CHAPTER -8

Graphical Languages

8.1Need for Machine Independent Graphical Languages

- A language that's NOT machine independent would either be unable to be run on certain types of computer- or perhaps may be designed in such a way that identical programs run differently on different computers.
- A Machine Independent language is one that can run on any machine.
- There are two general classifications for graphics:

1. **General Programming Packages**

- General programming packages provides an extensive set of graphics functions that can be used in a high-level programming language, such as C or FORTRAN.
- Example, GL(Graphics Library) system on Silicon Graphics equipment, which includes basic functions for generating picture components (straight lines, polygons, circle, and other figures), setting colors and intensity values, selecting views, and applying transformations.

2. **Special Purpose Application Packages**

- Special purpose applications packages are, in contrast designed for nonprogrammers, so that users can generate displays without worrying about how graphics operating works.
- The interface to the graphics routines in such packages allows users to communicate with the programs in their own terms.
- For example: the artist's painting programs and various business, medical, and Computer-Aided Design (CAD) systems.

There are many graphics software available in market; they can be categories as:

1. Paint Program: Paint program works with bit map images.
2. Photo manipulation Program: It works with bit map images and is widely used to edit digitized photographs.
3. Computer Aided Design Program: CAD software is used in technical design fields to create models of objects that will be build or manufactured. CAD software allows user to design objects in 3 dimensions and can produce 3-D frames and solid models.
4. 3-D Modelling Programs: It is used to create visual effects. 3D modeling program works by creating objects like surface, solid, polygon etc.
5. Animation: Computer are used to create animation for use in various fields, including games, and movies composing to allow game makers and film makers to add characters and objects to scenes that did not originally contain them.

8.2 Graphics Software Standard

- The primary goal of standardized graphics software is portability.
- When packages are designed with standard graphics functions, software can be moved easily from one 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 programs.
- Following standards are available
    - GKS (Graphics Kernel System)
    - PHIGS (Programmer's Hierarchical Interactive Graphics Standard)

8.3 Graphical Languages :PHIGS, GKS

1. GKS

- GKS stands for Graphics Kernel System
- This system provides standard methods for developing programs.
- It is sued by various programming languages like graphics software standard by ISO and by National Standard Organization including ANSI.
- GKS was originally designed as 2D graphics package.
- PHIGS was the extension of GKS which provides 3D graphics package.
- GKS includes various types of methods, reserved words.
- The two dimensional computer graphics which is closely related to six output functions of Graphical Kernel System. These are as follows:

  POLYLINES :- The GKS function for drawing line segments in called POLYLINE.
    - POLYLINE command takes an array X-Y coordinates and creates line segment joining them.

  POLYYMARKERS:- The GKS POLYMARKER function permits to draw symbols of marker centered at coordinate points.

  FILLAREA :- The GKS FILLAREA function permits to denote a polygonal shape of a zone to be filled with various interior shapes.

  TEXT :- The GKS TEXT function permits to sketch a text string at a specified coordinate place.

  CELL ARRAY:- The GKS CELL ARRAY function shows raster like pictures in a device autonomous manner.

  , Polymarker, text, Fill area, Cell-array, Generalized drawing primitives.
- Advantages of GKS

    - It provides improved algorithm
    - It makes system portable
    - Rewriting of code is not required.

2. PHIGS

- It stands for Programmer's Hierarchical Interactive Graphics Standard.
- PHIGS was the extension of GKS which provides 3D graphics package, increased capabilities for object modeling, color specification, surface rendering, and picture manipulations.
- PHIGS+ was developed to provide three-dimensional surface shading capabilities not available in PHIGS.
- PHIGS is an API standard for rendering 3D computer graphics, at one time considered to be the 3D graphics standard for the 1990s.

