08:12:2017

COMPUTER GRAPHICS

1. Survey of computer graphics :(or) Application of Graphics

* Graphics
* Animation
* Morphing

computer graphics is used in many fields education, medicine etc.

CAD : Computer Aided Design

It is used in designing building, models

Wireframe Displays :

We can see the internal parts using these wireframe displays.

Advantage - Interactive adjustment.
Personalized Symbols.
To test the performance of vehicle

Animations in CAD Application :

Animation - frame by frame

Animation used in Virtual Reality

- Virtual Reality Walk Through.

We can view the final view appearance.

2. PRESENTATION GRAPHICS :

* 35 mm slides
* It is used for financial, mathematical data and reports
* Ex : Bar chart, pie chart

3. COMPUTER ART :

* Fine Art and Commercial Art.

Fine Art – Paint Brush Programs.

* It is used to paint the pictures on the video monitor screen. Uses stylus.

Animation : Frequently used in advertisement

* Television , Commercial etc.
* Animation is produced by frame by frame.
* Each frame will be having motion and given positions.
* Successive frames differ by position.
* Each frame should be saved as image file.
* Film or video buffer. (save)
* For first time 24 frames Film animation requires for each second
* For play back 30 frames. on the video monitor.

4. ENTERTAINMENT ;

* Used in Movies, film
* Computer graphics is used in making motion pictures , television shows etc.

13:12:2017

5. Education and training:

It is mainly used for pilot trauing, flight stimulators

It helps to understand the performance in better way.

Input parameter - keyboards.

6. Visualization :

Large amount of data can be visualized to study the behaviour.

* Scientific Visualization

* Business Visualization commerce, industrial,

Effective visualization depends upon the characteristic of data.

Collection data has scalar values, vector values.

Visualization used by scientist and engineers.

Color Coding is one way to visualize the data set.

Image processing technique are combined with comp graphic visualization

7. Image Processing: 2M Difference between image processing and computer graphic.

Computer Graphic is used to create a picture whereas in image processing is used to process the image or picture (applying the technique to modify or interrupt the picture).

Application:

Improving the image quality.

Machine Perception. of visual information used in Robo.

Steps to be followed for image processing :

1. Image should be digitalized
2. Should apply the image processing technique

Purpose:

We can rearrange the parts of the picture, improve the colour, or to improve the quality of the picture.

Ex: Satellite picture undergo image processing

Tomography is used in medical field where X-ray is used to view the cross sectional view (X-ray Tomography).

CT - Computed Tomography.
PET - Position Emission Tomography.

Used in the field of nuclear fields, medicine etc.
surgery

// Artificial limbs : Combining Image Processing + Computer Graphic. //

8. Graphical User Interface:

Icon is a graphical symbol which is designed to use the application

Advantage:

Less screen space.

Easier interpretation.

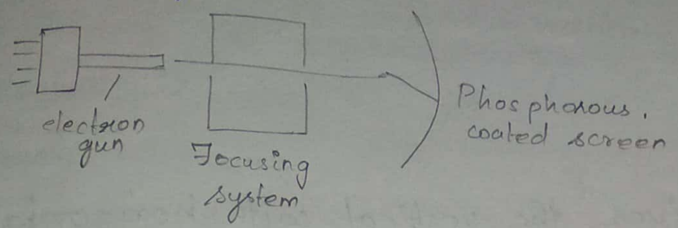
Window Manager - Allows to open multiple window.

Scanned by CamScanner

VIDEO DISPLAY DEVICE: 8m.

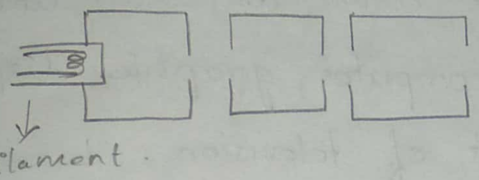
TV - Monitors - Picture Tube - CRT (Cathode Ray Tube).

Cathode Ray Tube :



2M Refresh CRT - keeps the phosphorous glowing to redraw the picture.

Components of Electron Gun :



1. heated metal cathode are heated through filament.

2. Control Grid.

* Control grid is used to control the brightness of the picture.

* Focusing system is needed to converge the electron.

* Electro static focusing system is used in television computer graphic control system.

Persistence : Even when the CRT is removen how long the phosphorous glows.

Low persistence phosphorous used to avoid flickering of picture.

High persistence phosphorous resistance used to display complex pictures.

RESOLUTION: 2m.

Maximum no. of pixels without overlapping displayed in the CRT is called resolution.

1280 x 1024

Aspect ratio:

This number gives the vertical with horizontal. 3/4.

RASTER SCAN DISPLAY

Rows in display screen are called as scan line.
Most common form of computer graphic display.
Based on the ~~cap~~ concept of television.
It has illuminated spots.
The picture definition is stored in refresh buffer or storage buffer.

Each screen point (pixel) - picture element in raster scan is used to give realistic display

Example of raster scan display: TV

One bit per pixel: 0 - ~~OFF~~
1 - ON

2M
One bit per pixel in black & white ~~TV~~ system is called bit map.

2M
Multiple bit per pixel in black and white system is called pix map.

Scanned by CamScanner

While scanning, returning to the left of the line while refreshing horizontal replace.

Vertical replace. - returing to next scan line after refresh is called vertical replace.

Interlace and refresh procedure - If beam is crossing each scan line, if there is any left out in second pass (so that every line is scanned is called interface and refresh procedure).

It is effective technique to avoid flicker.

Random Scan Display (or) Vector display (or) Stroke Writting (or) Calligraphic Display :

Electron will be directed to the screen where the picture is to be drawn.

Components of random scan display.

Ex: Pen plotter.

Refresh Display file - Set of line drawing in refresh area.

2m. Difference between raster scan and random scan.

Colour CRT monitor:

Deep penetration method

shadow Masking method.

24 bits per pixel.

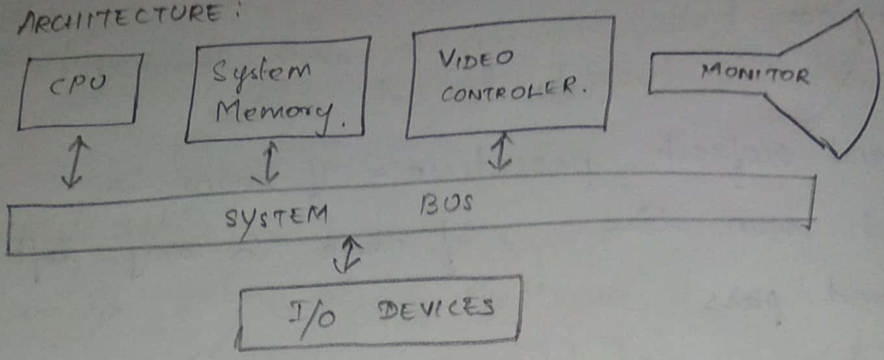
DVST - Direct View Storage Tube.

Non-emissive - LCD.

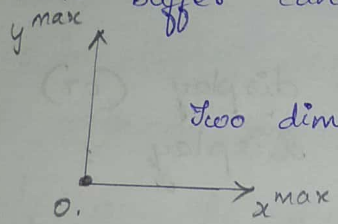
18/12/2017

Raster Scan System : 4m

ARCHITECTURE :



* Video controler - display controler. - to control display device
* Frame buffer can be anywhere in the system memory.



Two dimension Scan line :

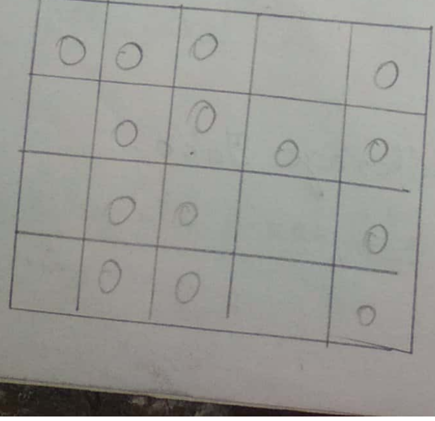
y max to 0.

0 to x max.

* Two register is used : X register is said to 0 y register is set to y max .
* X register is incremented by 1 when we measure next pixel on the scan line. On reaching the last line again x register is said to 0 and y - register is decremented by 1.

Cross Mechanism for real time animation :

Character defined as rectangular grid of pixel position



Or Curve section.

B

Raster Scan Display Processor (or) Graphic controller (or) Display Core Processor.

To free CPU from graphic

2M / A major task of the display processor is digitizing the picture definition in a given program into a pixel intensity value for storage in the frame buffer. This digitalization process is called scan conversion. //

Line styles:

Solid ________

Dotted .....

Dash - - - -

In an effort to reduce the memory requirements, methods used are link list and encoding.
One way to do this is to store each scan line as a set of integer pairs. One number of each pair intensity value and second number specifies no. of adjacent pixels of the scan line that are to have the intensity. This technique is called
2M RUN-LENGTH ENCODING. It is used to reduce storage space. It is used to construct pictures of long runs of single color each. similar

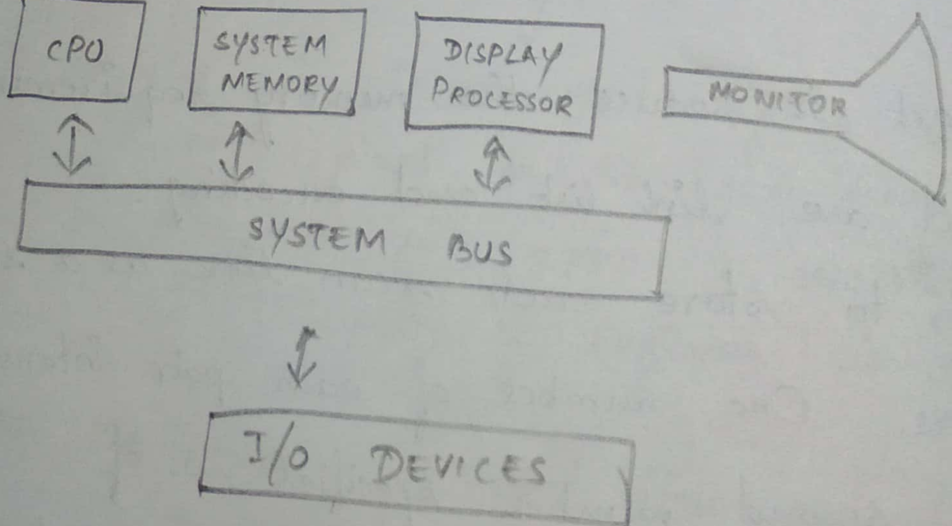
approach can be taken if the pixel intensity changes linearly.

