Paper Code: BCA 306

Paper ID: 20306

Paper: Computer Graphics & Multimedia Applications

UNIT – I
Introduction: The Advantages of Interactive Graphics, Representative Uses of Computer Graphics, Classification of Applications, Development of Hardware and Software for Computer Graphics, Conceptual Framework for Interactive Graphics, Overvie w, Scan Converting Lines, Scan Converting Circles, Scan Converting Ellipses.
Graphics Hardware
Hardcopy Technologies, Display Technologies, Raster-Scan Display Systems, The Video
Controller, Random-Scan Display Processor, Input Devices for Operator Interaction, Image Scanners, Working exposure on graphics tools like Dream Weaver, 3D Effects etc.
Clipping
Southland-Cohen Algorithm, Cyrus-Beck Algorithm, Midpoint Subdivision Algorithm
[No. of Hrs.: 12]
UNIT – II
Geometrical Transformations
2D Transformations, Homogeneous Coordinates and Matrix Representation of 2D Transformations, Composition of 2D Transformations, The Window-to-Viewport Transformation, Efficiency, Matrix Representation of 3D Transformations, Transformations
as a Change in Coordinate System. **[No. of Hrs.: 10]**
UNIT – III
Representing Curves & Surfaces
Polygon Meshes, Parametric Cubic Curves, Quadric Surfaces.
Solid Modeling
Representing Solids, Regularized Boolean Set Operations, Primitive Instancing, Sweep Representations, Boundary Representations, Spatial Partitioning Representations, Constructive Solid Geometry, Comparison of Representations, User Interfaces for Solid Modeling. **[No. of Hrs.: 10]**
UNIT – IV
Introductory Concepts: Multimedia, Defintion, CD-ROM and the multimedia highway,
Uses of Multimedia, Introduction to making multimedia – The stages of Project, the hardware & software requirements to make good multimedia, Multimedia skills and training,
Training Opportunities in Multimedia,

Motivation for Multimedia usage

UNIT-I

Computer graphics tells us that what are the actual working of graphics. Computer Graphics remains one of the most existing and repladly growing computer feild. Computer Graphics as the pictorial representation or graphical representation of a computer

1. **Application of computer graphics:**

(1) Computer – Aided Design

(2) Presentation Graphics

(3) Computer Art

(4) Entertainment

(5) Education and Training

(6) Graphics provides one of the most natural means of communicating with a computer.

(7) Interactive computer graphics permits extensive, high-bandwidth user-computer interaction.

---

2. Representative Uses of Computer Graphics

Computer graphics is used today in many different areas of industry, business, government, education, and entertainment.

- User interfaces: GUI, etc.
- Business, science and technology: histograms, bar and pie charts, etc.
- Office automation and electronic publishing: text, tables, graphs, hypermedia systems, etc.
- Computer-aided design (CAD): structures of building, automobile bodies, etc.
- Simulation and animation for scientific visualization and entertainment: flight simulation, games, movies, virtual reality, etc.
- Art and commerce : terminals in public places such as museums, etc.
- Cartography: map making
- Simulation and animation for scientific visualization and entertainment.
- Multimedia textbooks

---

3. Classification of Applications

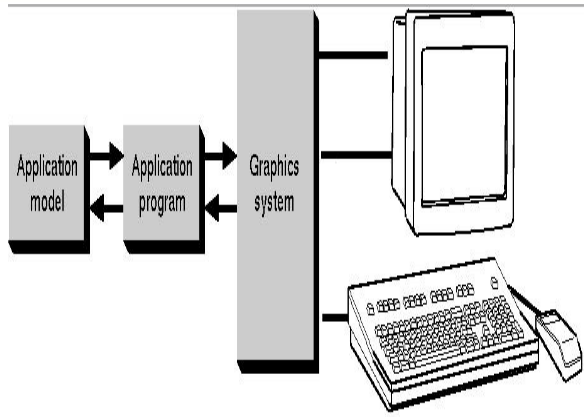
- **Paint programs**: Allow you to create rough freehand drawings.
- **Animation software**: Enables you to chain and sequence a series of images to simulate movement. Each image is like a frame in a movie
- **CAD software**: Enables architects and engineers to draft designs. It is the acronym for computer-aided design. A CAD system is a combination of hardware and software that enables

engineers and architects to design everything from furniture to airplanes.

