Computer Graphics and Visualisation

Geometry for Computer Graphics

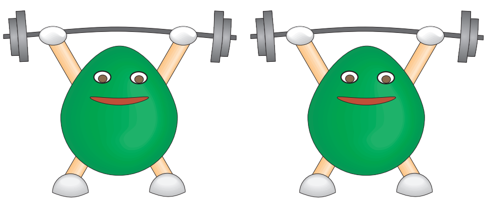
Overhead Projection (OHP) Overviews

Developed by

F Lin
K Wyrwas
J Irwin
C Lilley
W T Hewitt
T L J Howard

Computer Graphics Unit
Manchester Computing Centre
University of Manchester

Department of Computer Science
University of Manchester



THE UNIVERSITY of MANCHESTER

Produced by the ITTI Gravigs Project
Computer Graphics Unit
Manchester Computing Centre
MANCHESTER M13 9PL

First published by UCoSDA in May 1995

ISBN 1 85889 059 4

For further information on this and other modules please contact:
The ITTI Gravigs Project, Computer Graphics Unit, Manchester Computing Centre.
Tel: 0161 275 6095 Email: *gravigs@mcc.ac.uk*

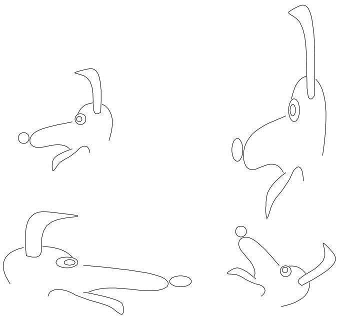
To order further copies of this or other modules please contact:
Mrs Jean Burgan, UCoSDA.
Tel: 0114 272 5248 Email: *j.burgan@sheffield.ac.uk*

These materials have been produced as part of the Information Technology Training Initiative, funded by the Information Systems Committee of the Higher Education Funding Councils.

The authors would like to thank Janet Edwards for her assistance in the preparation of these documents.

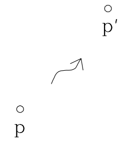
Printed by the Reprographics Department, Manchester Computing Centre from PostScript source supplied by the authors.

Two-Dimensional Transformations

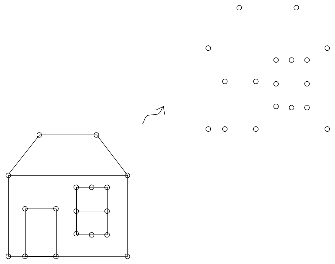


Two-Dimensional Transformations

- Transforming a point

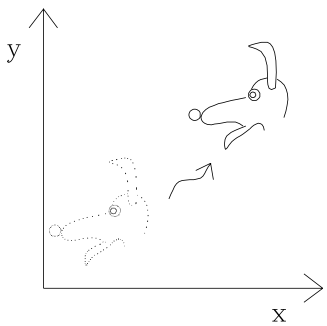


- Transforming an object



Types of Transformation

- Translation

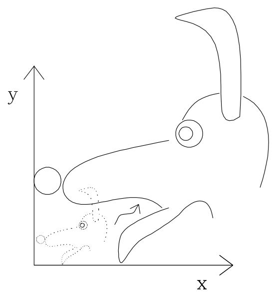


$$x' = x + t_x$$

$$y' = y + t_y$$

Types of Transformation

- Scaling



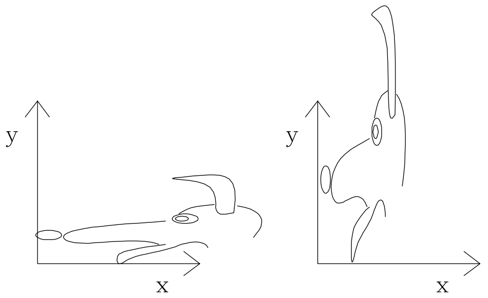
Sx=Sy=3

$$x' = x \cdot s_x$$

$$y' = y \cdot s_y$$

Symmetric vs asymmetric scaling

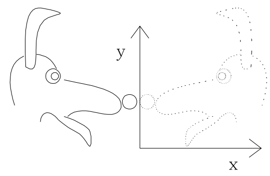
- Symmetric scaling: $s_x = s_y$
- Asymmetric scaling: $s_x > s_y$ or $s_x < s_y$



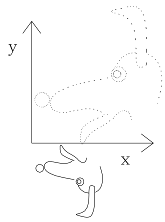
sx > sy   sx < sy

Scaling to achieve reflection

- Reflection in *y* axis: $s_x < 0$



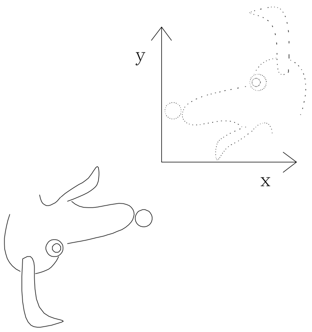
- Reflection in *x* axis: $s_y < 0$



Scaling to achieve reflection

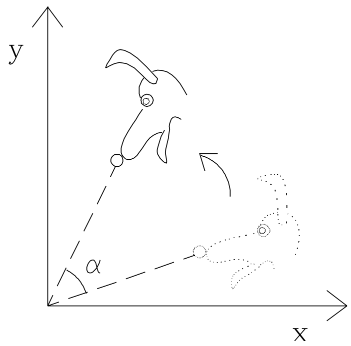
- Reflection in *x* and *y* axes:

  $s_y < 0$ *and* $s_x < 0$

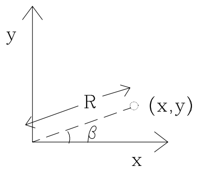


Types of Transformation

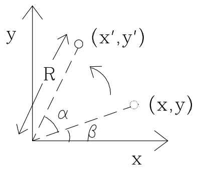
- Rotation



Rotation



$$x = R \cdot \cos\beta$$
$$y = R \cdot \sin\beta$$



$$x' = R \cdot \cos(\alpha + \beta)$$
$$y' = R \cdot \sin(\alpha + \beta)$$

Rotation

$$x' = R \cdot \cos(\alpha + \beta)$$
$$y' = R \cdot \sin(\alpha + \beta)$$

Expanding the formulae for $\cos(\alpha+\beta)$ and $\sin(\alpha+\beta)$:

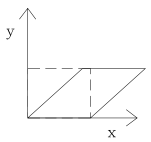
$$x' = R \cdot \cos\alpha \cdot \cos\beta - R \cdot \sin\alpha \cdot \sin\beta$$
$$y' = R \cdot \sin\alpha \cdot \cos\beta + R \cdot \sin\beta \cdot \cos\alpha$$

Substituting for $R \cdot \cos\beta$ and $R \cdot \sin\beta$:

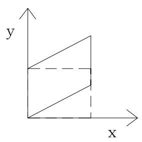
$$x' = x \cdot \cos\alpha - y \cdot \sin\alpha$$
$$y' = x \cdot \sin\alpha + y \cdot \cos\alpha$$

Types of Transformation

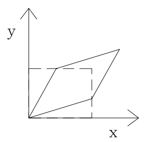
- Shearing



shear in x



shear in y



shear in x and y

$$x' = x + y \cdot a$$
$$y' = y + x \cdot b$$

- ❏ Shear in x: $a \neq 0$
- ❏ Shear in y: $b \neq 0$
- ❏ Shear in x and y: $a \neq 0$ and $b \neq 0$

Matrix Representation of Transformations

- Translate:

$$x' = x + t_x$$
$$y' = y + t_y$$

- Scale:

$$x' = x \cdot s_x$$
$$y' = y \cdot s_y$$

- Rotate:

$$x' = x \cdot \cos\alpha - y \cdot \sin\alpha$$
$$y' = x \cdot \sin\alpha + y \cdot \cos\alpha$$

Matrix Representation of Transformation

- Shear:

$$x' = x + y \cdot a$$
$$y' = y + x \cdot b$$

- In general:

$$x' = a \cdot x + b \cdot y + c$$
$$y' = d \cdot x + e \cdot y + f$$

Matrix Representation of Transformations

$$x' = a \cdot x + b \cdot y + c$$
$$y' = d \cdot x + e \cdot y + f$$

Using matrices:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ d & e \end{bmatrix} \bullet \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ f \end{bmatrix}$$

Include *a* - *f* in one matrix:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Matrix Representation of Transformations

$$x' = a \cdot x + b \cdot y + c$$
$$y' = d \cdot x + e \cdot y + f$$

Using a square matrix:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Matrix Representation of Transformations

- Translate: $\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$

- Scale: $\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$

- Rotate: $\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Matrix Representation of Transformations

- Shear: $\begin{bmatrix} 1 & a & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- Identity matrix:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = x$$
$$y' = y$$
$$w' = 1$$

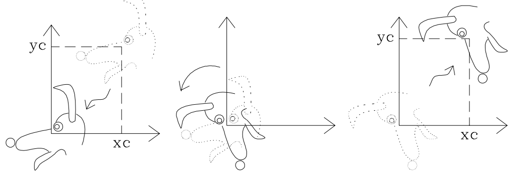
Combining Transformations

Can perform complex transformations by combining simple ones.

For example, rotating an object about its centre:

Combining Transformations

1. Translate by ($-x_c$, $-y_c$)
2. Rotate about the origin
3. Translate by (x_c, y_c)



1 2 3

Combining Transformations

Rotating an object about its centre:

1. Translate by ($-x_c$,$-y_c$):

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2. Rotate about the origin:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Combining Transformations

3. Translate by (x_c, y_c):

$$\begin{bmatrix} x_3 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_c \\ 0 & 1 & y_c \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

Total transformation is:

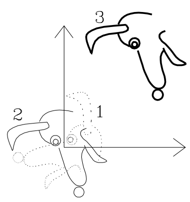
$$\begin{bmatrix} 1 & 0 & x_c \\ 0 & 1 & y_c \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{bmatrix}$$

