CMP 477 Computer Graphics
Module 4 Basic Geometric Objects

Dr. S.A. Arekete

Redeemer's University, Ede

Graphic Primitives

- The basic geometric objects in CG are usually called primitives or graphic output primitives
- A primitive is a graphics object that is essential for the creation or construction of complex images.
- Fortunately, graphics is constructed from a few basic elements, as opposed to the great variety of graphics applications.

Points

- Points are uniquely defined by their x- and y-coordinates
- Points are usually not drawn themselves
- Their main function is the description of other objects like lines that can be defined by their two endpoints

Lines, Polylines or Curves

- These can be defined by two or more points
- For lines two points are needed
- Curves need two points and additional control points
- Polylines are connected sequences of lines

Areas

- Areas are bounded by closed polylines or polygons
- Areas can be filled with colour or a texture

Pixel

- A pixel is a point of light.
- It is just one tiny dot on the raster displays.
- Though it has no structure, it is definitely a building block and hence it can be considered as the graphics primitive.
- The resolution of CRT is related to the dot size, the diameter of a single dot.
- A resolution of 100 dots lines/inch implies a dot size of 0.01 inch.
- However, in reality, pixels are more elliptic than circle.
- The shape of a pixel purely depends upon the characteristics of the visual display unit.

Pixel..

- The ratio of the distance between the centres of two adjacent horizontal pixels to that of the vertical ones is called the *pixel ratio*.
- For example, if 100-pixel lines in x and y measure 6cm and 8cm respectively, then

$$Xresolution = 100pixels/6cm \ \dots\dots\dots\dots\dots (1)$$
$$Yresolution = 100pixels/8cm \ \dots\dots\dots\dots\dots (2)$$

- Pixel ratio should be considered in line-generating algorithms.

Line

- Line, especially straight lines, constitute an important building block of computer images.
- For example, line is the basic building block of Line graphs, bar and pie charts, two and three-dimensional graphs of mathematical functions, engineering drawings and architectural plans.
- In computer graphics, straight line is so basic in creating images that we call it a graphics primitive.
- Straight lines can be developed in two different ways.
- A *structural method* determines which pixels should be set before drawing the line;
- a *conditional method* tests certain conditions to find which pixel should be set next.