- **Desktop publishing**: Provides a full set of word-processing features as well as fine control over placement of text and graphics, so that you can create newsletters, advertisements, books, and other types of documents
- **business software:** enables users to create highly stylized images for slide shows and reports. The software includes functions for creating various types of charts and graphs and for inserting text in a variety of fonts.

4. Conceptual Framework for Interactive Graphics.

The high-level conceptual framework shown here can be used to describe almost any interactive graphics system



The three major parts of the framework are:

Application Modeling

Calculating what is to be displayed

Displaying the Model

Calling the graphics API routines

Interaction Handling

Handling user interaction, which will change the model, and therefore the display.

typically an event driven loop

5. Scan Converting Lines

Converting the geometric definition of a primitive form into a set of pixels that make up the primitive in the image space. This conversion task is scan conversion.

Types of Scan Conversion

1. Digital Differential (DDA) Algorithm
2. Bresenham's Line Algorithm

DDA algorithm is an incremental scan conversion method.

- Incremental scan-conversion method
- Faster than the direct use of the line equation
- a floating point operation is still required
- The line drifts away from the original line when the line is relatively long

AN ALGORITHM TO DRAW A LINE

1. Compute

   dx = x2-x1 dy = y2-y1

2. If abs(dx) > abs(dy) then steps = abs(dx)

3. Else steps = abs(dy)

4 Plot a point at (x, y)

5. xinc = dx / steps;

6. yinc = dy/steps;

7. x = x1 and y = y1

8. Plot a point at (x, y)

9. k=1

10. if k = steps , stop

11. x = x +xinc

12. y = y + yinc

13. Plot a point at (x, y)

14. k = k+1

15. Go to step 7

BRESENHAM LINE ALGORITHM

An accurate and efficient raster line generating algorithm, the **Bresenham's line-drawing algorithm.** This algorithm was developed by Jack E. Bresenham in 1962 at IBM.

1. Highly efficient incremental method

2. Produces mathematically correct results using simple calculations

Bresenham's Line Drawing Algorithm for $m < 1$:

(1) Input the two line endpoints & store the left end point in $(x0, y0)$.

(2) Load $(x0, y0)$ into frame buffer that is plot the first point.

(3) Calculate constants Δx, Δy, $2\Delta y$ and $2\Delta y - 2\Delta x$ and obtain the

starting value for the decision parameter as : $P0 = 2\Delta y - \Delta x$.

(4) At each xk along the line starting at $k = 0$, perform the followingtest if $Pk < 0$ the next point to plot is $(xk+1 , yk)$ and $Pk+1 = Pk + 2\Delta y$

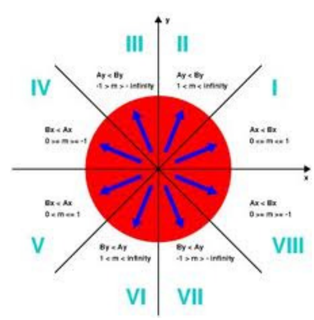
Otherwise the next point to plot is $(xk+1 , yk+1)$ and $Pk+1 = Pk + 2\Delta y - 2\Delta x$.

(5) Repeat step 4 Δx times.

---

6. **<u>Scan Converting Circles</u>**

circle is a symmetrical figure , eight points can be plotted for each value that the algorithm calculates



A circle is a set of points that are at a given distance r form the center position (xc, yc). This distance relationship is given as :

$(x - xc)2 + (y - yc)2 - r2 = 0$ This equation is used to calculate the position of points along the circle path by moving in the x direction from (xc - r) to (xc + r) and determining the corresponding y values as :

y = yc 2 2 c (x - x) – r

Algorithm :

(1) Input radius r and circle center (xc, yc) and obtain the first point on

circumference of a circle centered on origin (x0, y0) = (0, r)

(2) Calculate the initial value of the decision parameter as : P0 =5/4- r

(3) At each xk position, starting at k = 0 if Pk < 0 the next point alongthe circle is (xk+1, yk) and Pk+1 = Pk + 2xk+1 + 1, otherwise the next point along the circle is (xk + 1, yk - 1) and Pk+1 = Pk + 2xk+1 + 1 –

2yk+1 where 2xk+1 = 2xk + 2 & 2yk+1 = 2yk – 2.