8.4 Overview of Graphics File Formats

- Graphics images are stored digitally using a small number of standardized graphic file formats, including bit map, TIFF, JPEG, GIF, PNG.
- Image file formats provide a standardized method of organizing and storing image data.
- Image files are made up of either pixel or vector (geometric) data, which is rasterized to pixels in the display process, with a few exceptions in vector graphic display.
- The pixels that make up an image are in the form of a grid of columns and rows.
- Each pixel in an image consists of numbers representing brightness and color.
- In each image file, the image size is dependent to the number of rows and column.
- The more rows and columns implies the greater the image resolution and greater the file size.
- Also, each pixel making up the image increase in size as color depth is increased.
- An 8-bit pixel (1 byte) can store 256 colors and a 24-bit pixel (3 bytes) can store 16 million colors. The latter is known as true color.

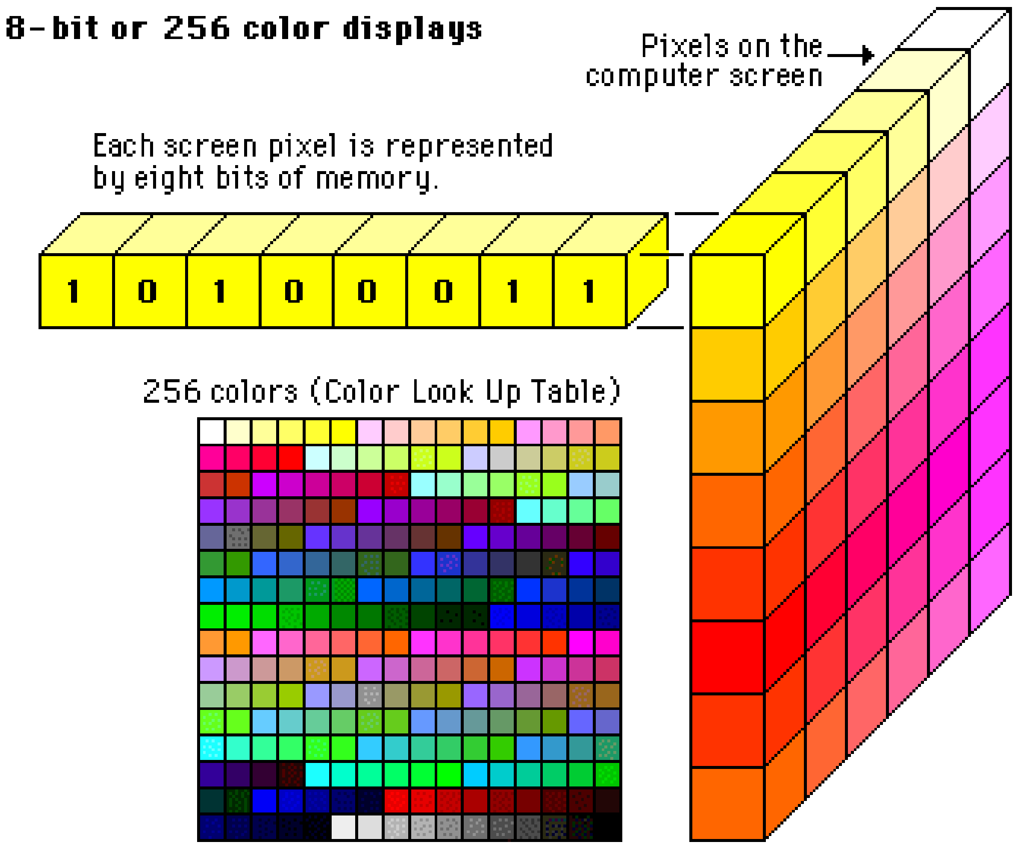

8-bit or 256 color displays
Pixels on the computer screen
Each screen pixel is represented by eight bits of memory.
1 0 1 0 0 0 1 1
256 colors (Color Look Up Table)

Image Compression

- Image compression is a method of using algorithms to decrease file size.
- High resolution cameras produce large image files.
- Files sizes may range from hundreds of kilobytes to many megabytes depending on the resolution of the camera and the format used to save the images.
- High resolution digital cameras record 8 megapixels (MP) (1 MP = 1000000 pixels or 1 million pixels) images, or more, in true-color.
- Consider an image taken by an 8 MP camera. Since each of the pixels uses 3 bytes to record true color, the uncompressed image would occupy 24,000,000 bytes of memory.
- That is a lot of storage space for just one image, and cameras must store many images to be practical.
- Faced with large file sizes, both within the camera, and later on disc, image file formats have been developed to address the storage problem.
- Thus, the image compression techniques defined on image file format have greater role to store the high resolution image with less memory requirement.
- There are two types of image file compression algorithms: lossy and lossless.

