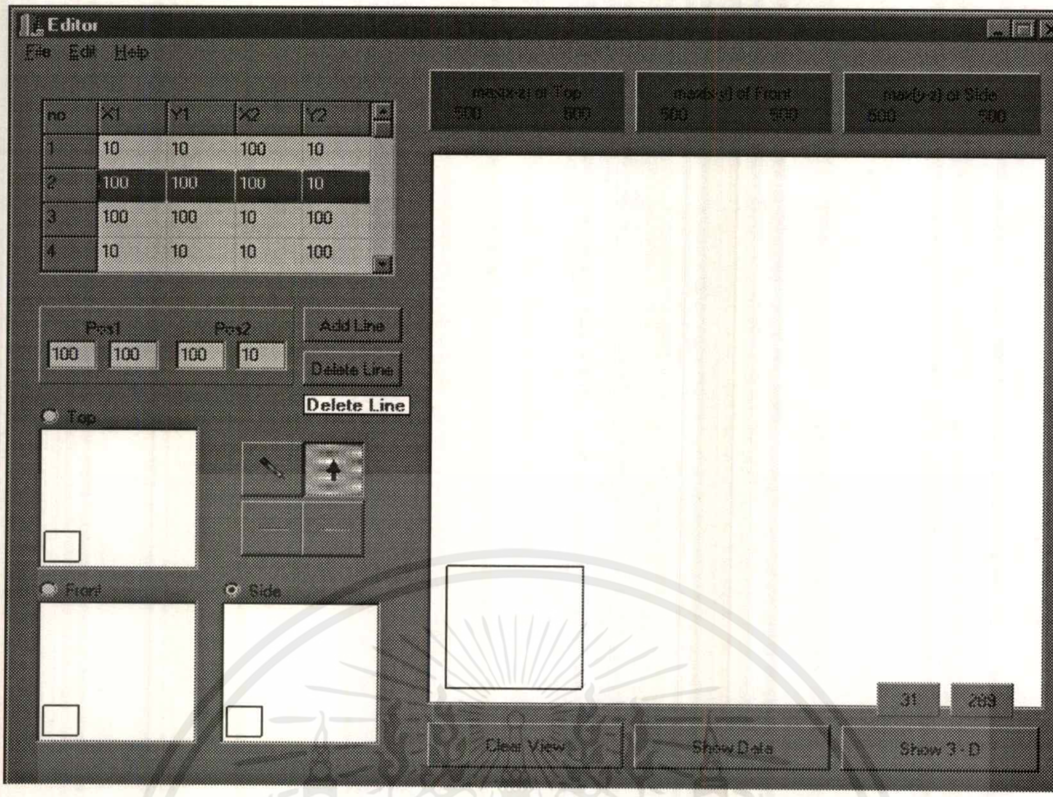
5. หากต้องการ Save โปรแกรมจะขึ้นหน้าจอให้ใส่ชื่อไฟล์