(4) Determine symmetry points in other seven octants.

(5) Move each calculated pixel position (x, y) onto the circular path centered on (xc, yc) & plot coordinate values x = x + xc & y = y + yc.

(6) Repeat step (3) through (5) until x ≥ y.

---

7. <u>Scan Converting Ellipses</u>

An ellipse is defined as a set of points such that the sum of distances from two fixed points (foci) is same for all points given a point P = (x, y), distances are d_1 & d_2, equation is :

$d_1 + d_2 = \text{constant}$

In terms of local coordinates

$F_1 = (x_1, y_1)$ & F_2 (x_2, y_2)

---

8. <u>Display Technologies</u>

In some graphics systems a separate processor is used to interpret the commands in the display file. Such a

Raster Display System:

- **Interactive raster graphics employs several processing units**
- **Apart from CPU, a special purpose processor called video controller or display controller -> control operation of the display device**

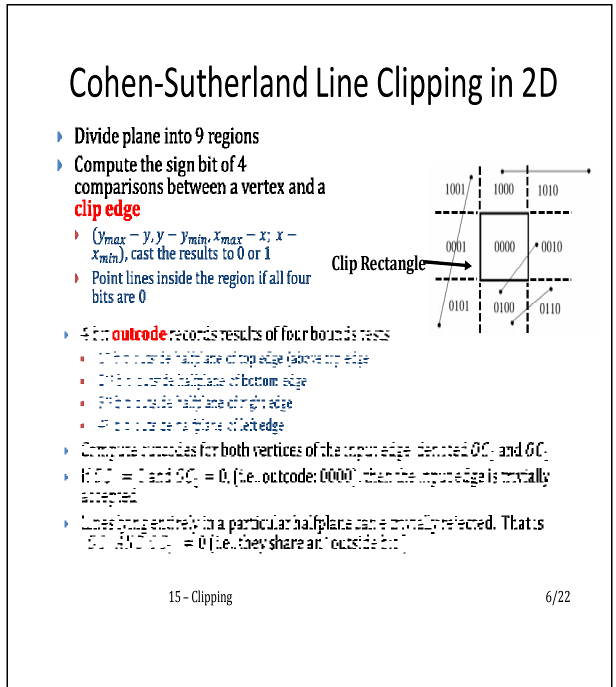
- **Simple raster graphics system**

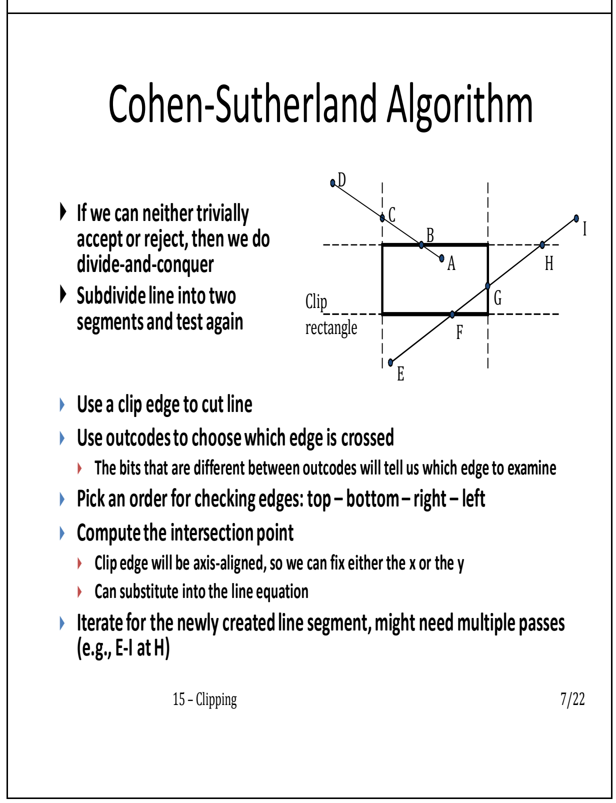
Random Scan Display:

- The random scan display system with display processor.
- It except the frame buffer.
- In random scan display no local memory is provided for scan conversion algorithm.