  - Lossless Compression algorithms reduce file size with no loss in image quality, though they usually do not compress to as small a file as a lossy method does.
  - When image quality is valued above file size, lossless algorithms are typically chose.
  - Lossy compression algorithms take an advantage of the inherent limitations of the human eye and discard information that cannot be seen. Most lossy compression algorithms allow for variable levels of quality (compression) and as these levels are increased, file size is reduced. At the highest compression levels, image deterioration becomes noticeable. This deterioration is known as compression artifacting.

A. GIF (Graphics Interchange Format)

- GIF format was originally developed by CompuServe in 1987.
- It is most commonly used for bitmap images composed of line drawings or blocks of a few distinct colors.
- The GIF format supports 8 bits of color information or less.
- The GIF format supports animation and is still widely used to provide image animation effects.
- It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.
- It is basically used for simple diagrams, shapes, logos and cartoon style images.
- In addition, the GIF89a file format support transparency, allowing you to make a color in your image transparent.
- This feature makes GIF a particularly popular format for Web images.

B. JPEG (Joint Photographic Experts Group)

- Like GIF, the JPEG format is one of the most popular formats for Web graphics that use the lossy compression techniques.
- It supports 24 bits of color information, and is most commonly used for photographs and similar continuous-tone bitmap images.
- The JPEG file format stores all of the color information in an RGB image
- JPEG was designed so that changes made to the original image during conversion to JPEG would not be visible to the human eye.
- Be aware that the chances of degrading your image when converting it to JPEG increase proportionally with the amount of compression you use.
- Unlike GIF, JPEG does not support transparency.

C. **BMP (Bitmap File Format)**
   - The BMP is used for bitmap graphics on the Windows platform only.
   - Unlike other file formats the BMP format stores image data from bottom to top and pixels in blue/green/red order.
   - Compression of BMP files is not supported, so they are usually very large.
   - The main advantage of MP files is their wide acceptance, simplicity, and use in Windows programs.

D. **TIFF (Tag Interchange File Format)**
   - The TIFF is a tag-based format that was developed and maintained by Aldus (now Adobe).
   - TIF, which is used for bitmap images, is compatible with a wide range of software applications and can be used for bitmap images, is compatible with a wide range of software applications and can be used across platforms such as Macintosh, Windows, and UNIX.
   - The TIFF format is complex, so TIFF files are generally larger than GIF or JPEG files.
   - TIFF can be lossy or lossless.
   - Some types of TIFF files offer relatively good lossless compression for bi-level (black and white, no grey) images.
   - Some high-end digital cameras have the option to save images in the TIFF format, using the LZW compression algorithm for lossless storage.
   - The TIFF image format is not widely supported by web browsers.
   - TIFF is still widely accepted as a photograph file standard in the printing industry.
   - TIFF is capable of handling device-specific color spaces such as the CMYK defined by a particular set of printing press inks.

E. **PNG (Portable Network Graphics)**
   - The PNG format will likely be the successor to the GIF file format.
   - The PNG file format supports true color (16 million colors) whereas the GIF file format only allows 256 colors.
   - The lossless PNG format is expected to become a mainstream format for web images and could replace GIF entirely.
   - It is platform independent and should be used for single images only (not animation).
   - Compared with GIF, PNG offers greater color support and better compression, gamma correction for brightness control across platforms, better support for transparency, and a better method for displaying progressive images.