Ordering Transformations

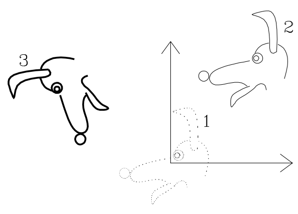
- Matrix multiplication is NOT commutative, $\mathbf{M}_1 \bullet \mathbf{M}_2 \neq \mathbf{M}_2 \bullet \mathbf{M}_1$
- Order of transformations is important

Ordering Transformations

- Rotate then translate



- Translate then rotate



Ordering Transformations

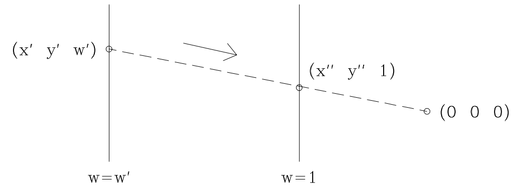
- POSTCONCATENATE $\mathbf{M}_2$ with $\mathbf{M}_1$:
  $\mathbf{p}' = \mathbf{M}_2 \bullet \mathbf{M}_1 \bullet \mathbf{p}$ ($\mathbf{M}_1$ applied first)
- PRECONCATENATE $\mathbf{M}_2$ with $\mathbf{M}_1$:
  $\mathbf{p}' = \mathbf{M}_1 \bullet \mathbf{M}_2 \bullet \mathbf{p}$ ($\mathbf{M}_2$ applied first)
- Premultiply $\equiv$ Postconcatenate
- Postmultiply $\equiv$ Preconcatenate

Homogeneous Coordinates

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Point in **2D** space expressed in **3D homogeneous** coordinates
- If bottom row of matrix is [0 0 1], $w' = 1$
- If $w' \neq 1$ project point (x', y',w') onto plane w =1 by **homogeneous division** (using the origin as the centre of projection)

Homogeneous Coordinates



- Real world coordinates are x'' and y'' where

$$x'' = x'/w'$$
$$y'' = y'/w'$$

Homogeneous Coordinates

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 4 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

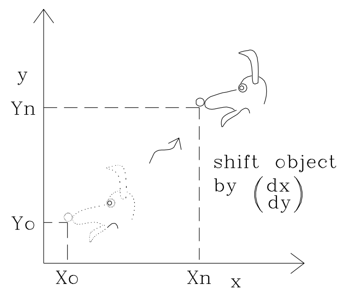
Perform homogeneous division to get “real world” coordinates:

$$x'' = x'/w' = x/4$$
$$y'' = y'/w' = y/4$$

Effect is OVERALL scaling.

Object vs Axis Transformation

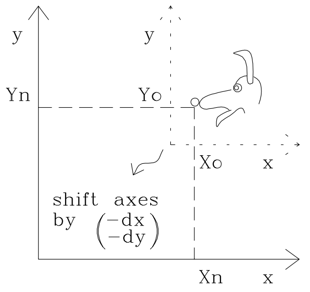
Object transformation



- Object transformed
- Axes fixed

Object vs Axis Transformation

Axis transformation

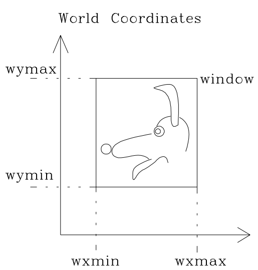


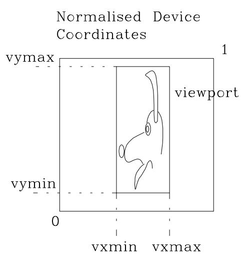
- Object fixed in space
- Axes transformed

Object vs Axis Transformation

- Object translation by (d_x, d_y)
  - ≡ Axis translation by ($-d_x$, $-d_y$)
- Object scale by (s_x, s_y)
  - ≡ Axis scale by ($1/s_x$, $1/s_y$)
- Object rotation by α
  - ≡ Axis rotation by $-\alpha$

Normalization Transformation in GKS





1. Translate bottom left hand corner of window to origin:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -w_{xmin} \\ 0 & 1 & -w_{ymin} \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Normalization Transformation in GKS

2. Scale window to size of viewport:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

where

$$s_x = \frac{v_{xmax} - v_{xmin}}{w_{xmax} - w_{xmin}}$$

$$s_y = \frac{v_{ymax} - v_{ymin}}{w_{ymax} - w_{ymin}}$$

Normalization Transformation in GKS

3. Translate to bottom left hand corner of viewport:

$$\begin{bmatrix} x_3 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & v_{xmin} \\ 0 & 1 & v_{ymin} \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

Normalization transformation is:

$$\begin{bmatrix} 1 & 0 & v_{xmin} \\ 0 & 1 & v_{ymin} \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & 0 & -w_{xmin} \\ 0 & 1 & -w_{ymin} \\ 0 & 0 & 1 \end{bmatrix}$$

Two-Dimensional Transformations

- Different types: translation, scaling, rotation, shearing.
- Object vs axis transformations
- Matrix representation

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Combine transformations by multiplying matrices
- Homogeneous division to get "real world" coordinates

$$x'' = x'/w', y'' = y'/w'$$

Three-Dimensional Transformations

Three-Dimensional Transformations

- For manipulating pictures (as in 2D)
- Help us to understand 3D shape
- Right-handed coordinate system