Video Controller:

- **Fixed area of system memory reserved for frame buffer**
- **Video controller given direct access to frame buffer to refresh the screen**
- **Coordinator origin is defined at lower left corner**
- **Scan lines labeled from y_{max} at the top of the screen to 0 at the bottom**





9. <u>**Cohen- Sutherland Algorithm**</u> ***(PPT)***

10. **Cyrus-Beck Algorithm** (PPT)

Cyrus-Beck / Liang-Barsky Parametric Line Clipping

- Use the parametric line formulation:
  - $P(t) = P_0 + t(P_1 - P_0)$
- Determine if the line intersects with a clip line (both extended to infinity)
- Decide if the intersection actually occurs on the polygon
- This is a very similar strategy for intersection tests in ray-tracing

15 – Clipping 11/22

---

11. Image Scanners

Image Processing is any form of signal, processing for which Input is an Image such as photographs or frames of video, the output of Image processing can be either an Image or a set of characteristics or parameters related to the Image.

UNI_II

---

1. <u>2D Transformations</u>

Moving of an object to one place in Window area to another place is called a Transformation.

Transformation is to change the′ object's

- Position (translation)
- Size (scaling)
- Orientation
- rotation)
- Shapes (shear)

Rotation:

- Rotation is applied to an object by repositioning it along a circular path in the XY plane

- Positive values of theta for counter clockwise rotation
- Negative values of theta for Clockwise rotation
- To generate a rotation , we specify

  Rotation angle theta

  Pivot point (Xr,Yr)

Scaling:

- Scaling alters the size of an object .
- Uniform scaling means this scalar is the same for all components.
- Non –Uniform scaling different per component
- Operation can be carried out by multiplying each of it component by a scalar

Reflection: A reflection is a transformation that produces a mirror image of an object

- Reflection along x axis
- Reflection along y axis
- Reflection relative to an axis perpendicular to the xy plane and passing through the coordinate origin
- Reflection of an object relative to

Shearing: A transformation that distorts the shape of an object such that the transformed object appears as if the object were composed of internal layers that had been caused to slide over each other.

PPT

2D Transformatio

- Translation

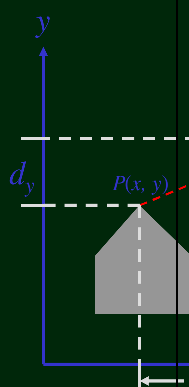
$P(x, y)$ move to $P'(x', y')$

$x' = x + d_x$

$y' = y + d_y$

$P = \begin{bmatrix} x \\ y \end{bmatrix}, \; P' = \begin{bmatrix} x' \\ y' \end{bmatrix}, \; T = \begin{bmatrix} d_x \\ d_y \end{bmatrix}$

$P' = P + T(d_x, d_y)$



2D Transformation

- Scaling

Scale by s_x along the x axis

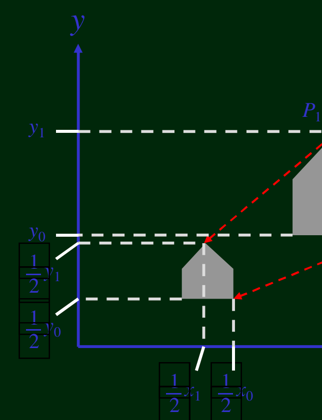
and by s_y along the y axis

$x' = s_x \cdot x$

$y' = s_y \cdot y$

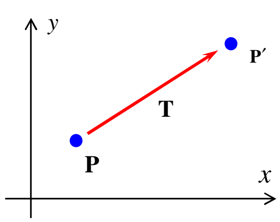
$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$