Another approach is to encode ~~to~~ raster as a set of rectangular area is called CELL ENCODING.

Disadvantage of run-length encoding :

Intensity changes are difficult, to make and storage requirements actually increase as the length of runs decreases. It is difficult for the ~~disp~~ display controller to process the raster when many short runs are involved.

Random Scan System : 4m



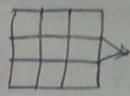
Display processing unit or graphic controller. It is used to process the output and displays it

20:12:2017.

WORKING OF RANDOM SCAN SYSTEM:

Application program → Input is stored in system memory along with the graphic package → Display processor processes the graphic command → Displays the out on the monitor.

Directing the electron beam along the lines of the picture.

Mediawall - Multiscreen system.  Mullions.

Used in passenger terminal, museums etc.

INPUT DEVICES: 8m. (no. diagram needed).

Keyboard - Alphanumeric keyboard - Used for non-graphical data - Cursor control keys - Function keys.

Ergonomic - User should feel comfortable.
Ergonomic designed keyboards were developed.

Mouse - Handheld device. → Controlling the cursor movement

Z-mouse will be used in virtual reality - CADD - Animation.
Top - standard ball.
bottom - Mouse ball

Track ball - It will move, operated using the palm - 2D

Space ball - It will not move. but operates same as track ball. - 3D.

Strain Guage - It is used to measure the pressure applied.

Joystick: Used to control the screen cursor.

Isometric Joystick: Pressure sensitive Joystick.

Data glove : Grasping virtual objects - Used to detect the movement of finger and hand position.
It has reciever and transmitting antenna's.
Consist of sensors.

Digitizers : Common devices used for drawing, painting , 2D - input , 3D - input . One type of digitizer ~~stylus~~ graphic Tablet . (Stylus based)

Types : Electro magnetic pulses
Acoustic Tablets.

Image Scanners :

Optical Scanning Mechanism → Text are scanned and stored in devices.

Touch Pannel :

Allows to display object using finger touch. Mainly used for selecting processing option. Input can be recorded by

* Optical - LED - Light detectors
* Electrical - Transparent plates } 2 plates < conducting material / resistance material.
* Acoustical - Sound waves will be generated. along the glass plates.

Waves will be emitted from the fingers to display. Transmission of each wave and ~~Recieved~~ reflection to the emitter.

Light Pens : It is also used to select the screen positions. (Pencil shape) Based on the lighting, reading will be different.

Disadvantage : Half screen only can be seen.
Prolong usage results in hand pain.

VOICE SYSTEM :

Storing the common words in the system. Advantage is there is no need of input commands, does not requires much attention.

It requires dictionary, voice recognizing pattern, microphone, headphones.

HARDCOPY DEVICES : 8m

PRINTER :

Pictures on the paper can be printed by plotter or printer.

Dot size - No. of dots per inch. or lines per inch.

Printers produce output either by impact method or non impact method.

Type writter
Ex : Line Printer.

by passing light
Ex : Plotters

Non-impact method uses laser technique, ink Jet spray, Xerographic process.

Electrostatic Method and electro thermal methods are also used by non-impact printers

(* all methods should be explained in detail)

Impact printer - color output is limited.

Non-impact printer uses three color pigments (cyan, magenta, yellow.)

Graphic layouts will be generated by pen plotters.

GRAPHIC SOFTWARE :

General programming Packages

Special purpose application packages.

General purpose programming packages → high level languages → It will be using GL

GL - Graphic library.

Basic functions - Straight line, color, polygon, circle intensity transformation. (To generate picture components) selecting views.

Application Graphic image designed for users who are not computer professionals or programmers.

Ex: CADD app, medical app.

User should be able to use the s/w in his ~~my~~ own forms.

Co-ordinate representation : 2M

Cartesian co-ordinate : x-y co-ordinates.

We can construct the shape of individual objects, ~~tree~~ such as trees, furnitures in

a scene within
a single frame called modelling co-ordinates local co-ordinates or master co-ordinates.

Once individual objects shape have been specified we can place the object into appropriate position within the scene using reference frame called world co-ordinate. Finally the world co.ordinate description of the scene is transferred to one - or more output device reference frames for displaying. These display co-ordinate systems are referred to as device co-ordinates or screen coordinates incase of video monitor. Graphic system first converts world co-ordinates position to normalized device co-ordinates in the range 0 to 1. Before final conversions to specific device coordinates, this makes the system independant of various devices that might be used at a particular work station.

$$(xmc : ymc) \Rightarrow (xwc : ywc) \Rightarrow (xnc , ymc) \Rightarrow (xdc : ydc).$$

Basic building blocks for pictures are referred to as output primitives. They include character string, geometric entities such as points, straight line, curved lines, filled areas (polygon, circle) and shapes defined with array of color points. Routines for output primitives provide the basic tool for constructing pictures.

22:12:2017 OUTPUT PRIMITIVES:

Attributes are the properties of the output primitives. An attributes ~~displays~~ describes how a particular primitive should be displayed. They include intensity, colour specifications, line styles. text styles and area ~~building~~ filling patterns.

We change the size, position or orientation of an object within a scene using geometric transformation. Viewing transformation are used to specify the view (i.e) to be presented and the portion of output displayed area. that is to be used.

Pictures can be subdivided into component parts subject structure. or segments or objects.

Input function are used to control and process the data flow from the interactive devices.

Software Standards:

Primary goal of standaridized software is portability. When packages are designed with std graphic functions, software can be easily moved from one ~~organization~~ hardware system to another. and used in different implementations and Applications. Without standards programs designed for one hardware system often cannot be transferred to another system. without extensive rewriting of the program.

GRAPHIC STANDARDS:

Graphical kernal System - This is the first graphic software standard adopted by ISO (International Standard Organisation) and ANSI (American National Standard Institute).

GKS was originally designed as 2D graphic package. GKS+ is developed for 3D graphic package.

Programmers Hierarchical Interactive Graphic Standards (PHIGS): it is an extension of GKS, and PHIGS+ is an extension of GKS+.

There is no standard methodology for displaying output.

Interface Output device Standards:

Standardization for device interface method is given in computer graphic interface (CGI).

CGM (Computer Graphic Metaface) System specifies standard for archiving and transporting pictures

WORK STATION:

Work station refers to a computer system with a combination of input and output devices which is designed for a single use

POINTS & LINES:

OUTPUT PRIMITIVES:

Graphic programming packages provides function to describe a scene in terms of basic geometric structures referred to as output primitives.

Points and straight line segments are the simplest geometric components of pictures. Additional output primitives that can be used to construct a picture can include circles and other ~~coding~~ conic sections, quadric surfaces, spline curves and surfaces, polygon color areas and character strings.

Point plotting is accomplished by converting a single co-ordinate position furnished by ~~single~~ an application program into appropriate operations for the output device in use. With a CRT monitor for example: Electron beam is turned on to illuminate the screen phosphorous at the selected location. how the electron beam is positioned depends upon the display technology.

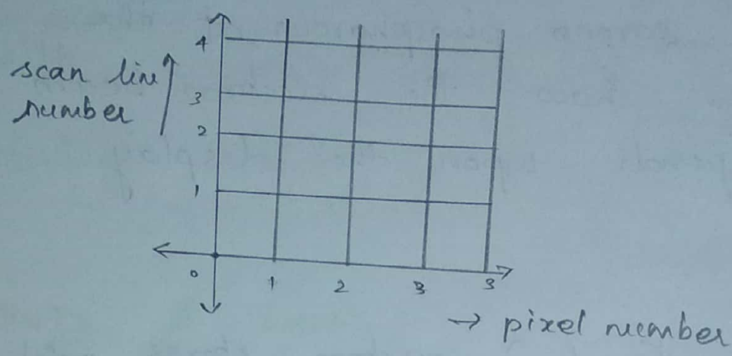
Random scan (vector) system stores point-plotting instructions in the display list and co-ordinate values in these instructions are converted to deflection voltages that position the electron beam at a screen location to be plotted during each refresh cycle.

For a black & white Raster system, a point is plotted by setting the bit value corresponding to a specified screen position within the screen buffer 2-1. Then as the electron beam swipes across each horizontal scan line, it emits a brusts of electrons (plots a point) whenever a value of one is encountered in the frame buffer.

26 : 12 : 2017

LINE DRAWING :

Calculating the intermediate (points) to draw a line path. For analog device such as pen, plotter to draw the lines.



Line drawing algorithm :

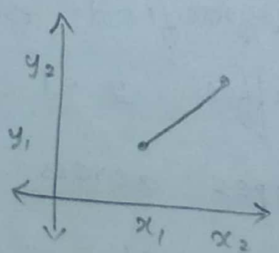
Equation of straight line ⇒ $y = mx + b$

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

$$b = y - mx$$

$$\boxed{\Delta y = m \Delta x.}$$

Line path between end point position



DDA ALGORITHM

V. IMP Algorithm is must.

(Digital Differential Analyzer) :

It is scan line conversion algorithm based on calculating Δx or Δy

aw

$$\Delta y = m\,\Delta x.$$

$$\Delta x = \frac{\Delta y}{m}$$

For lines with +ve slope:

$\Delta x = 1 \Rightarrow \Delta y = m$

$$y_{k+1} - y_k = m.$$

$$\boxed{y_{k+1} = y_k + m} \rightarrow (1)$$

k can take values from 1 to until reaches the end point. m takes +ve integer between 0 and 1.

For lines with +ve slope more than 1:

$$\Delta y = 1$$

$$\Delta x = \frac{1}{m}$$

$$\boxed{x_{k+1} = x_k + 1/m} \rightarrow (2)$$

Based on equation (1) & (2)

the lines are represented from left endpoint to right end point.