<u>เหตุการณ์ที่ 2</u>

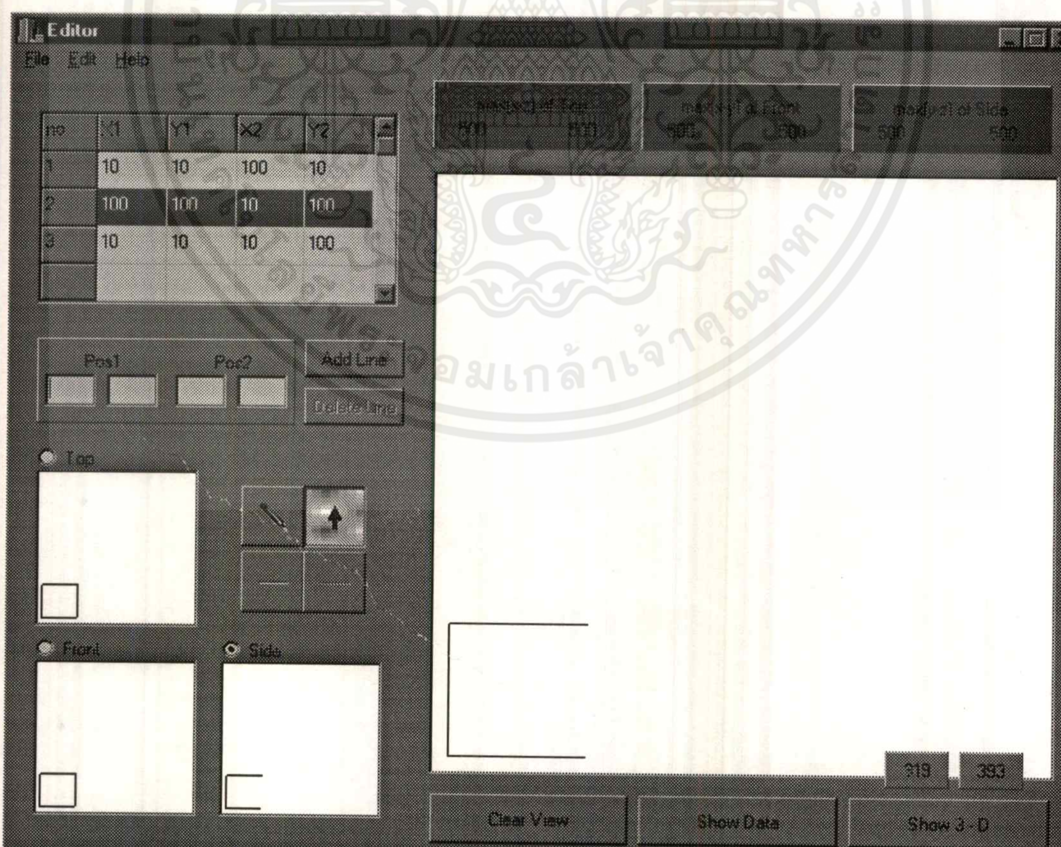
สมมุติมีไฟล์(*.3D)อยู่และต้องการเปิดมาเพื่อทำการแก้ไข มีขั้นตอนดังนี้

1. ทำการเรียกใช้โปรแกรม
2. กดที่เมนู->File->เลือกไปที่ Open
3. จะปรากฏหน้าจอให้ใส่ชื่อไฟล์ที่ต้องการเปิด และจะปรากฏข้อมูลภาพของ Front View และข้อมูลของ Front View คือการทำงานตอนนี้จะอยู่ที่ Front View และจะแสดงภาพของ View อื่นๆที่รูปย่อถ้าต้องการลบเส้นจะมี 2 วิธี

— กดปุ่มรูปลูกศรแล้วนำ Mouse ไปกดรูปที่เส้นที่ต้องการลบ(จะเป็นเส้นสีแดง) แล้วกดปุ่ม Delete


Editor
File Edit Help
no X1 Y1 X2 Y2
Pos1 Pos2
Add Line
Delete Line
Delete Line
Top
Front
Side
Clear View
Show Data
Show 3 - D