$P' = S(s_x, s_y) \cdot P$



41

2D Translation



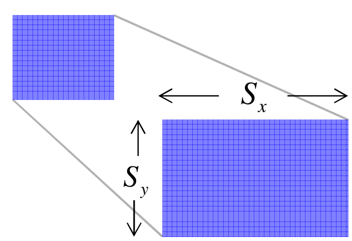
$x' = x + t_x, \quad y' = y + t_y$

$\mathbf{P} = \begin{bmatrix} x \\ y \end{bmatrix}, \; \mathbf{P}' = \begin{bmatrix} x' \\ y' \end{bmatrix}, \; \mathbf{T} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$

$\mathbf{P}' = \mathbf{P} + \mathbf{T}$

April 2010

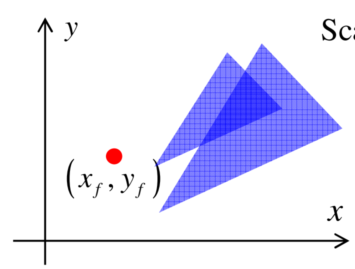
2D Scaling



$x' = x \cdot s_x, \quad y' =$

$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$

$\mathbf{P}' = \mathbf{S} \cdot \mathbf{P}$



Scaling about a fixed point $($

$x' = x \cdot s_x + x_f (1$

$y' = y \cdot s_y + y_f (1$

$\mathbf{P}' = \mathbf{P} \cdot \mathbf{S} + \mathbf{P}_f \cdot ($

April 2010

तेजस्वि नावधीतमस्तु

ISO 9001:2008 & 14001:2004

FAIRFIELD

Institute of Management & Technology

Managed by 'The Fairfield Foundation'

(Affiliated to GGSIP University, New Delhi)

2D Transformatio

- Rotation

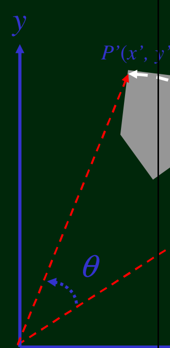
Rotate through an angle θ about the origin

$$x' = x \cdot \cos\theta - y \cdot \sin\theta$$

$$y' = x \cdot \sin\theta + y \cdot \cos\theta$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

$$P' = R(\theta) \cdot P$$



2D Transformation

- Derivation of the rotation equation

$$x = r \cdot \cos\phi$$

$$y = r \cdot \sin\phi$$

$$x' = r \cdot \cos(\phi + \theta)$$

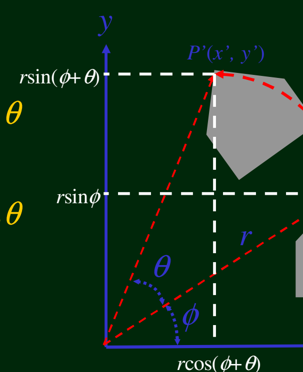
$$= r \cdot \cos\phi \cdot \cos\theta - r \cdot \sin\phi \cdot \sin\theta$$

$$y' = r \cdot \sin(\phi + \theta)$$

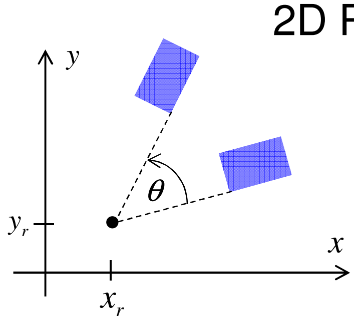
$$= r \cdot \cos\phi \cdot \sin\theta + r \cdot \sin\phi \cdot \cos\theta$$