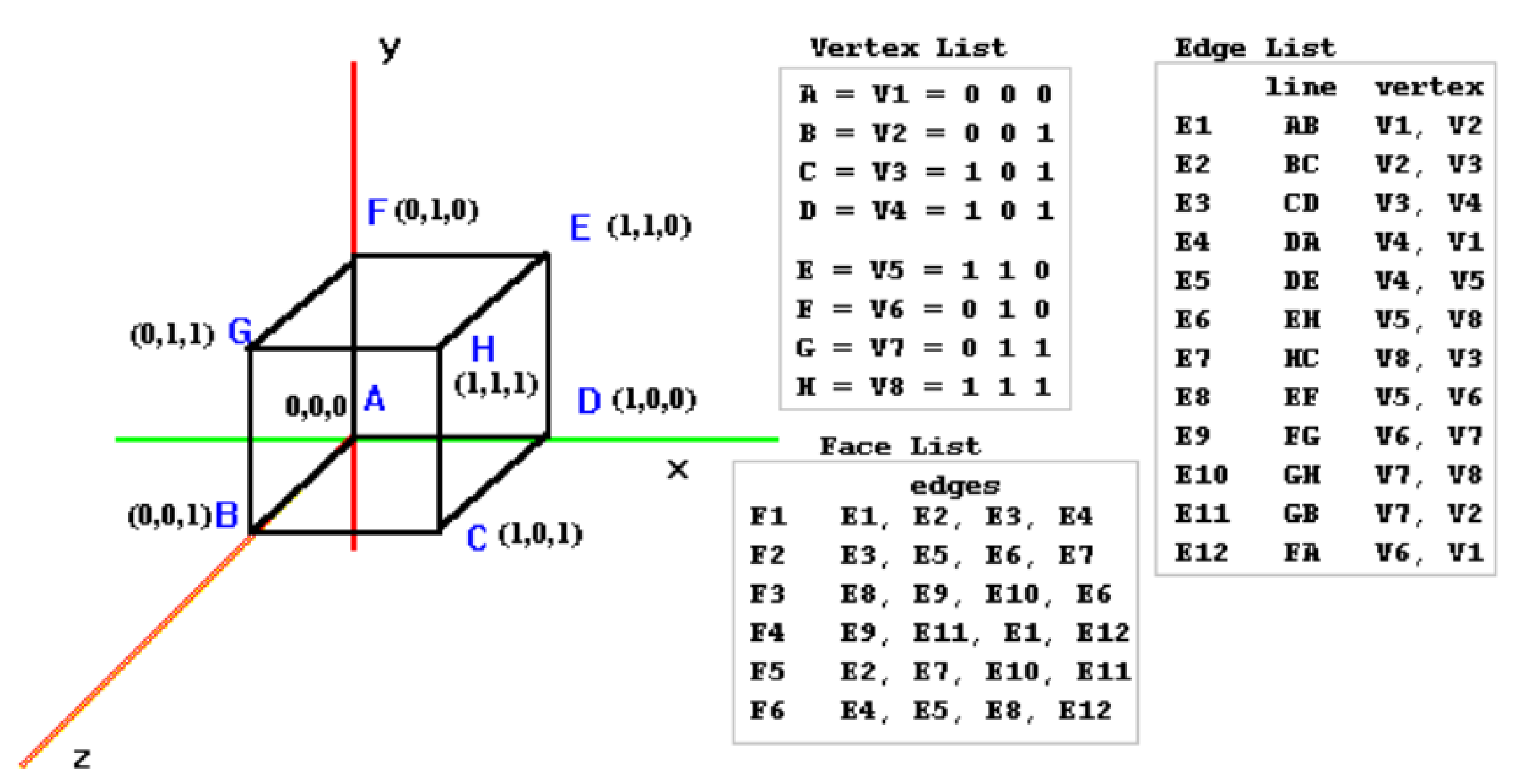
Polygon

- A polygon, even though generally constructed from straight lines, is an important graphics primitive.
- So often we want to handle polygon as a single entity, as images of objects from the real world consist in large part of polygons.
- A polygon is a closed area of image bounded by straight or curved lines and filled with one solid colour.
- Since images are two dimensional, a polygon is a closed planar figure.
- Implementing a polygon as a graphics primitive is natural and helpful.
- We can define polygon as an image which consists of a finite ordered set of straight boundaries called *edges*.
- Alternately, the polygon can be defined by an ordered sequence of *vertices*, the corners of the polygon.

Polygon..

- The edges of the polygon are then obtained by traversing the vertices in the given order;
- The edge list is sufficient for wireframe drawings. Two consecutive vertices define one edge.
- We close the polygon by connecting the last vertex to the first.
- Face list is required in order to fill the polygon.
- We can decompose a scene from real world into a collection of polygons of simple shapes.
- For example, a simple house can be constructed with a square and a rectangle.
- However, neither straight lines nor polygons precisely describe a real world scene;
- It is only an approximation of the scene we can get; such scenes actually seem to be of fractal nature.

Polygon..



Vertex List

A	=	V1	=	0	0	0
B	=	V2	=	0	0	1
C	=	V3	=	1	0	1
D	=	V4	=	1	0	1
E	=	V5	=	1	1	0
F	=	V6	=	0	1	0
G	=	V7	=	0	1	1
H	=	V8	=	1	1	1

Edge List

	line	vertex
E1	AB	V1, V2
E2	BC	V2, V3
E3	CD	V3, V4
E4	DA	V4, V1
E5	DE	V4, V5
E6	EH	V5, V8
E7	HC	V8, V3
E8	EF	V5, V6
E9	FG	V6, V7
E10	GH	V7, V8
E11	GB	V7, V2
E12	FA	V6, V1