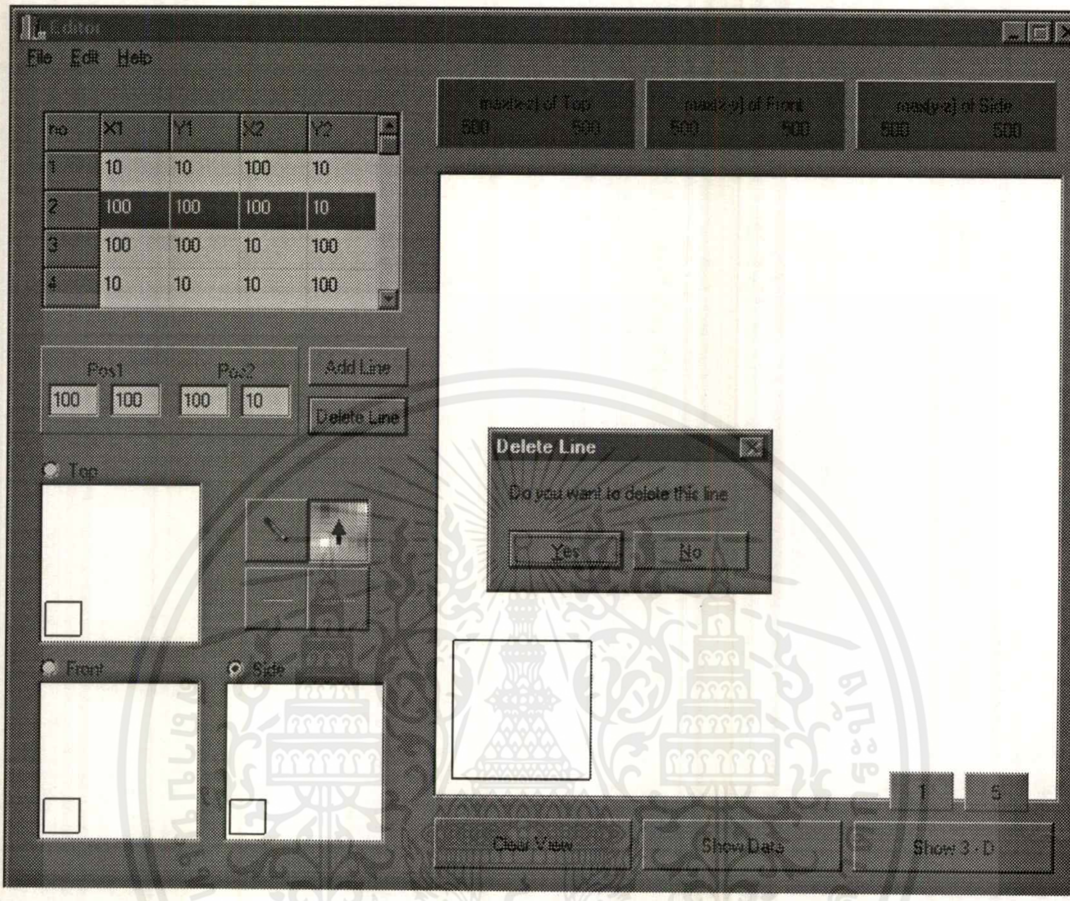
รูปที่ 4.3


Editor
File Edit Help
no X1 Y1 X2 Y2
Pos1 Pos2
Add Line
Delete Line
Top
Front
Side
Clear View
Show Data
Show 3 - D

รูปที่ 4.4

— เลือกข้อมูลในตาราง เช่นถ้ากดแถว 3 แล้วจะปรากฏเส้นที่แสดงหนึ่งเส้นที่กรอบรูป แล้วจึงทำการกดปุ่ม Delete Line