$$x' = x \cdot \cos\theta - y \cdot \sin\theta$$

$$y' = x \cdot \sin\theta + y \cdot \cos\theta$$

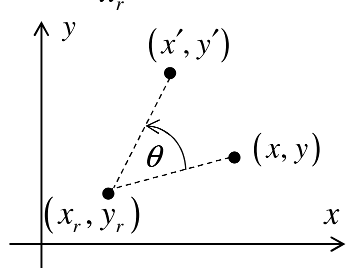


2D Rotation



Rotation in

pivot (rotat



$$x' = x_r + (x - x_r)c$$

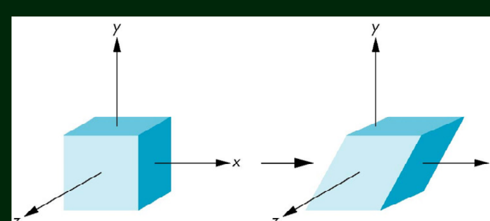
$$y' = y_r + (x - x_r)s$$

$$\mathbf{P}' = \mathbf{P}_r + \mathbf{R} \cdot (\mathbf{P} -$$

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

Shear

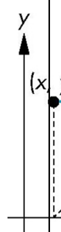
- Helpful to add one more basic transformat
- Equivalent to pulling faces in opposite dire



Shear Matrix

Consider simple shear along x a

$x' = x + y \cot\theta$
$y' = y$
$z' = z$



$$\mathbf{H}(\theta) = \begin{bmatrix} 1 & \cot\theta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Angel: Interactive Computer Graphics 3E © Addison-Wesley 2002

Homogeneous Co

The general form of four dimensiona coordinates is

$\mathbf{p}=[x\ y\ z\ w]^T$

We return to a three dimensional po

$x \leftarrow x/w$

$y \leftarrow y/w$

$z \leftarrow z/w$

If $w=0$, the representation is that of a

Note that homogeneous coordinates three dimensions by lines through dimensions

Angel: Interactive Computer Graphics 3E © Addison-Wesley 2002

Homogeneous Co and Computer G

- Homogeneous coordinates computer graphics system
  - All standard transformations (translation, scaling) can be im matrix multiplications with 4 x
  - Hardware pipeline works with representations
  - For orthographic viewing, we for vectors and $w=1$ for points
  - For perspective we need a *pe*

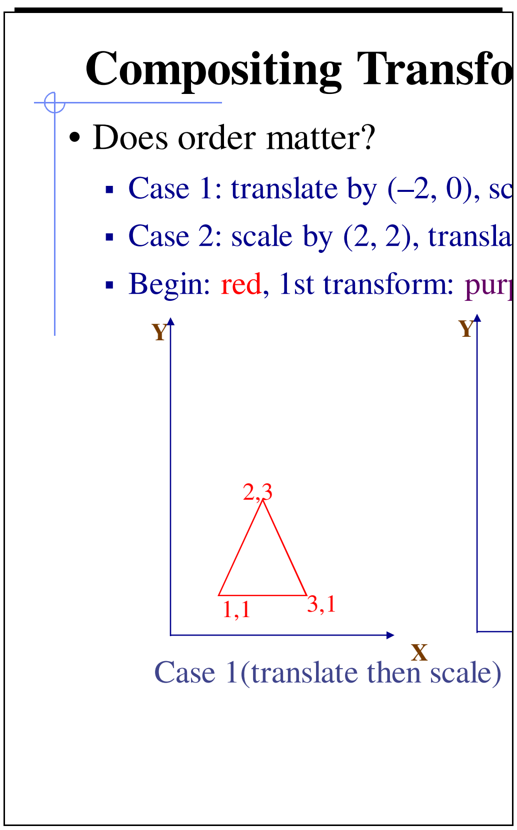
Angel: Interactive Computer Graphics 3E © Addison-Wesley 2002