8.5 Data Structure in Computer Graphics

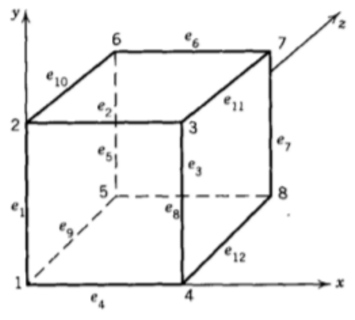
- Basic data structures for graphics
    - Primitive data types – INTEGER, REAL, BOOLEAN (LOGICAL), CHARACTER
    - Static data structure (array) – fixed, predefined values and occupies fixed memory locations.
    - Dynamic data structure (pointer) – Dynamic Storage allocation

- Representation of polyhedral objects using static arrays (Figures 7.6, 7.7, 7.8)

The three arrays used in the description of the cube are:

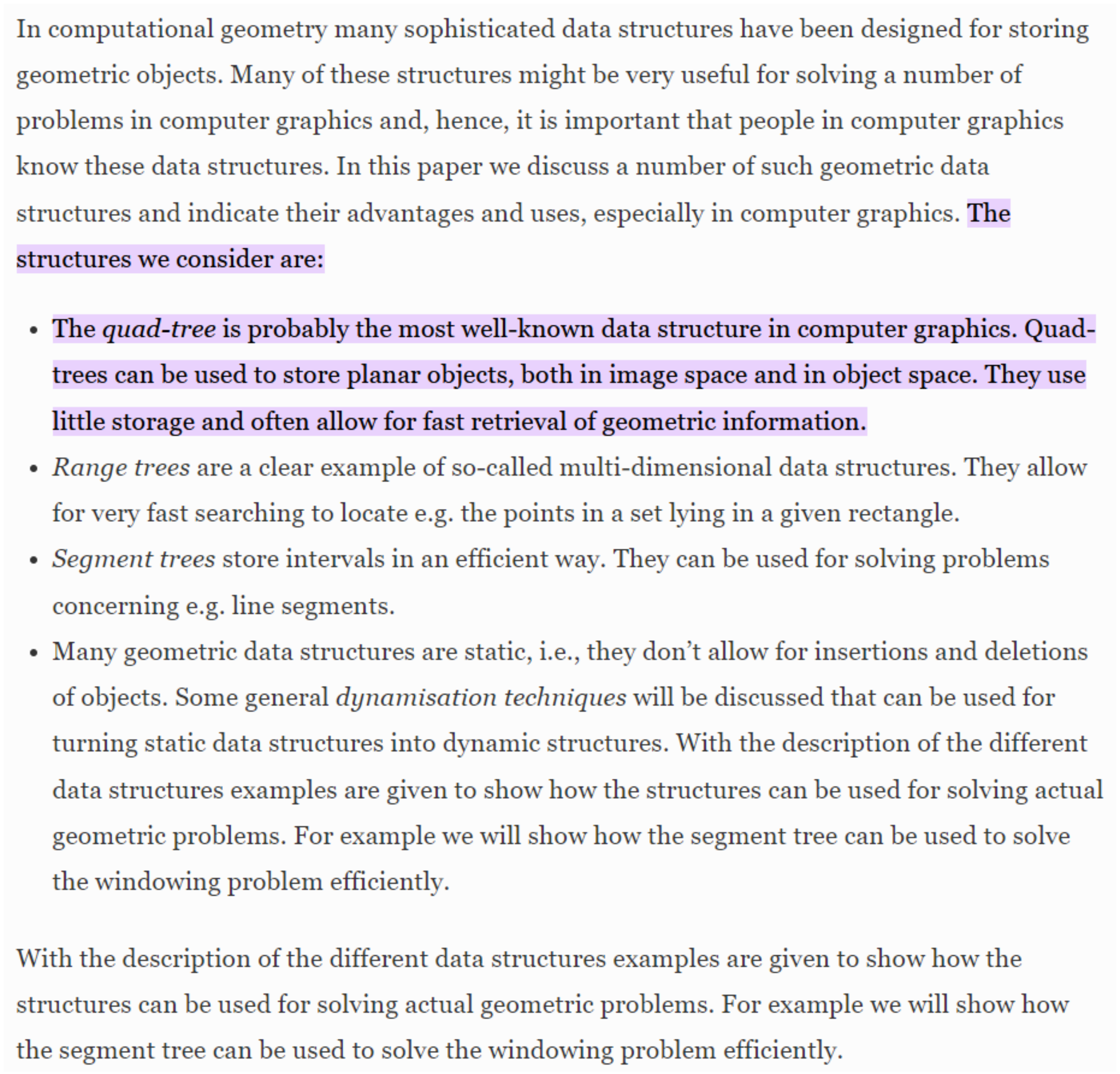
1. A vertex array listing the x, y, z coordinates of each vertex.
2. An edge array, pointing to the vertex array, indicating the two vertices at the end of each edge
3. A face array, pointing to the edge array, listing the edges that form each face