If the process are reversed,

$$\Delta x = -1$$

$$\Delta y = -m$$

$$y_{k+1} - y_k = -m$$

$$\boxed{y_{k+1} = y_k - m} \rightarrow (3)$$

Scanned by CamScanner

when $\Delta y = -1$

$$\Delta x = -1/m$$

$$\boxed{x_{k+1} = x_k - 1/m} \rightarrow (4)$$

Procedure line DDA (xa, ya, xb, yb ; integer):

Var dx, dy, steps, k : integer;

⋮

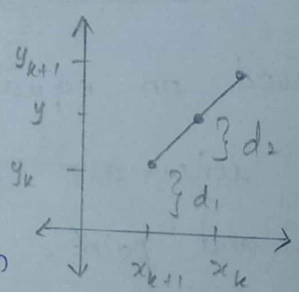
end line DDA //

DDA algorithm is mainly used for calculating the pixel position, Rounding off error successive addition, floating point is a problem.

BRESENHAM ALGORITHM: 8M. * IMP.

It is accurate and efficient algorithm for drawing raster line.

$x_k + 1$



Considering the scan line conversion process,

$$y = m(x_k + 1) + b \rightarrow (1)$$

$$d_1 = y - y_k$$

$$d_1 = m(x_k + 1) + b - y_k$$

$$d_2 = (y_{k+1}) - y$$

$$d_2 = y_k + 1 - m(x_k + 1) - b$$

$$d_1 - d_2 = 2m(x_k+1) - 2y_k + 2b - 1 \rightarrow (2)$$

$$m = \frac{\Delta y}{\Delta x}$$

$$P_k = \Delta x (d_1 - d_2)$$

$$P_k = 2\Delta y . x_k - 2\Delta x . y_k + c \rightarrow (3).$$

$$\Delta x (2m(x_k+1) - 2y_k + 2b - 1)$$

$$= \Delta x 2m x_k + 2m\Delta x - 2y_k \Delta x + 2b\Delta x - \Delta x$$

$$= \Delta x \times \frac{\Delta y}{\Delta x} 2x_k + 2\frac{\Delta y}{\Delta x}\Delta x - 2y_k \Delta x + 2b\Delta x - \Delta x$$

$$= 2\Delta y x_k + 2\Delta y - 2\Delta x y_k + 2b\Delta x - \Delta x$$

$$= 2\Delta y x_k - 2\Delta x y_k + c$$

$$\boxed{c = 2\Delta y + 2b\Delta x - \Delta x}$$

$$P_{k+1} = 2\Delta y x_{k+1} - 2\Delta x y_{k+1} + c$$

$$P_{k+1} - P_k = 2\Delta y (x_{k+1} - x_k) - 2\Delta x (y_{k+1} - y_k)$$

$$= 2\Delta y (x_k + 1 - x_k) - 2\Delta x (y_{k+1} - y_k)$$

since $x_{k+1} = x_k + 1$

$$P_{k+1} - P_k = 2\Delta y - 2\Delta x (y_{k+1} - y_k)$$

$$P_{k+1} = P_k + 2\Delta y - 2\Delta x (y_{k+1} - y_k)$$

($y_{k+1} - y_k$ will be 0 or 1 based on parameter P_k).

Substitute (x_0, y_0) in eq (3)

$$P_0 = 2\Delta y - \Delta x$$

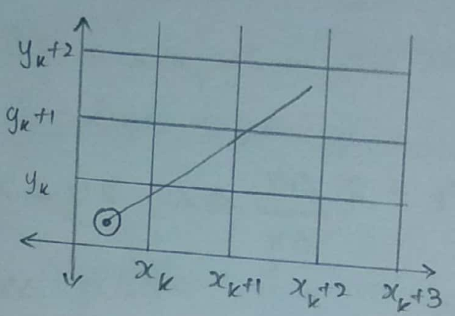
27:12:2017

$$P_k = \cancel{2\Delta y x_k} + 2\Delta y - \cancel{2\Delta x y_k} + 2\Delta x b - \Delta x$$ constant (so left out).

Put (x_0, y_0)

$$\boxed{P_0 = 2\Delta y - \Delta x} \leftarrow (x_k+1, y_k).$$

$$P_k+1 = P_k + 2\Delta y$$



It is used for display circles and other curves.

Breshenham Algorithm must

To Illustrate the algorithm line with end points (20, 10) (30, 18), slope = 0.8 with $\Delta x = 10$, $\Delta y = 8$. Find initial decision parameter P_0, k, P_k (x_k+1, y_k+1).

$P_0 = 2\Delta y - \Delta x.$

$P_0 = 2(8) - 10$

$P_0 = 16 - 10$

$P_0 = 6.$

k	P_k	(x_{k+1}, y_{k+1})
0	6	(21, 11)
1	2	(22, 12)
2	−2	(23, 12)
3	14	(24, 13).
4	10	(25, 14).

If $k = 0$, $P_k = 6$ $(\because P_0 = 2\Delta y - \Delta x)$.

$(x_{k+1}, y_{k+1}) = (20+1, 10+1)$

$(x_{k+1}, y_{k+1}) = (21, 11).$

If $k = 1$, $P_k = 2$ $(x_{k+1}, y_{k+1}) = (22, 12).$

$P_1 = 2.$

If $P_k < 0$, then $P_{k+1} = P_k + 2\Delta y \Rightarrow (x_k+1, y_k)$

If $P_k > 0$, then $P_{k+1} = P_k + 2\Delta y - 2\Delta x$.
(x_k+1, y_k+1)

since $P_0 = 6 > 0$, $P_{k+1} = P_k + 2\Delta y - 2\Delta x$
$= 2$.

since $P_1 = 2 > 0$ $\quad P_{k+1} = P_k + 2\Delta y - 2\Delta x$.
$= 2 + 2(8) - 2(10)$
$= 18 - 20$
$P_{k+1} = -2$.

since $P_2 < 0$, $\quad P_{k+1} = P_k + 2\Delta y$.
$= -2 + 2(8)$
$= -2 + 16$
$= 14$.

since $P_3 > 0$ $\quad P_{k+1} = P_k + 2\Delta y - 2\Delta x$.
$= 14 + 2(8) - 2(10)$
$= 14 + 16 - 20$.
$= 30 - 20$
$= 10$.

since $P_4 > 0$ $\quad P_{k+1} = P_k + 2\Delta y - 2\Delta x$.
$= 10 + 16 - 20$.
$= 26 - 20$
$= 6$.

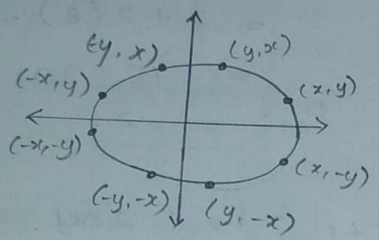
$k = 9$.

CIRCLE GENERATING ALGORITHM : 13 mark. V.IMP

Equation of circle is $(x-x_c)^2+(y-y_c)^2=r^2$
(cartesian co-ordinates).

Equation of circle : $x = x_c + r\cos\theta$ (in polar co-ordinates)
$y = y_c + r\sin\theta$.

Midpoint circle Algorithm :



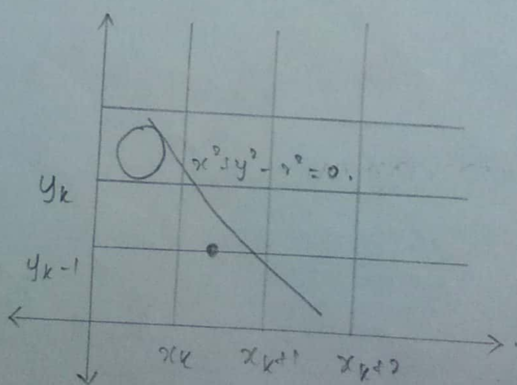
Symmetry of circle (diagram).

Circle equation, $f_{circle}(x,y) = x^2+y^2-r^2 \rightarrow (1)$

Any point (x,y) on the boundary of circle with radius r satisfies the equation,

$f_{circle}(x,y) = 0$.

$$f_{circle}(x,y) = \begin{cases} <0 & \text{if } (x,y) \text{ is inside the circle boundary} \\ =0 & \text{if } (x,y) \text{ is on the circle boundary} \\ >0 & \text{if } (x,y) \text{ is outside the circle boundary} \end{cases}$$



Mid point between candidate pixels and sampling position x_k+1 along a circular path.

Assuming that we have just ~~spotted~~ plotted the pixel at (x_k, y_k), we next need to determine whether the pixel at position (x_k+1, y_k) or one at the position (x_k+1, y_k-1) is closer to the circle.

Our decision parameter is the circle function evaluated at a midpoint between these two pixels.

Mid point of (x_k+1, y_k) and (x_k+1, y_k-1) is $(x_k+1, y_k - 1/2)$.

$$P_k = f_{circle}(x_k+1, y_k - 1/2).$$

$$f_{circle}(x_k+1, y_k-1/2) = (x_k+1)^2 + (y_k-1/2)^2 - r^2$$

$$x_k^2 + 2x_k + 1 + y_k^2 - y_k + 1/4 - r^2$$

If $P_k < 0$, this mid point is inside the circle. and the pixel on scan line, y_k is closer to the circle boundary. otherwise $(P_k \geq 0)$ ~~is~~ the midpoint is outside or on the circle boundary, we select the pixel on scan line $y_k - 1$.

We obtain a ~~regressive~~ recursive expression for the next decision parameter by evaluating the circle function at sampling position

$x_{k+1}+1 = x_k+2.$

$$P_{k+1} = f_{circle}\left(x_{k+1}+1,\ y_{k+1}-\tfrac{1}{2}\right)$$

$$P_{k+1} = (x_{k+1}+1)^2 + (y_{k+1}-\tfrac{1}{2})^2 - r^2.$$

$$P_{k+1} = (x_{k+1})^2 + 2x_{k+1}+1 + (y_{k+1})^2 - y_{k+1} + \tfrac{1}{4} - r^2$$

$$P_k = \overset{(-)}{-x_k^2} \overset{(-)}{+2x_k} \overset{(-)}{+1} + \overset{(+)}{y_k^2} \overset{(-)}{-y_k} \overset{(+)}{+\tfrac{1}{4}} - r^2$$

$$P_{k+1}-P_k = 2(x_k+1) + (y_{k+1}^2 - y_k^2) - (y_{k+1}-y_k)$$

$$(x_k+1)^2 + 2x_k+1 - x_k^2 - 2x_k = \cancel{x_k^2}+2\cancel{x_k}+1+2x_k+1-\cancel{x_k^2}-2\cancel{x_k}$$

$$= 2x_k+2 = 2(x_k+1)$$

$$P_{k+1} = P_k + 2(x_k+1) + (y_{k+1}^2 - y_k^2) - (y_{k+1}-y_k)$$

Increments for obtaining P_{k+1} are either $(2x_{k+1}+1)$ if $P_k < 0$ (or) $2x_{k+1}+1-2y_{k+1}$. Evaluation of terms $2x_{k+1}$, $2y_{k+1}$ can also be done incrementally $2x_{k+1} = 2x_k+2$. At start position $(0, r)$ these two terms have values 0 and $2r$ respectively. Each successive value is obtained by adding 2 to the previous value of $2x$ and substracting 2 from the previous value of $2y$.

$$2x_{k+1} = 2x_k+2$$

$$2y_{k+1} = 2y_k - 2.$$

Initial decision parameter is obtained by evaluating the circle function at the start position $(x_0, y_0) = (0, r)$.