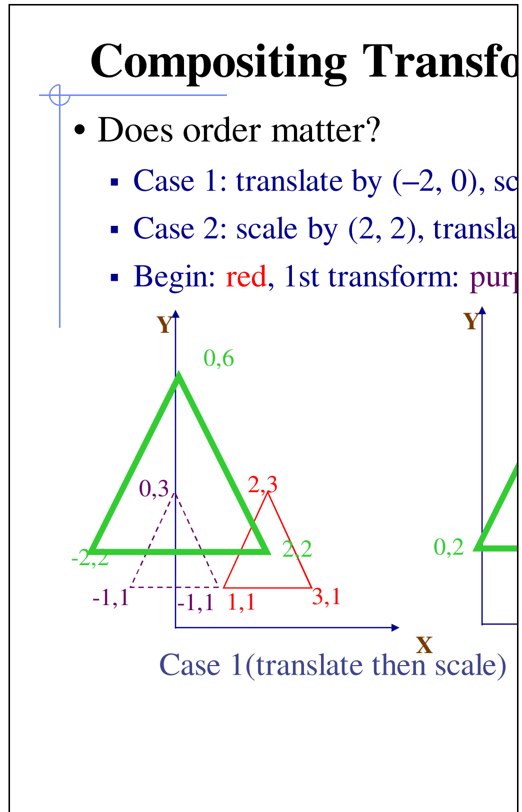
2. **Homogeneous Coordinates and Matrix Representation of 2D Transformations**,

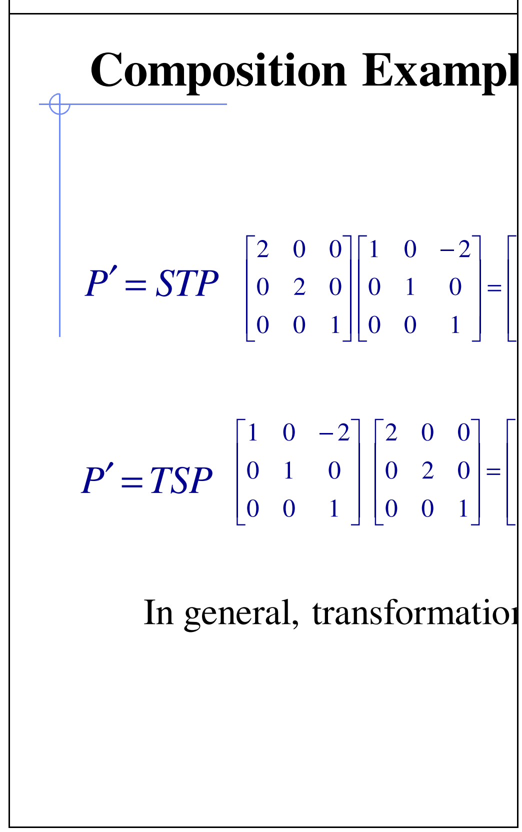
PPT

3. Composite Transformations

1. Sequence of composite transformation matrix and transformations could be setup by the matrix product of the individual transformations
2. Also as Concatenation or Composition of Matrices







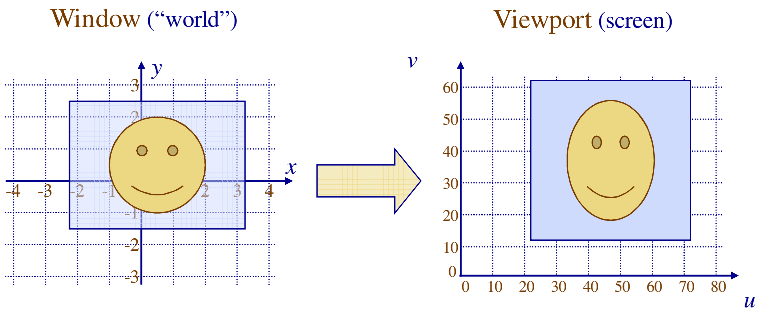
4. Window-to-Viewport Transformation

Window area on which object to be display and view port is generated in the window area and finally display in window area after selection of a particular object that should be consider as in view port.