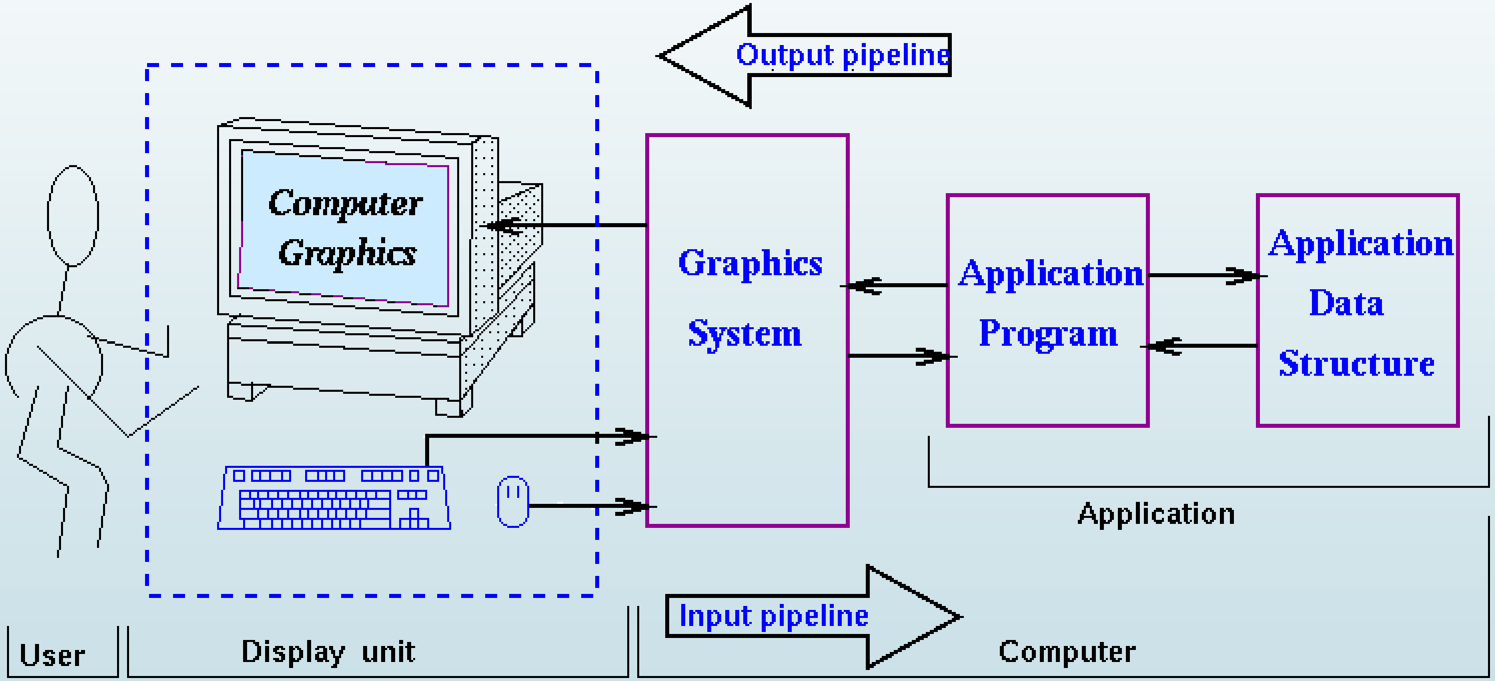
Face List

	edges
F1	E1, E2, E3, E4
F2	E3, E5, E6, E7
F3	E8, E9, E10, E6
F4	E9, E11, E1, E12
F5	E2, E7, E10, E11
F6	E4, E5, E8, E12

Graphics Pipeline

- The graphics pipeline specifies a series of steps needed to display data on an output device.
- The steps vary depending on the needs of the application.
- The application may be realistic, fast, aesthetic or informatic.
- However, the graphics Pipeline has three major components, viz. *the application program* which stores into and restores from *the application data structure* and sends graphics command to the *graphics system*.

Graphics Pipeline..



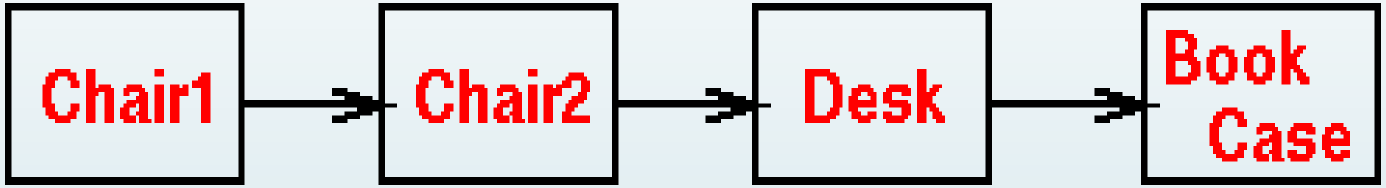
Graphics Pipeline..

- **Application program**: This is a collection of output plotting subroutines based on 2D or 3D geometry of the object to be displayed.
- The application program performs the following:
  - **View transforms**: specifies what part of the world scene is to be displayed and converts those points into view coordinate points.
  - **Converts from viewport to normailized device coordinates (NDC)**: specifies where in the view surface should the object be displayed.
  - **Clips**: Determine visible surfaces and shading. Then, the object outside the NDC is clipped.
  - **Maps data to device coordinates**: All the geometric and non-geometric details in the data structure are converted to one of the graphics output primitives and passed to the graphics system i.e. the objects are scan converted into pixels and the framebuffer displayed.

Graphics Pipeline..

- **Application data structure**: This is a database of descriptions and properties, like geometric coordinates, colour, surface texture and connectivity relationships of objects to be displayed on the display unit of the graphics system.
- For example, lets consider the design of a room with a few furniture in it.
- Then structure would contain
  - Description of primitives that defines the shape of the objects in the room.
  - Object attributes like line style, colour, texture that defines the "look" of the primitives
  - Connectivity relations and positioning data that defines how components fit together
  - Geometry spectrum that defines the layout of physical objects to description of concepts without geometry (eg. statistics)
  - Textual, numeric data (equations, formulas, etc.) and procedures often included in the models

Graphics Pipeline..



Chair1

Chair2

Desk

Book Case

Database of the set of furnitures in the room

Graphics Pipeline..

- **Graphics system**: This handles the low-level architecture of the display processor and xy co-ordinate system of the physical screen, hiding these details from the user.