$P_k = f_{circle}(x_k + 1, y_k - 1/2)$.

At $(0, r)$

$$P_0 = f_{circle}(1, r - 1/2)$$

$$P_0 = 1 + (r - 1/2)^2 - r^2$$

$$= 1 + \cancel{r^2} - r + \frac{1}{4} - \cancel{r^2}$$

$$= 5/4 - r.$$

$$P_0 = \frac{5}{4} - r.$$

$$\boxed{P_0 = 1 - r.}$$

$$\frac{5}{4} \approx 1$$

Algorithm :

set Pixel.

⋮

end. //

03:01:2018

Radius $r = 10$, midpoint circle algorithm along the circle in the 1st quadrant $x = 0$, $x = y$.

The value of the decision parameter

$$P_0 = 1 - r$$

$$P_0 = 1 - 10$$

$$P_0 = -9$$

Then for the circle centered on coordinate origin, $(x_0, y_0) = (0, 10)$ and initial increment terms for calculating decision parameters are

$$2x_0 = 0 \qquad 2y_0 = 20.$$

Successive decision parameter values and the position along the circle path are calculated using the midpoint method as

k	P_k	(x_k+1, y_k+1)	$2x_k+1$	$2y_k+1$
0	-9	(1, 10)	2	20
1	-6	(2, 10)	4	20
2	-1	(3, 10)	6	20
.	.	.	.	.
.	.	.	.	.
6	5	(7, 7)	14	14

// not for theory only for lab

Ellipse generating Algorithm:

An ellipse is elongated circle, therefore eliptical curves can be generated by modifying circle drawing procedures to taken into account. Different dimensions of an ellipse along major and minor axes.

General equation

$$ax^2 + by^2 + cxy + dx + ey + f = 0.$$

where coefficients

a, b, c, d, e, f are evaluated in terms of focal coordinates and dimenscions of major and minor axis of the ellipse.

Major axis is the straight line segment extending from one side of the ellipse to the other through the foci.

Minor axis spans the shorter dimension of the ellipse, bisepting the major axis at the half wayposition (ellipse centre) between the two foci.

Ellipse function equation with $(x_c, y_c) = (0,0)$

$f_{ellipse}(x,y) = r_y^2 x^2 + r_x^2 y^2 - r_x^2 r_y^2$ which has the following properties

$f_{ellipse}(x,y) < 0$ if (x,y) (inside the ellipse boundary)

$f_{ellipse}(x,y) = 0$ if (x,y) is on the ellipse boundary

$f_{ellipse}(x,y) > 0$ if (x,y) is outside the ellipse boundary.

Filled Area Primitives :

A standard output primitive in general graphic package is solid color or patterned polygon area. Other kinds of area primitives are available but polygons are easier to process since they have linear boundaries.

2m There are two basic approaches to fill the area on raster system.

1. To fill an area is to determine the overlap intervals for the scan lines that cross the area.

2. Area filling is to start from given interior position and paint outward from this point until we encounter the specified boundary condition.

Scan line approach is used in general graphic packages like circle, polygon, ellipse and other simple curves

Fill method starting from interior point are useful with more complex boundaries and interactive painting system.

Scan line Polygon Fill Algorithm:

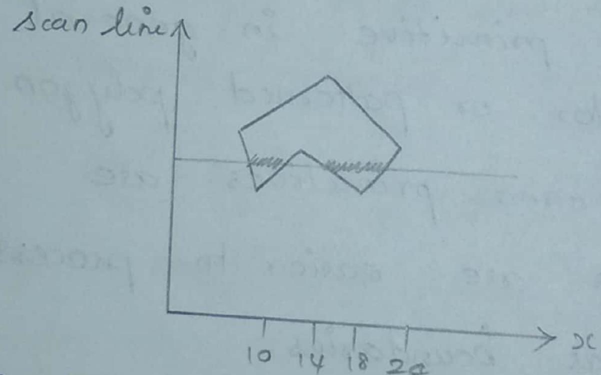
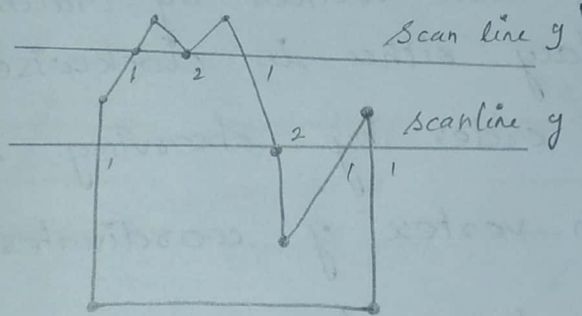


Fig: Interior pixels along a scan line passing through a polygon area.

The above figure illustrates the scanline procedure for solid filling of polygon areas. For each scan line ~~passing~~ crossing the polygon, area fill algorithm locates the intersecting points of the scanline with polygon edges. These intersection points are then sorted from left to right and corresponding frame buffer positions between the each intersection pair are ~~said~~ set to specify fill color.

In figue, the 4 ~~section~~ pixel intersection position with polygon boundaries define two stretches of interior pixels of $x = 10$ to $x = 14$ and from $x = 18$ to $x = 24$.

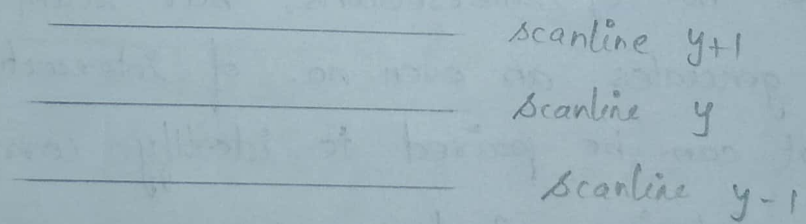


Intersection points along scanline that intersects polygon vertices scanline y generates an odd no. of intersections, but scan line y' generates an even no. of intersection that can be paired to identify correctly the interior pixel spans.

Fig-2 shows 2 scanline at position y and y' that intersect edge endpoints. Scanline y intersects 5 polygon edges. Scanline y' intersects an even no. of edges although it also passes through a vertex. Intersection points along scanline y' correctly identifies the interior pixel spans. But with scanline y we need to do some additional processing to correct interior points.

For scanline y two intersecting edges sharing a vertex are on opposite sides of the scanline. But for scanline y', two intersecting edges are both above the scanline. Thus the vertices that require additional processing are those that have connecting edges on the opposite sides of the scanline.

We can identify these vertices by tracing around the polygon boundary either in clockwise or counter clockwise order by observing the relative changes in vertex y - coordinates. As we move from edge to next.



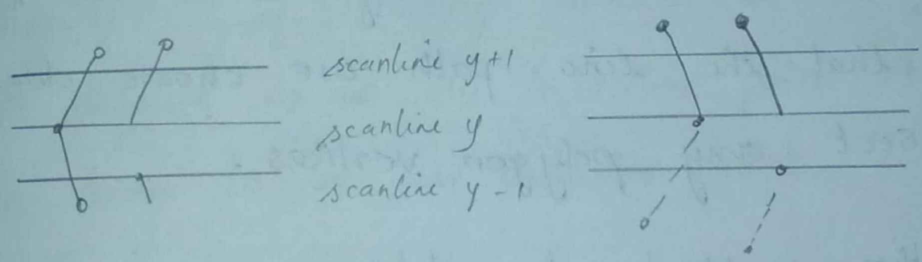
Coherence :

The properties of one part of a scene are related in some way to other parts of the scene so that relationship can be used to reduce processing.

Coherence method often involves incremental calculations applied along a single scan line or between successive scan line.

Note:

Shortning of an edge: when the ~~edge~~ end point y coordinates of two edges are increasing, the y-value of upper end point for the current edges decrease by 1. When end point y values are monotonically decreasing, we derease the y-coordinate of upper end point of the edge following the current edge.



INSIDE OUTSIDE TEST: 8m.

Polygon is defined as having no self intersection example of standard polygon include triangle, rectangle, octagons doccagons,. the component edges of these objects are joint only at the vertices and otherwise the edges have no common point in the plane. Identifying the interior regions of standard polygon is generally a straight forward process.

Graphic packages normally use odd-even rule or non-zero ~~binding rule~~ winding number rule to identify the interior regions of an object.

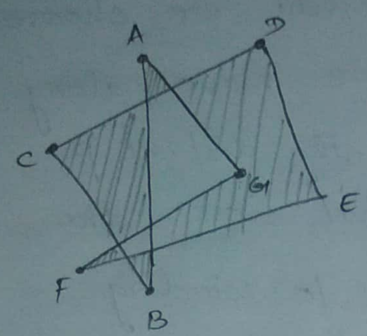
Odd-even rule also called odd-parity rule or even-odd rule. Drawing a line from any position p to a distant point outside the coordinate extends of the object and counting the no. of edge crossing along the line. If no. of polygon edges cross by this line is odd. then p is an interior point otherwise the p is an exterior point.

To obtain an accurate edge count we must be sure that the line path we choose does not intersect any polygon vertices.

Another method for defining interior regio is non-zero winding number rule which counts the no. of times, the polygon edges wind around a particular point in the counterclockwise direction. This count is called winding number. On the interior points of an 2D object are defined to be those that have a non-zero value for the winding number. We apply non-zero winding number rule to polygons by initializing the winding number to 0. The line we choose must not pass any vertices as we move along the line from position P from the distant P we count the no. of edges that cross the line in each direction. We ~~had~~ add 1 to winding number, everytime we intersect the polygon edge that crosses the line from right to left and we subtract one, everytime we intersect edge that crosses from left to right. Final value of winding number after all edge crossing have been counted determine the relative position of P. If winding number is non-zero then P is defined to be an interior point otherwise P is taken to be an exterior point.