Window-to-Viewport Transform

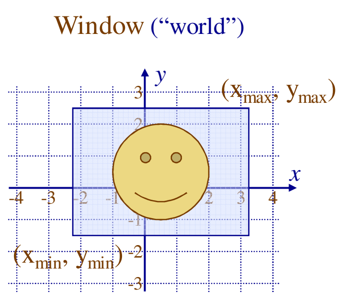
Need to transform points from "world" view (*window*) to the screen view (*viewport*)

- Maintain relative placement of points (usually)
- Can be done with a translate-scale-translate sequence



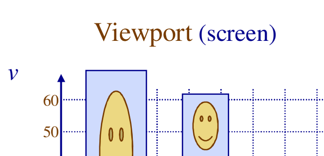
Window

- "Window" refers to the area in "world space" or "world coordinates" that you wish to project onto the screen
- Location, units, size, etc. are all determined by the application, and are convenient for that application
- Units could be inches, feet, meters, kilometers, light years, etc.
- The window is often centered around the origin, but need not be
- Specified as (x,y) coordinates



Viewport (cont)

- You can have multiple viewports
  - They can contain the same view of a window, different views of the same window, or different views of different windows



UNIT – III

Representing Curves & Surfaces

Polygon Meshes, Parametric Cubic Curves

Solid Modeling

Representing Solids, Regularized Boolean Set Operations, Primitive Instancing, Sweep Representations, Boundary Representations, Spatial Partitioning Representations, Constructive Solid Geometry, Comparison of Representations, User Interfaces for Solid Modeling.

1. polygon mesh

- it is a collection of vertices
- it is collection of edges and
- collection of faces
- it is a large sub-field of computer graphics and geometric modeling
- operations performed on meshes may include Boolean logic, smoothing, simplification, and many others

2. Parametric Cubic Curves

Curves and surfaces can have explicit, implicit, and parametric representations. Parametric representations are the most common in computer graphics.

- A parametric cubic curve is to be fitted to interpolate four points. The first and last points are to be at u=0 and u=1. The other two points are at u=1/3 and u=2/3, respectively. Find the equation of the curve in the form

$\mathbf{P(u)=U^{T}[M_{P}]B}$.

- The geometric matrix G of a parametric cubic curve defines a straight line
- A Bezier cubic curve obtained by a set of points.
- A set of control points explain what happen to a Bezier segment when two of the control points are coincident.

---

3. Solid Modeling

a) Primitive Instancing

- It is set of Primitive 2D/3D solid shapes
- Similar to parameterized object
- A family with few difference in members
- Relatively complex object
- Without combing object

b) Sweep Representations

- Sweeping a object in 2D and 3D
- Translational sweep
- Rotational sweep
- General sweep

c) Boundary Representations

- Object description in terms of vertices, faces and edges
- Some b-reps are restricted to planer , polygon etc...

d) Spatial Partitioning Representations

- A solid is decomposed into a collection of adjoin nonintersecting solids
- 1-Cell decomposition
- 2-spatial occupancy enumeration
- 3-Octress
- 4- Binary space –portioning trees
- Unambiguous but not necessary unique

e) Constructive Solid Geometry

- Operators at the internal nodes and easy primitive at the leaves
- Not Unique
- Deleting and adding replacing – modifying subtree etc..

f) Comparison of Representations

- Accuracy
- Domain
- Uniqueness
- Closure
- Compactness and efficiency

Reference:

Book Reference

- Himalaya publication: Computer Graphics, Sumit chahan
- D. Hearn & Baker: Computer Graphics with OpenGL, Pearson Education, Third Edition

Web Reference

1. http://www.slideshare.net/KRvEsL/solid-modeling
2. http://web.iitd.ac.in/~pmpandey/RP_html_pdf/assignment%20Parametric%20Cubic%20and%20Bezier%20Curve.pdf
3. http://www.bcanotes.com/Online/Computer%20Graphics/2D%20Transformation.html
4. **http://ecomputernotes.com/computer-graphics/two-dimensional-transformations/what-is-transformation-type-of-transformation**
5. http://askguru.net/d/2d-transformation-in-computer-graphics-ppt-download
6. *https://sites.google.com/site/assignmentssolved/system/app/pages/search?scope=search-site&q=Antialiasing*
7. *http://www.slideshare.net/KRvEsL/solid-modelinghttp://www.slideshare.net/KRvEsL/solid-modeling*
8. http://www.google.co.in/search?q=bresenham+line+drawing+algorithm&source=lnms&tbm=isch&sa=X&ei=

mnbSUYWsJYTkrAfvpIHwAQ&ved=0CAcQ_AUoA

Q&biw=1024&bih=587