1. โปรแกรมจะขึ้นหน้าจอถามว่าต้องการลบหรือไม่



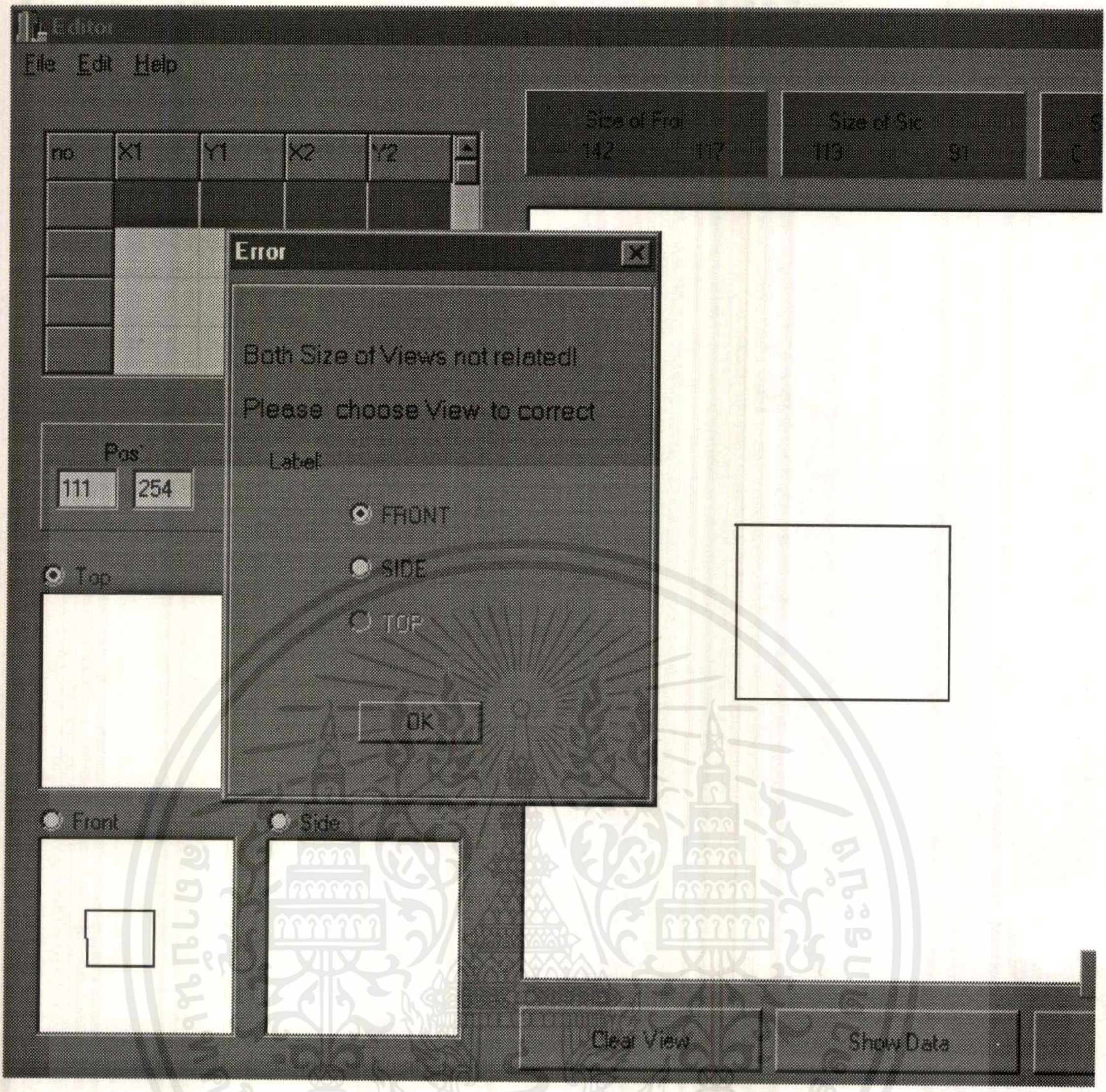
รูปที่ 4.5

— ถ้ากด Yes โปรแกรมจะทำการลบเส้นนั้นออกจาก View

— ถ้ากด NO โปรแกรมจะไม่ทำการลบ ก็คือเสมือนว่าเราไม่ได้ทำอะไร

<u>เหตุการณ์ที่ 3</u>

— กรณีเกิดข้อผิดพลาดเนื่องจากข้อมูลของภาพ 2 มิติ ที่ใส่เข้ามาไม่มีความสัมพันธ์ตามกฎการรวมภาพ 3 มิติกับข้อมูลภาพที่ทำการใส่แล้วก่อนหน้านี้ จะทำให้เกิด Dialog error เกิดขึ้น โดยความสัมพันธ์ที่ไม่สอดคล้องกันเกิดจาก



รูปที่ 4.6

— กรณีมีข้อมูลของด้านหน้า แต่ข้อมูลด้านข้างที่ใส่เข้ามามีความสูงไม่เท่ากับความสูงของด้านหน้า หรือในทางกลับกันมีข้อมูลด้านข้างแล้วใส่ข้อมูลด้านหน้าที่มีความสูงไม่เท่ากับด้านข้าง เหตุการณ์นี้จะทำให้เกิด Dialog Box Error และให้เลือกว่าจะแก้ไขข้อมูลด้าน Front หรือด้านSideกรณีมีข้อมูลของด้านหน้า แต่ข้อมูลด้านบนที่ใส่เข้ามามีความกว้างไม่เท่ากับความกว้างของด้านหน้า หรือในทางกลับกันมีข้อมูลด้านบนแล้วใส่ข้อมูลด้านหน้าที่มีความกว้างไม่เท่ากับด้านบน เหตุการณ์นี้จะทำให้เกิด Dialog Box Error และให้เลือกว่าจะแก้ไขข้อมูลด้านหน้า หรือด้านบน

