Name : Samiksha Patil

Sap ID : 60004180092

SE-B Computer

ASSIGNMENT 1

computer graphics

Q1. **Definition of scan conversion, rasterization and rendering.**

* Scan Conversion:-

Scan conversion is the process of representing continuous graphics as a collection of discrete pixels. Each pixel on monitor screen is either turned off or on according to picture definition stored in frame buffer.

* Rasterization:-

Rasterization is the process of converting vector graphics into raster graphics. It converts the shapes into a set of pixels and displays it on monitor screen or printer.

* Rendering:-

Rendering refers to the process of generating photorealistic image form 2D or 3D models using computer program. It retrieves information like vertex position ,viewpoint ,color ,etc. and renders the scene on monitor screen.

Q2. Differentiate between Flat Panel displays like LCD and LED.

|  |  |
| --- | --- |
| LCD | LED |
| * Liquid crystal display(LCD) is a non-emissive display device. * Uses optic effect to convert natural or synthetic light to graphic scene on screen. * Grid of crystalline cells called nematic cells are used to design LCD. * Intersection of conductor plates screen resolution and pixel position. * Scene is rendered depending on ‘on’ or ‘off’ state of cell alignment. * LCD have lesser viewing angle. | * Light Emitting Diode(LED) is an emissive display device. * Converts electrical energy into light to generate display. * Multiple diodes are used to design LED. * Matrix of diodes are arranged to form the pixel positions. * Scene is rendered when picture definition is read out from refresh buffer. * LED have wider viewing angle than the LCD. |

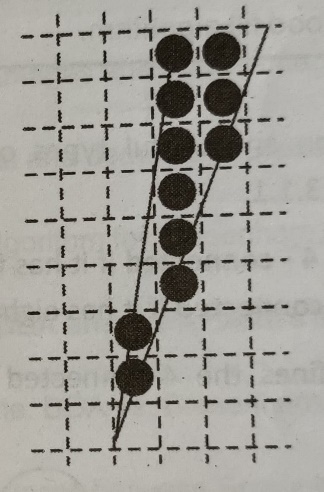
Q3. Scan line Polygon Fill algorithm.

This method compares the visible span of a scan line within polygon boundaries. It selects the pixel which is closer to the edge irrespective of on which side of the polygon it lies.

Scan line polygon filling approach works as follows:

* find intersections of scan line polygon edges
* sort all intersections on the x-coordinate
* fill the span using a pair of sorted intersection points

Horizontal edges are not considered for processing. a problem with this approach is Sliver. The figure demonstrates sliver :



Sliver

Scan line approach takes advantage of Edge coherence.

Edges intersecting scan line k are also intersected by scan line k + 1. If the current intersection point is (xk,yk) the intersection point on the next scan line is computed as,

yk+1 = yk + 1

xk+1 = xk + 1/m

This approach two data structures, edge table and active edge table. Edge table maintains the following information for each edge in the polygon, except horizontal edges :

|  |  |  |  |
| --- | --- | --- | --- |
| Ymax | Xmin | 1/m | \*ptr |

Each entry in the edge table maintains edge list in increasing order of their x-coordinate. Scan line algorithm works as follows:

* set Yscan to smallest y for which there is an entry in ET
* AET is initially empty
* repeat following steps until both, AET and ET are empty :
  + move edges from ET to AET whose ymax = yscan
  + fill the visible span
  + remove edges with
  + increment y by 1
  + for each edges in AET, update x-coordinate
  + sort all edges on x

This algorithm optimizes the computation by using scanline coherence and edge coherence.

**Q4. Composite transformation.**

Composite transformation is any sequence of basic transformation operations.

We can generate a composite transformation matrix by multiplying matrices of basic transformation operation like translation, scaling and rotation, reflection, shearing, etc.

For column measure representation matrices are placed from right to left.

Let’s say M1, M2, . . . . ,Mn are the transformation matrices , for column measure representation of transformed coordinates given as,

P’ = Mn  . . . . M2.M1.P

Where Mi indicates ith operation in sequence.

For row measure representation , transformed coordinates given as,

P’ T = PT . M1T. M2T . . . . MnT

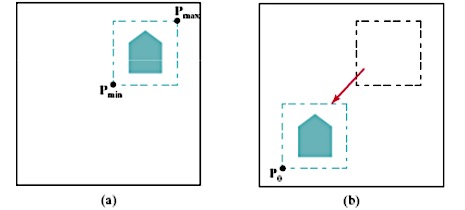
* Successive Transformations:

These include :

* Translation
* Rotation
* Scaling
* General Pivot Point Rotation:
* Translate the pivot point to the origin.
* Rotate object.
* Translate back the pivot point to its original place
* General Pivot Point Scaling:
* Translate pivot point to the origin.
* Scale the object by given scaling parameters.
* Translate back the pivot point to the original place.

**Q5. Raster method for transformation.**

* All bit settings in the rectangular area shown are copied as a block into another part of the frame buffer :



* Rotate a two-dimensional object or pattern 90 counterclockwise by reversing the pixel values in each row of the array, then interchanging rows and columns.
* For array rotations that are not multiples of 90, we need to do some extra processing .
* Similar methods to scale a block of pixels :