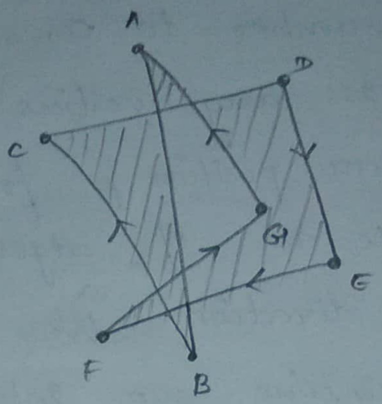
For standard polygons and other simple shapes, non-zero winding number rule and odd-even rule give the same results. But for more complicated shapes, ~~may~~ two methods may yield different interior and exterior regions.

08 : 01 : 2018



odd even rule

shaded - interior.



Non-zero winding number rule.

 – Elliptical arc interior fill

Scanline fill of curved boundary areas:

Scanline fill of regions with curved boundaries requires more work than polygon filling. Since intersection calculations involves non-linear boundaries for simple curves such as circles or ellipse performing a scanline fill is a straight forward process. We only need to calculate two scanline intersections on the opposite sides of the curves. This is same as generating pixel position along the curve boundary by using midpoint method we simply fill in by horizontal pixel span between the boundary points on the opposite sides of the curve.

Symmetries between ~~coordrants~~ ~~Octo~~ quadrants (between octants) ~~for~~ are used to reduce the boundary calculations.

Interior region is bounded by the ellipse section and the straight line segment that closes the curve by joining the begning and the ending positions of the arc.

BOUNDARY FILL ALGORITHM: 2m or 4m.

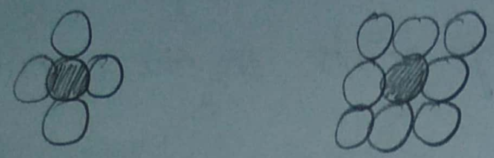
Paint the interior untill reach the boundary.

Another approach to area filling is to start at a point inside the region and paint the interior outward to the boundary.

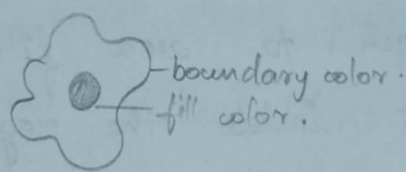
If the boundary is specified in a single color the fill algorithm proceeds outward pixel by pixel untill the boundary color is encountered. This method is called boundary fill algorithm. It is very useful in interactive painting packages. where interior points are easily selected.

Scanned by CamScanner

4 - connected 8 - connected.



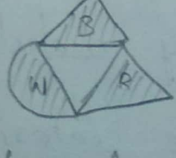
A boundary fill procedure accepts ~~the~~ as input the coordinates of an interior point (x, y), a fill color and a boundary color. Starting from (x, y) the procedure test neighbouring positions to determine whether they are of boundary color if not, they are painted with fill color. and their neighbours are tested. This process continues untill all pixels upto the boundary color for the area have been tested.



Color boundary for boundary fill procedure.

Flood Fill Algorithm: 2m or 4m.

We want to fill in ((or) recolor), the area that is not defined within a single color boundary



Multiple color boundary to represent a area.

This diagram shows the area ~~bordod~~ bordered by several different color region. We can paint such areas by replacing the specified interior color instead of searching for a boundary color value. This approach is called flood fill algorithm.

We start from a specified interior point (x, y) and reassign all pixel values that are currently said to a given interior color with a desired fill color. If the area we want to paint as more than one interior color we can first reassign the pixel values so that all interior points have the same color.

In this approach we stack only the begning positions for those pixel spans having the value old color. Starting at the first position of each span the pixel values are replaced until a value other than old color is encountered.

Character Generation :

Overall design style for a set (or family) of characters is called type phase. (or fonts).

Type phase is divided into 2 groups.

1). serif.
2.) sans

22:01:2018

2D - Geometric Graphic transformation : 8m or 13 m.

The basic geometric transformations are

* translation
* rotation
* scaling (Change in shape).

And other transformations are reflection and shear.

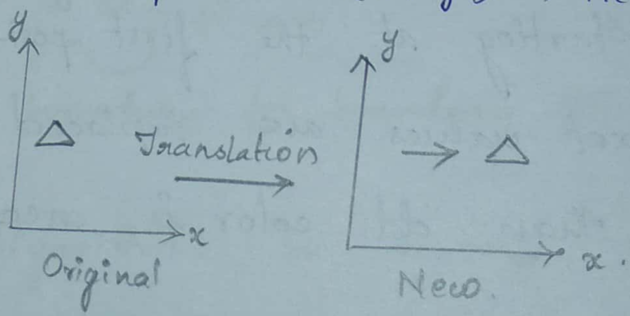
Translation :

Translation is applied to an object by repositioning it along a straight line path from one cordinate location to another.

Translation distance :

tx, ty

Original cordinate position (x, y), new position (x', y')



$$x' = x + tx.$$

$$y' = y + ty$$

The translation distance pair (tx, ty) is called translation vector or shift vector.

$$P = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$P' = \begin{bmatrix} x_1' \\ x_2' \end{bmatrix}$ $T = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$ $P = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$\boxed{P' = P + T.}$ – translation equation.

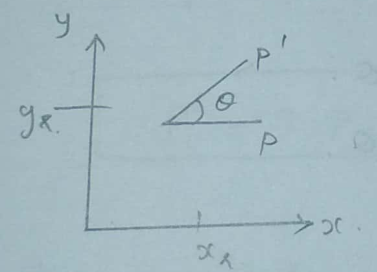
2D. translation equation in matrix form is $P' = P+T$.

Translation is a rigid body transformation that moves object without deformation. (i.e) Every point on the object is translated by the same amount.

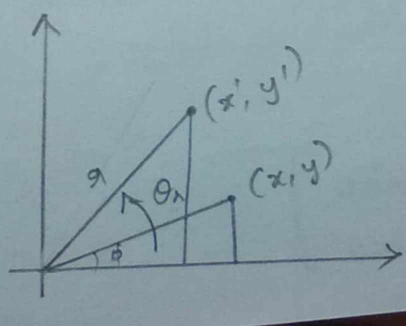
ear.

Rotation :

A 2D rotation is applied to an object by repositioning it along a circuit path in x, y plane. To generate a rotation we specify a rotation angle θ and position (x_r, y_r) of the rotation point or pivot point about which the object is to be rotated.



Positive values for the rotation angle defines the counter clockwise rotation about the pivot point.
And negative values rotates the object in clockwise direction.



r - constant distance of the point from the origin

ϕ - Original Angular position of the point from the horizontal.

θ - Rotational Angle.

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

$x' = r\cos\phi\cos\theta - r\sin\phi\sin\theta \rightarrow ①$

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

$y' = r\cos\phi\sin\theta + r\sin\phi\cos\theta \rightarrow ②.$

Original cordinates of the point in polar coordinate are

$x = r\cos\phi \; ; \; y = r\sin\phi \rightarrow ③.$

Using ③ in ① & ②

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

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

$\boxed{P' = R \cdot T}$ R - Rotation.

where

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

$$(P')^T = (R \cdot P)^T$$
$$= P^T \cdot R^T$$

where $R^T = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$

from

he

zontal.

adinates

For rotation matrix R transpose is obtained by simply changing the sign of sine term.

Rotations are rigid body transformations the moves the object without deformation. Every point on an object is rotated through the same angle.

Scaling:

Scaling transformation alters the size of an object.

$$x' = x \cdot S_x$$

$$y' = y \cdot S_y$$

S_x, S_y - Scaling Factor.

Scaling factor S_x scales the object in x direction and S_y scales the object in y-direction.

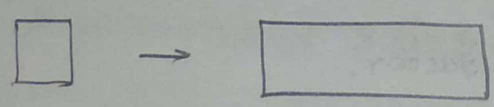
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$\boxed{P' = S \cdot P}$ where S is 2x2 scaling matrix.

Any positive numeric values can be assigned to scaling factors S_x, S_y. The values less than 1 reduces the size of object and values greater than 1 produce an enlargement. The value of 1 for both S_x and S_y leaves the size of object unchanged.

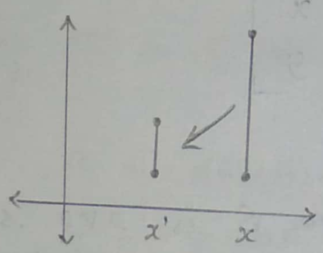
27:01:2018

When S_x and S_y are assigned the same value a uniform scaling is produced that maintains relative object proposition. Unequal values for S_x and S_y result in a differential scaling that is often used in design application where pictures are constructed from few basic shapes that can be adjusted by scaling and positioning transformation.



$S_x = 2$ $S_y = 1$ (differential scaling)

if $S_x = S_y$, it is called uniform scaling.



$S_x = S_y = 0.5$

A line is scaled with $p = S.p.$ using $S_x = S_y = 0.5$ is reduced in size and move closer to the coordinate origin.

We can control the location of scaled object by choosing a position called fixed point. (i.e) to remain unchanged after the scaling transformation.

$$x' = x_f + (x - x_f) S_x$$

$$y' = y_f + (y - y_f) S_y .$$ // 2D transformation

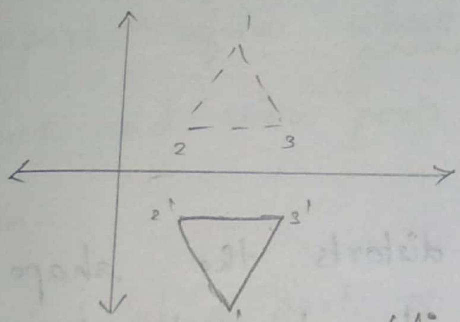
// not in syllabus
Composite transformation:

Forming product of transformation matrices is often referred to as concatenation or composition of matrices. /

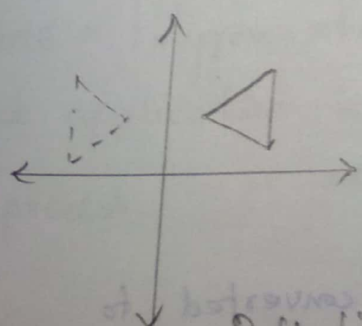
OTHER TRANSFORMATION:

REFLECTION:

Reflection is a transformation that produces a mirror image of an object. The mirror image for 2D



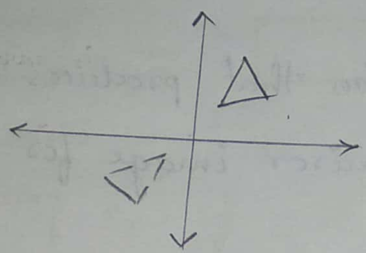
reflection is generated relative to an axis of generation. reflection by rotating the object 180° about the reflection axis. Reflection of an object about x-axis is shown in the above diagram.