Arrays: Face ➔ Edge ➔ Vertex



	Edges	Vertices
Face 1	1, 2, 3, 4	1, 2, 3, 4
Face 2	1, 9, 5, 10	1, 5, 6, 2
Face 3	8, 7, 6, 5	5, 8, 7, 6
Face 4	3, 11, 7, 12	3, 7, 8, 4
Face 5	4, 9, 8, 12	1, 5, 8, 4
Face 6	2, 10, 6, 11	2, 6, 7, 3

FIGURE 7.6 Representation of a unit cube.



In computational geometry many sophisticated data structures have been designed for storing geometric objects. Many of these structures might be very useful for solving a number of problems in computer graphics and, hence, it is important that people in computer graphics know these data structures. In this paper we discuss a number of such geometric data structures and indicate their advantages and uses, especially in computer graphics. The structures we consider are:

- The *quad-tree* is probably the most well-known data structure in computer graphics. Quad-trees can be used to store planar objects, both in image space and in object space. They use little storage and often allow for fast retrieval of geometric information.
- *Range trees* are a clear example of so-called multi-dimensional data structures. They allow for very fast searching to locate e.g. the points in a set lying in a given rectangle.
- *Segment trees* store intervals in an efficient way. They can be used for solving problems concerning e.g. line segments.
- Many geometric data structures are static, i.e., they don't allow for insertions and deletions of objects. Some general *dynamisation techniques* will be discussed that can be used for turning static data structures into dynamic structures. With the description of the different data structures examples are given to show how the structures can be used for solving actual geometric problems. For example we will show how the segment tree can be used to solve the windowing problem efficiently.

With the description of the different data structures examples are given to show how the structures can be used for solving actual geometric problems. For example we will show how the segment tree can be used to solve the windowing problem efficiently.

8.6 Introduction to OpenGL

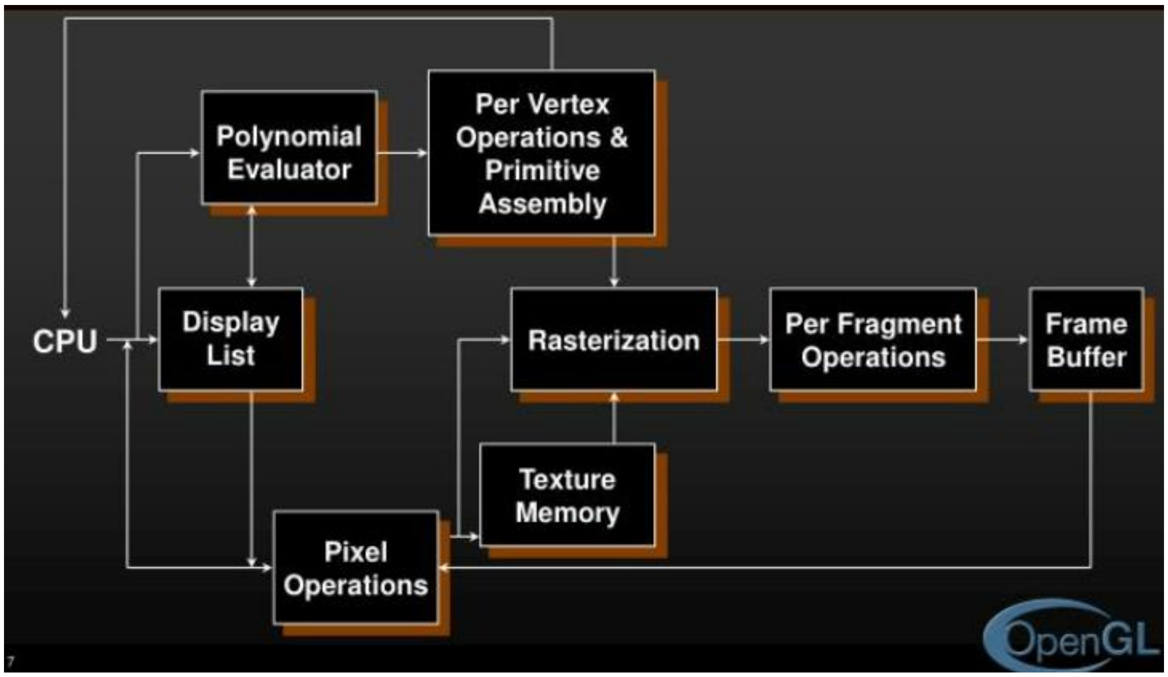
- OpenGL (Open Graphics Library) is a cross-language, cross-platform application programming interface (API) for rendering 2D and 3D vector graphics.
- The API is typically used to interact with a graphics processing unit (GPU), to achieve hardware-accelerated rendering.
- Silicon Graphics, Inc. (SGI) began developing OpenGL in 1991 and released it on June 30, 1992.
- It is extensively used in the fields of Computer-Aided design (CAD), virtually reality (VR), scientific visualization, information visualization, flight simulation, and video games.
- Many games also use OpenGL for their core graphics-rendering engines.
- OpenGL is just a graphics library; unlike DirectX, it doesn't include support for sound, input, networking, or anything else not directly related to graphics.