— กรณีมีข้อมูลของด้านข้าง แต่ข้อมูลด้านบนที่ใส่เข้ามามีความสูงไม่เท่ากับความกว้างของด้านข้าง หรือในทางกลับกันมีข้อมูลด้านบนแล้วใส่ข้อมูลด้านข้างที่มีความกว้างไม่เท่ากับความสูงด้านบน เหตุการณ์นี้จะทำให้เกิด Dialog Box Error และให้เลือกว่าจะแก้ไขข้อมูลด้านข้าง หรือด้านบน

— กรณีมีการใส่ข้อมูลภาพ 2 มิติแล้ว 2 มุมมอง แล้วมีการใส่ภาพที่ 3 เข้ามา ที่มีความสัมพันธ์ไม่สอดคล้องกับข้อมูลภาพ ที่มีอยู่ก่อนหน้านี้ 2 ภาพ จะปรากฏ Dialog Box Error ให้ทำการเลือกว่าจะแก้ไขข้อมูลที่

มุมมองใด เมื่อปรากฏ Dialog Box Error เนื่องจากข้อมูลไม่สัมพันธ์กันให้ทำการ Click เลือกว่าจะแก้ข้อมูลทางด้านใด เมื่อ Click ที่ Radio Button นั้นแล้วให้ Click OK โปรแกรมจะทำการลบข้อมูลของภาพ2มิติในมุมมองที่ถูกเลือกออกมาทั้งหมดแล้วให้ผู้ใช้ทำการใส่ข้อมูลใหม่

บทที่ 5
สรุป วิจารณ์ และแนวทางการพัฒนา

5.1. ความสามารถของโปรแกรม

— สามารถสร้างภาพ 3 มิติจากข้อมูลของภาพ 2 มิติได้

— สามารถบันทึก และแก้ไขข้อมูลของภาพ 2 มิติได้

— สามารถแสดงภาพ 3 มิติออกทางจอภาพได้

5.2. ข้อจำกัดของโปรแกรม

เนื่องจากวิชา Drawing นั้นเป็นวิชาที่ต้องอาศัยทักษะ และการฝึกฝนในการมองวัตถุในมุมมองต่าง ๆ ดังนั้นการสร้างซอฟแวร์ที่รองรับการสร้างภาพ 3 มิติในรูปแบบใด ๆ ก็ได้นั้นเป็นเรื่องยาก สำหรับข้อจำกัดของซอฟแวร์นี้ สามารถสร้างได้เฉพาะรูปทรงเหลี่ยมอย่างง่ายเท่านั้น

5.3. แนวทางในการพัฒนาต่อไป

เนื่องจากวัตถุที่เราสามารถสร้างเป็นภาพ 3 มิติได้นั้น สามารถสร้างได้เพียงวัตถุรูปทรงเหลี่ยมอย่างง่าย และแนวความคิดในการสร้างภาพ 3 มิตินั้น ยังไม่ครอบคลุมวัตถุในทุก ๆ กรณี ดังนั้นจึงสามารถพัฒนาโปรแกรมให้สมบูรณ์ยิ่งขึ้นได้ โดย อาจเริ่มจาก การรับข้อมูลเข้า 2 มิติจากทั้ง 6 มุมมองเพื่อให้ได้ข้อมูลเข้าที่มากขึ้น หรือ อาจเปลี่ยนวิธีการจัดเก็บระบบพิกัดจาก "Physical Coordinate System" เป็น "Spherical Coordinate System"

บรรณานุกรม

1. Procedural Element for computer graphics / by David F. Rogers , McGraw_Hill Co , 1985
2. Computer Graphics principles and Practice / by Foley , VanDam , Feiner , Hughes , Addison_Wesley Publishing company , Inc , 1992
3. Mathematical Elements for Computer Graphics / by David F. Rogers , J.Alan Adam , McGraw Hill , 1990
4. Teach Yourself Borland C++Builder in 21 Day / by Kent Reisdorph , Ken Henderson , 1997
5. Procedural Element for computer graphics / by Rogess , D.F. , McCraw Hill Co. , 1995