Reflection of an object about y-axis.

Reflection about the line $y=0$, the x-axis is accomplished with the transformation matrix $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ this transformation keeps the x-value the same but flips the y-value of the coordinate position.

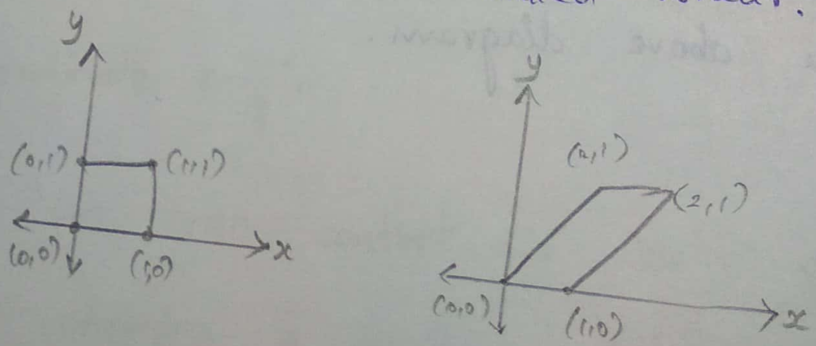
A reflection about y-axis flips x-coordinate while keeping the y-coordinate the same. The matrix for this transformation is $\begin{array}{c} \begin{array}{ccc} x\text{-value} & y\text{-value} & z\text{-value} \end{array} \\ \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & +1 \end{bmatrix} \end{array}$.



$\rightarrow \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

SHEAR

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



A unit square in fig. A is converted to

parallelogram B using x-direction shear matrix.

$$\begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ with $sh_x = 2$.

An x-direction shear is related to produce a transformation matrix $\begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ which transforms the coordinate position as $x' = x + sh_x \cdot y$, $y' = y$.

2DIMENSIONAL VIEWING: 2M.

A world coordinate area selected for display is called window. An area on a display device to which a window is mapped is called view port.

Window defines what is to be viewed.

View port defines where is to be displayed.

Windows and view ports are rectangles in standard position.

Mapping of a part of world cordinate scene to device cordinate is reffered to as viewing transformation or view window - to - view port transformation. or windowing transformation.

NOTE: Polygon shapes and circles are used in some application but these shapes takes longer to process.

2D VIEWING TRANSFORMATION PIPELINE

Diagram.

Modelling coordinates ⇒ Construct world coordinates scence using modelling coordinate transformation ⇒ World coordinates

⇒ Convert world coordinates to viewing coordinates ⇒ View coordinates ⇒ Map viewing coordinates to normal viewing coordinates using window-viewport specification.

NVC ⇒ Map normalised view port to device coordinates ⟹ Device Coordinates.

View ports are typically defined within the unit square (normalised coordinates). This provides a means for seperating the viewing and other transformation from specific o/p device requirements, ~~show~~ so that the graphic packages are largely device independant.

Different o/p devices can be used by providing the appropriate device drivers.

Clipping procedures are of fundamental importance in computer graphics. They are used not only in viewing transformation but also in

2m.

World
coordinate

viewing
es to normalised
coordinates
dow-viewport
tion.

the unit

s a

her

porfance

only

window - managing system. In painting and drawing packages to eliminate parts of a picture inside or outside of a designed screen area and in many other applications.

CLIPPING OPERATION: 2m. Alg. - 13 m.

Any procedure that identifies those portions of a picture that are either inside or outside of a specified region of space is referred to as clipping algorithm (or) clipping. The region against which an object is to be clipped is called a clipped window.

APPLICATIONS OF CLIPPING:

* Extracting part of a defined scence for viewing.
* Identifying visible surfaces in 3D views.
* Antialiasing the line segments or object boundaries.
* Creating objects using solid modeling procedures
* Displaying a multi window environment
* Drawing and painting operations. that allow part of a picture to be selected for copying, moving, erasing or duplicating.

For viewing transformation, we want to display only those picture parts that are within window area (assuming that the clipping flags have not been set to no clip) everything outside the window is discarded.

Types of Clipping: 2m

1. Point Clipping
2. Line Clipping (straight line segment).
3. Area Clipping (Polygon).
4. Curve Clipping
5. Text Clipping

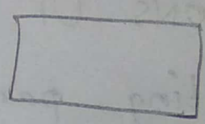
Line and polygon clipping are standard components of graphics packages.

31:01:2018

POINT CLIPPING:

Clipping window is rectangle shape

$P = (x, y)$

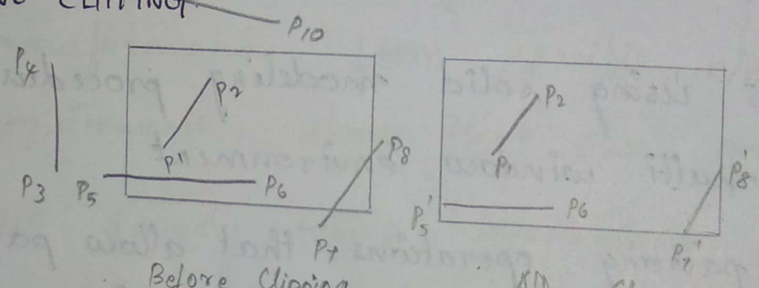


$xwmin \le x \le xwmax$

$ywmin \le y \le ywmax$

Used to depict explossion, sea foam, sea particles.

LINE CLIPPING:



Before Clipping

After Clipping

Parameteric representation: (x_1, y_1) (x_2, y_2)

$$x = x_1 + u(x_2 - x_1)$$

$$y = y_1 + u(y_2 - y_1)$$

$$0 \le u \le 1$$

1. Take a line whether the line is inside or outside.
2. If it is neither inside nor outside then intersection we follow inside-outside test.
3. If it is outside the window, discard the line

COHEN SUTHERLAND LINE CLIPPING ALGORITHM : 13 mark diagram

This is one of the oldest most popular line clipping algorithm. Generally the method speaks up the processing of line segment by performing the initial test to reduce the no. of intersection that must be calculated. Everyline endpoint in a picture is assigned. A four digit binary code called region code that identifies the location of the point relative to the boundaries of clipping rectangle.

1001	1000	1010
0001	0000 Window	0010
0101	0100	0110

Each bit position in the region code is used to indicate one of the four relative coordinate position of the point with respect to clip window left, right, top, bottom.

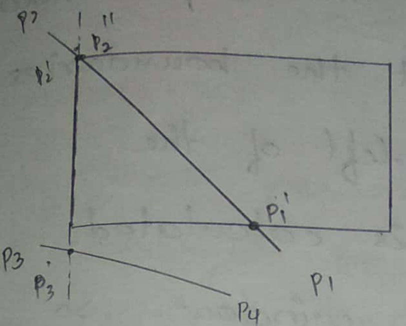
Bit-1 – LEFT
Bit – 2 – RIGHT
Bit 3 – bottom
Bit 4 – TOP

The value of 1 in any bit position ... that the point is in the relative position otherwise the point is set to 0. If the point is within the clipping rectangle then the region code is 0000. A point that is below and to the left of rectangle has a region code of 0101.

Bit values in the region code are determined by comparing endpoint coordinate value (x, y) to the clipping boundaries. Bit-one is set to 1 x is less than xw_{min} and the other 3 bit values can be determined using similar comparison. Bit-1 is sign bit of $x - xw_{min}$, bit-2 is sign bit of $xw_{max} - x$, bit-3 is sign bit of $y - yw_{min}$ and bit-4 is sign bit of $yw_{max} - y$.

Any lines that are completly contained within the window boundary have a region code 0000 for both endpoints, we accept these lines. Any lines that have 1 in same bit position in the region code for each end point are completly outside the clipping rectangle, we reject these lines. We would discard the line that has a region code of 1001 for one endpoint and code of 0101 for other endpoint.

Both endpoints of this line are left of clipping rectangle, as indicated by 1 in bit-1 of each region code. logical AND operation is performed with both the region codes to test the line. If the result is not 0000, then the line is completely outside the clipping region.



Lines that cannot be identified as completely inside or completely outside in the clip window are checked for intersection with the window boundary as shown in above figure. Such lines may or may not cross into window interior. Clipping process for a line is done by comparing a outside endpoint to a clipping boundary to determine how much of line can be discarded.

Cohen Sutherland algorithm is processed with the above diagram, starting with bottom endpoint P_1 to P_2, we check P_1 against left, right and bottom boundaries inturn find the this point is below the clipping rectangle.

We then find the intersection point P_1' with the bottom boundary and discard the line section. Discard the line section from P_1 to P_1'. The line now has been reduced to the section P_1' to P_2. Since P_2 is outside the clip window we check this endpoint against the boundaries and find that it is the left of the window. Intersection point P_2' is calculated but this point is above the window. So the final intersection calculation is P_2''. Line from P_1' to P_2'' is saved. This completes the processing for this line so we save this part and go on to next line. Point P_3 in the next line is left of the clipping rectangle so we determine the intersection P_3' and eliminate P_3 to P_3'. By checking region codes for the line section P_3' to P_4, we find that the remainder of the line is below the clip window so can be discarded also.

Algorithim : must.

LIANG - BARSKIN LINE CLIPPING ALGORITHM : 8M or 8M.