- Features of OpenGL
    - 3D Transformations – Rotation, scaling, translation
    - Color models – Values: R. G, B, alpha
    - Lighting
    - Rendering
    - Modeling
    - Device and platform independence: operating system independent
    - Abstractions (GL, GLU, GLUT)
    - Open source

Basic OpenGL Syntax

- Function names in the OpenGL basic library (also called the OpenGL core library) are prefixed with **gl**.
- Each component word within a function name has its first letter capitalized.
- Example **glBegin(), glClear(), glEnd(), glFlush(), glVertex2f(), glVertex3fv()** etc.
- Symbolic constants that are used with certain functions as parameters are all in capital letters, preceded by "GL", and component are separated by underscore.
- Example **GL_2D, GL_RGB, GL_POLYGON, GL_AMBIENT_AND_DIFFUSE**

OpenGL Architecture



OpenGL as a Renderer

- **Geometric primitives**
  - Points, lines and polygons
- **Images Primitives**
  - Images and bitmaps
  - Separate pipeline for images and geometry
- **Rendering depends on state**
  - Colors, materials, light sources, etc.

OpenGL APIs

- AGL, GLX, WGL
  - Glue between OpenGL and windowing systems
- GLU (OpenGL Utility Library)
  - Part of OpenGL
  - NURBS, tessellators, quadric shapes, etc.
- GLUT (OpenGL Utility Toolkit)
  - Portable windowing API
  - Not officially part of OpenGL

Header Files

- #include<GL/gl.h>
- #include<GL/glu.h>
- Include<GL/glut.h>

Sample Program:

```
#include<windows.h>
#include<GL/glu.h>
#include<GL/glut.h>
void MyInit()
{

   glClearColor(0,0,0,1);
   glColor3f(0,1,0);
}
void Draw()
{
   glClear(GL_COLOR_BUFFER_BIT);

   glPointSize(5);

   glBegin(GL_POINTS);
   glVertex2f(-0.5, 0.5);
   glVertex2f( 0.5, 0.5);
   glVertex2f( 0.5,-0.5);
   glVertex2f(-0.5,-0.5);
   glEnd();

   glFlush();
}

int main(int argC,char *argV[])
{
   glutInit(&argC,argV);
   glutInitWindowPosition(100,150);
   glutInitWindowSize(600,600);
   glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
   glutCreateWindow("Basic OpenGL Program");
   MyInit();
   glutDisplayFunc(Draw);
   glutMainLoop();

   return 0;

}
```