Faster line clippers have been developed that are based on the analysis of parameteric equation of a line segment

$$x = x_1 + u\Delta x.$$
$$y = y_1 + u\Delta y \quad , \quad 0 \le u \le 1.$$

where $\Delta x = x_2 - x_1$ $\quad \Delta y = y_2 - y_1$

Using these parameteric equation, cyrus and beck developed an algorithm that is generally more efficient than cohen sutherland algogorithm. Later Liang - Barskin independantly deviced a faster line clipping algorithm. We first write the point clipping conditions in parameteric form

$$xw_{min} \le x_1 + u\Delta x \le xw_{max} \, ,$$

$$yw_{min} \le y_1 + \Delta y u \le yw_{max}.$$

Each of these 4 inequalities can be expressed as $uP_k \le q_k$, $k = 1, 2, 3, 4$ where parameters p and q are defined as $P_1 = -\Delta x$, $P_2 = \Delta x$,
$P_3 = -\Delta y$, $P_4 = \Delta y$, $q_1 = x_1 - xw_{min}$,

$q_2 = xw_{max} - x_1$,

$q_3 = y_1 - yw_{min}$,

$q_4 = yw_{max} - y_1$.

Any line that is parallel to one of the clipping boundary as $P_k = 0$ for the value of k corresponding to that boundary ($k = 1, 2, 3, 4$)

corresponding to left, right, bottom and top boundary respectively. For value of k we also find $q_k < 0$ then the line is completely outside the boundary and can be eliminated further consideration. If $q_k \geq 0$ the line is inside the parallel clipping boundary. If ~~$q_k \geq 0$~~. $P_k < 0$, infinite extension of line proceeds from outside to inside of the infinite extension of this particular clipping boundary. If $P_k > 0$, the line proceeds from inside to outside. For a non-zero value of P_k, we can calculate the value of u that corresponds to the point where the infinitely extended line intersect the extension of the boundary k as $U = q_k / P_k$.

05:02:2018

For each line we can calculate values for parameters u_1 and u_2 that define a part of the line that lies within the clip rectangle. The value of u_1 is determined by looking at the rectangle edges for which the line proceeds from outside to inside ($P<0$). For these edges we calculate $R_k = q_k / P_k$.

Value of U_1 is taken as largest of the set consisting of 0 and various values of R. Value of U_2 is determined by examining the boundaries for which line proceeds from inside to outside ($P>0$). Value of R_k is calculated for each of these boundaries and the value of U_2 is the minimum of the set consisting of 1 and calculated R values.

If $U_1 > U_2$ the line is completely outside the clip window and it can be rejected otherwise, the endpoints of the clip line is calculated from 2 values of the parameter U.

Line clipping Algo.

4 inequalities.

$Q_1\ Q_2,\ P_1,\ P_2\ \ P_3,\ P_4$.

Case - (4).

$U = q_k / P_k$.

Algorithm:

Clip Test.

2M: Difference b/w cohen sutherland & Liang Barskin

Liang - Barskin algorithm is more efficient than cohen sutherland algorithm since intersection calculations are reduced. Each update of parameters u_1 and u_2 requires only one division and window intersection of the lines are computed only once, when the final values of u_1 and u_2 have been computed.

In constrast cohen sutherland algorithm ~~Nicholl~~ repeatedly calculate the intersection along the line path. Eventhough the line maybe completely outside the clip window. And each intersection calculation requires both division and multiplication.

NOTE: Cohen Sutherland & Liang - Barskin algorithm can be extended to 3 dimensional clipping.

// not for exam

Nicholl - Lee - Nicholl Line Clipping: (Among the 3, this is the best algorithm). It supports only 2D.

By creating more regions around the clip window, NLE avoids multiple clipping of an individual line segment.

Compared to both Cohen Sutherland and Liang Barskin algorithm, NLN performs fewer comparisons & division. NLN algorithm can only be applied to 2D clipping whereas Liang-Barskien & Cohen Sutherland can be easily extended to 3D scenes. // not for exam.

05:02:2018

UNIT-III THREE DIMENSIONAL CONCEPTS

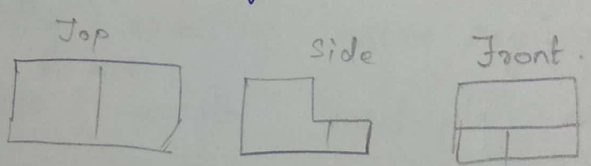
Viewing transformation in 3D is much mor complicated because we have many more parameters to select when specifying how a three dimensional scence is to be mapped to a display device.

3D DISPLAY METHOD: Ex: Camera

PARALLEL PROJECTION

One method for generating a view of soli object is to project points on the object surfa along parallel lines onto the display planes by selecting different viewing positions, we can select visible points on the object onto the display plane to obtain different 2D views of the object.

Ex: Victory Stand. (Top, front, side view - 2D)



In a parallel projection, parallel lines in the world cordinates scence project into parallel lines on the 2D display frame. This technique is used in engineering & architectural drawings with set of views to maintain relative proposition of the object. The appearance of the solid objects can then be reconstructed from the major views

07:02:2018

PRESPECTIVE PROJECTION:

Another method for generating a view of 3D scence is to project points to the display plain along converging part. This causes the objects further from the viewing position to be displayed smaller than the objects of the same size that are nearer to the viewing positions.

Ex: Airport.

In a perspective projection, the parallel lines in a scence that are not parallel to the display plane are projected into converging lines. Scences displayed using prespactive projection appears more realistic since this is the way our eyes or camera lenes forms images. Distant object appear smaller than objects closer to the viewing position.

DEPTH VIEWING: CUEING :

Depth information is important so that we can easily identify for a particular viewing

direction which is front or back of displayed object.
A simple method for indication depth with wire frame display is to vary the intensity of the object according to the distance from the viewing position. The lines closet to the viewing position are displayed with highest intensity and lines further away are displayed with decreasing intensity. Depth cueing is applied by choosing max and min intensity (or color) values and a range of distance over which intensities are to vary.

APPLICATION OF DEPTH CUEING:

Application of depth cueing is modelling the effect of atmosphere on the perseued intensity of the object. Most distant objects appear dimmer to us than the nearer objects due to light scatering ~~due~~ ~~by~~ dust particles Haze and smoke. Some atmospheric effects can change the persued color of an object and we can model these effects with depth cueing.

12:02:2018

VISIBLE LINE & SURFACE IDENTIFICATION:

We can also clarify the depth relationship in a wireframe display by identifying visible lines in some way. the simplest method is to highlight the visible lines or to display them in a different color.

Another technique commonly used for engineering drawing is to display the non-visible lines as DASH Lines. Another approach is to simply remove the non-visible lines. But removing the hidden lines also removes information about the shape of the back surfaces of an object. These visible line methods also identify the visible surfaces of the object.

When objects are to be displayed with color (or) shaded surfaces, we apply surface-rendering procedures to the visible surfaces so that the hidden surfaces are obscured.

Surface Rendering:

Added realism is attained in displays by setting the surface intensity of objects according to lightning conditions in the same and according to attained surface characteristics.

Lighting specification include intensity and position of light sources and general background illumination required for a scene.

Surface properties of object include degree of transparency or how rough or how smooth the surface should be. Surface rendering methods are combined with perspective visible - surface identification to generate a degree of realism ~~to~~ in a display ~~scene~~ scene.

Exploded and Cut away views :

Many graphics packages allow objects to be identified as hierarchical structures so that the internal details can be stored. Exploded and cut-away views of such objects can then be used to show the internal structure and relationship of object parts. Ex: Motor Engine.

An alternative to exploring the objects into its component part is the cut-away view which removes part of visible surfaces to show internal structures.

3D STEREOSCOPIC VIEWS :

Another method for adding a sense of reallism to a computer generated scene is to display

object using either 3D or stereoscopic views.
3D views can be obtained by reflecting a raster image from a vibrating flexible mirror. Vibration of mirror are synchronised with the display of scene on the CRT. As the mirror vibrates the focal length varies so that each point in the scene is projected to a position corresponding to its depth.

Stereoscopic devices present two views of a scene, one for the left eye and the other for the right eye. The two views are generated by selecting viewing position that correspond to the two eye position of a single viewer. These two views then can be displayed on an alternate refresh cycles of a raster monitor and viewed through glasses that alternately draken first one lens and then the other in synchroni-zation with the monitor refresh cycles.// Any 4 methods 8 mark.

3D OBJECT RERRESENTATION:

Representation schemes for solid objects are often divided into two board categories all though not all representations fall neatly into one or other of these two categories

Boundary Representation (B-REPS)
Space Partitioning Representation.

14 : 02 : 2018

Polygon surfaces: 8m

The most commonly used boundary representation for a 3D graphic object is a set of surface polygons that enclose the object interior. Many graphic system store all object descriptions as a set of surface polygon.

Reason for using polygon:

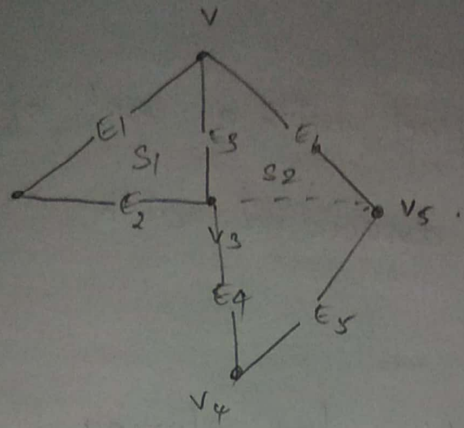
This surface polygon simplifies and speeds up the surface rendering and display of objects since all surfaces are described with linear equation, for this reason polygon descriptions are often referred to as "Standard Graphics Objects".

NOTE:

Polygonal representation is the only one available but many packages allow objects to be described with other schemes such as spline surfaces that are then converted to polygonal representations for processing.

POLYGON TABLE:

Polygon data tables can be organised into two groups (1) Geometric tables (2) Attribute tables.



(1) VERTEX TABLE

V_1 - x_1, y_1, z_1
V_2 - x_2, y_2, z_2
⋮
V_5 - x_5, y_5, z_5

(2) EDGE TABLE

E_1 : V_1, V_2
E_2 : V_2, V_3
E_3 : V_3, V_1
E_4 : V_4, V_3
E_5 : V_4, V_5
E_6 : V_5, V_1

(3) POLYGON SURFACE TABLE

S_1 : E_1, E_2, E_3
S_2 : E_3, E_4, E_5, E_6.

(4) Edge table for surfaces of the above digram expanded to include pointers to the polygon table

E_1 : V_1, V_2, S_1
E_2 : V_2, V_3, S_1
E_3 : V_3, V_1, S_2
E_4 : V_4, V_3, S_2.
E_5 : V_4, V_5, S_2
E_6 : V_5, V_1, S_2.

Geometric data table representation with two adjacent surfaces, 6 edges and 5 vertices.

Geometric datal table contain vertex coordinates and parameters to identify spatial orientation of the polygon surfaces.

Attribute information of an object includes parameters specifying the degree of transparency of the object and its surface reflextivity and texture and characteristics.

Scanned by CamScanner

//NOTE:
Polygon surface is specified with a set of vertex coordinates and associated with set of parameters.//

A convienient organisation for storing geometric data is to create 3 list.

(1) Vertex Table
(2) An edge Table
(3) Polygon Table.

Coordinate values for each vertex in the object are stored in vertex table. Edge table contains pointers back into vertex table to identify the vertices for each polygon edge. And polygon table contains pointers back into the edge table to identify edges for each polygon. This scheme is illustrated in the above diagram which uses two adjacent polygons on an object surface.

Listing the geometric data in three tables as in above diagram provides a convienient reference to the individual components (vertices, edges, polygons) of each object. Also the object can be displayed efficiently by using the data from the edge table to draw the component lines.

An alternative arrangement is to use, just two tables a vertex table and a polygon table. But this scheme is less convenient and some edges could get drawn twice.

Another posibility is to use only a polygon table but this duplicates coordinate information, since explicit coordinate values are listed for each vertex in each polygon. Also edge information has to be reconstructed for the vertex listing in the polygon table.

Since geometric data table may contain extensive listings of vertices and edges for complex objects. It is important for the data being checked for the consistency and completeness. More information included in data tables, by easier it is to check for errors. Therefore error checking is easier when three (3) data tables are used (Vertex, edge, polygon Table). Since this scheme provides the most information some of the test could be performed by a graphic package are

1. That every vertex is listed as an endpoint for atleast two edges.

(2) That every edge is part of atleast one polygon.

(3) That every polygon is closed

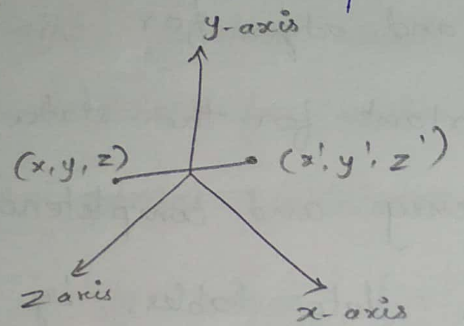
(4) Each polygon has atleast one shared edge.

(5) If edge table contains pointers to polygon every edge is referenced by a polygon pointer. As a reciprocal pointer back to the polygon.

19:02:2018

3D GEOMETRIC AND MODELING TRANSFORMATION: 13m

Translation : In 3D homogenous coordinate representation, a point is translated from position



Translating a point with translation vector $T(t_x, t_y, t_z)$

$P = (x, y, z)$ to position $P' = (x', y', z')$ with matrix operation

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \rightarrow (1)$$

$$\boxed{P' = T \cdot P}$$

The parameters t_x, t_y, t_z specifying the translation distance for the coordinate direction are assigned any real values. In equation (1) matrix representation is equivalent to three equations

$$\boxed{x' = x + t_x}$$
$$\boxed{y' = y + t_y}$$
$$\boxed{z' = z + t_z}$$

We obtain the inverse of the translation matrix in eq (1) by negotion of the translation distance t_x, t_y, t_z. This produces a translation in the opposite direction. The product of the translation matrix and its inverse produces the identity matrix.

Rotation :

To generate a rotation transformation for an object we must designate an axis of rotation (about which the object is to be rotated) and the amount of angular rotation.

Coordinate Axis Rotation :

Easiest rotation axis to handle are those that are parallel to the coordinate axis.
2D z-axis rotation equations are easily extended to 3D.

$$\boxed{\begin{aligned} x' &= x\cos\theta - y\sin\theta \\ y' &= x\sin\theta + y\cos\theta \\ z' &= z \end{aligned}} \longrightarrow (1).$$

Scanned by CamScanner

The parameter θ specifies the rotation angle in homogenous coordinate form. The 3D z-axis rotation, the equations are expressed as

for x-axis rotation replace $x \to y$, $y \to y$, $z \to x$ in eq ①.

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

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Transformation equation for rotation about the other two coordinate axis can be obtained with a cyclic permutation of the coordinate parameters (x, y, z) in equation ①. We use the replacement $x \to y \to z \to x$ -(2) substituting the permutation (2) in equation (1) we get the equation for x-axis rotation.

$$y' = y\cos\theta - z\sin\theta$$
$$z' = y\sin\theta + z\cos\theta$$
$$x' = x$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

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

Rotation of ~~axis~~ object around y axis,

$$z' = z\cos\theta - x\sin\theta.$$
$$x' = z\sin\theta + x\cos\theta$$
$$y' = y$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

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

An inverse rotation matrix is formed by replacing rotation angle θ by $(-\theta)$ negative values for rotation angle generate rotation in clockwise direction, so the identity matrix is produced when any rotation matrix is multiplied by its inverse.

Since only the sine function is affected by the change in the sign of the rotation angle, the inverse matrix can also be obtained by interchanging rows and columns

$$R^{-1} = R^{T}$$

Scaling :

The matrix expression for the scaling transformation relative to the coordinate origin $P = (x, y, z)$ can be written as $P' = S.P$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \rightarrow (1)$$

where the scaling parameters S_x, S_y, S_z are assigned any positive values.

$$\left.\begin{aligned} x' &= x \cdot S_x \\ y' &= y \cdot S_y \\ z' &= z \cdot S_z \end{aligned}\right\} \rightarrow (2)$$

Scaling an object with transformation with eq (1) changes the size of the object and repositions the object relative to the coordinate origin.

21:02:2018

Also, if the transformation parameter are not equal relative dimensions in the object are changed. We preserve the original shape of an object with a uniform scaling.

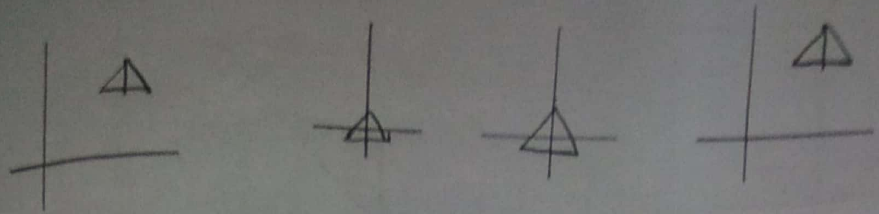
$$S_x = S_y = S_z$$

Scaling with respect to a selected fixed point position x_f, y_f, z_f can be represented with the following transformation segments.

(1) Translate the fixed point to the origin.

(2) Scale the object ...

3. Translate the fixed point back to its original position.



We form the inverse scaling matrix for equation(1) by replacing scaling parameters S_x, S_y and S_z with their reciprocals.

Inverse matrix generates an opposite scaling transformation so the concatination of any scaling matrix and its inverse produces the identity matrix.

Other transformations :
Reflection
Shear.

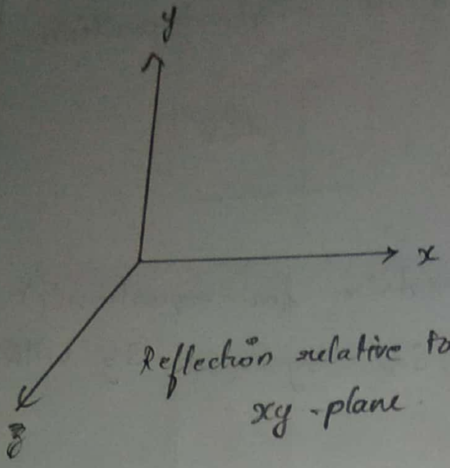
(1) Reflection :

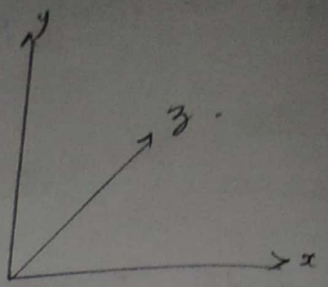
A 3D reflection can be performed relative to a selected reflection axis or with respect to a selected reflection plane.

In general, three dimensional reflection matrices are set up similarly to those for two dimension.

Reflection relative to a given axis are equivalent to 180° rotation about that axis.

When the reflection plane is a coordinate plane (either (xy, yz or xz plane). We can think of transformation as a conversion between left handed and right handed system.





Reflection relative to xy-plane

An example of reflection that from a right handed system to left handed system or viceversa as showed in the above diagram.

This transformation changes the sign of the z-coordin leaving x and y coordinate values unchanged.

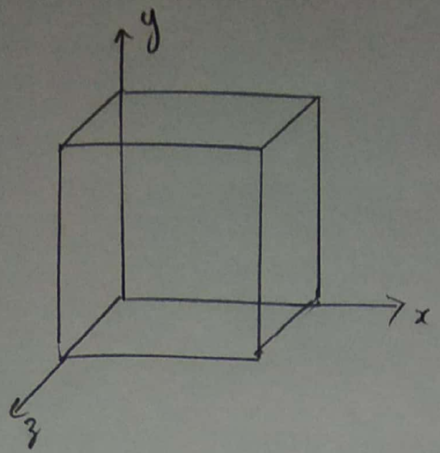
In matrix representation, of point relative to xy-plane is

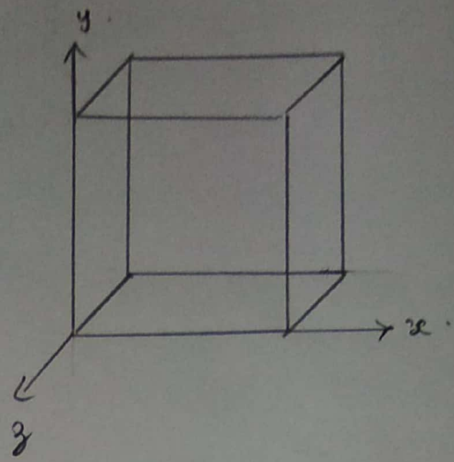
$$RF_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.) Shear:

Shearing transformations can be used to modify object shapes. They are also useful in three dimensional viewing for obtaining general projection transformation.

Examples of 3D shearing, the following transformation produces, z-axis shear.





$$SH_z = \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Parameters a and b can be assigned any real values, the effect of this transformation matrix is to alter x and y coordinate values by an amount that is proportional to the z-value while leaving the z-coordinate unchanged.