ATME College of Engineering, Mysuru

Department of CSE

Subject Presentation

BCS515A – COMPUTER GRAPHICS

Faculty: **Prof. Sandesh R**

Subject Details

Subject Code	BCS515A		CIE Marks	50
Number of Lecture / Hours	3		Exam Marks	50
Total No. of Lecture Hours	40		Exam Hours	03

Course outcomes:

The students should be able to:

CO1: Explain the fundamentals of computer graphics systems.
CO2: Develop event driven graphical applications by interfacing hardware devices.
CO3: Apply the Geometrical Transformations on geometrical objects.
CO4: Apply the concepts of viewing, lighting and shading on graphical objects.
CO5: Demonstrate algorithms for 2D graphical primitives.

ATME College of Engineering

Module 1

Course objectives:
CLO 1: Understand the basic principles of Graphical Systems.

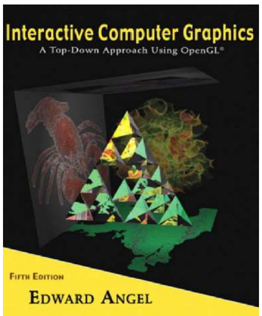
Graphics Systems and Models: Applications of Computer Graphics, A Graphics System, Images: Physical and Synthetic, Imaging Systems, The Synthetic-Camera Model, The Programmer's Interface, Graphics Architectures, Programmable Pipelines, Performance Characteristics.

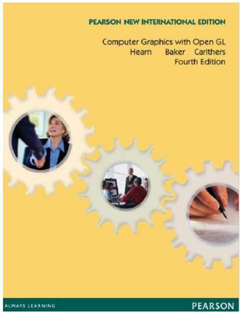
Course Outcome:
CO 1: Explain the fundamentals of computer graphics systems.

ATME College of Engineering

Text Books:

1. Edward Angel: **Interactive Computer Graphics A Top-Down Approach with OpenGL**, 5th Edition, Pearson Education, 2008.
2. Donald Hearn & Pauline Baker: **Computer Graphics with OpenGL Version**, 4th Edition, Pearson Education, 2011.



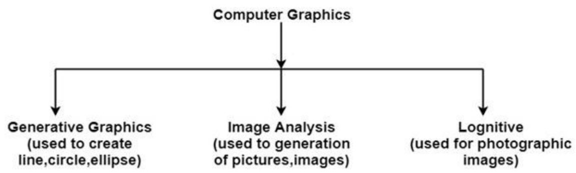


Module 1:

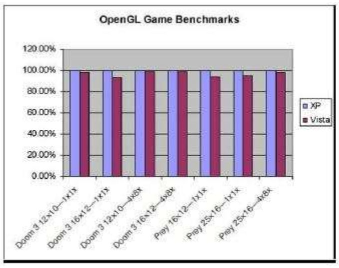
Graphics Systems and Models: Applications of Computer Graphics, A Graphics System, Images: Physical and Synthetic, Imaging Systems, The Synthetic-Camera Model, The Programmer's Interface, Graphics Architectures, Programmable Pipelines, Performance Characteristics.

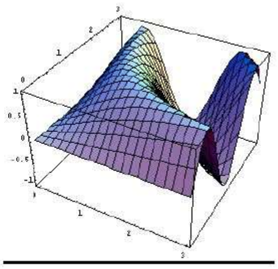
Basics of Computer Graphics

It is difficult to display an image of any size on the computer screen.
This method is simplified by using Computer graphics.
Graphics on the computer are produced by using various algorithms and techniques.
Computer Graphics is the creation of pictures with the help of a computer. The end product of the computer graphics is a picture it may be a business graph, drawing, and engineering.
Computer graphics is an art of drawing pictures, lines, charts, etc. using computers with the help of programming. Computer graphics image is made up of number of pixels. Pixel is the smallest addressable graphical unit represented on the computer screen.
In computer graphics, two or three-dimensional pictures can be created.



1. Graphs and Charts



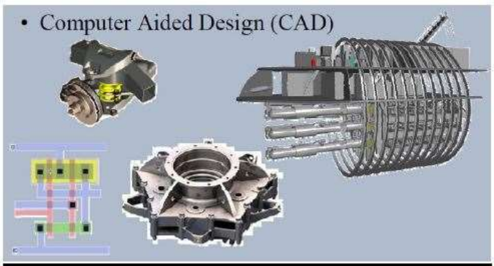


An e… ta graphs usual… common graphics application.

Graphs & charts are commonly used to summarize functional, statistical, mathematical, engineering and economic data for research reports, managerial summaries and other types of publications.

Typically examples of data plots are line graphs, bar charts, pie charts, surface graphs, contour plots and other displays showing relationships between multiple parameters in two dimensions, three dimensions, or higher-dimensional spaces

2. Computer-Aided Design



A major use of computer graphics is… ng and architectural systems.

CAD, computer-aided design or CADD, computer-aided drafting and design methods are now routinely used in the automobiles, aircraft, spacecraft, computers, home appliances.

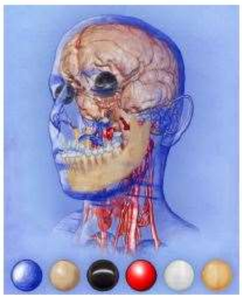
Circuits and networks for communications, water supply or other utilities are constructed with repeated placement of a few geographical shapes.

Animations are often used in CAD applications. Real-time, computer animations using wire-frame shapes are useful for quickly testing the performance of a vehicle or system.

3. Virtual-Reality Environments

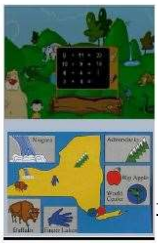
Animations in **virtual-reality environments** are often used to train heavy equipment operators or to analyze the effectiveness of various cabin configurations and control placements.
With virtual-reality systems, designers and others can move about and interact with objects in various ways. Architectural designs can be examined by taking simulated "walk" through the rooms or around the outsides of buildings to better appreciate the overall effect of a particular design.
With a special glove, we can even "grasp" objects in a scene and turn them over or move them from one place to another.

4. Data Visualizations



Producing … al data sets and processes is another fairly new application of computer graphics, which is generally referred to as scientific visualization. And the term business visualization is used in connection with data sets related to commerce, industry and other nonscientific areas.
There are many different kinds of data sets and effective visualization schemes depend on the characteristics of the data. A collection of data can contain scalar values, vectors or higher-order tensors.

5. Education and Training



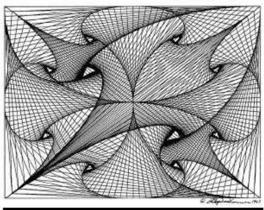
Computer gener... conomic & other systems are ofte...

Models of physical processes physiological functions, equipment, such as the color coded diagram as shown in the figure, can help trainees to understand the operation of a system.

For some training applications, special hardware systems are designed. Examples of such specialized systems are the simulators for practice sessions ,aircraft pilots, air traffic control personnel.

Some simulators have no video screens, for eg: flight simulator with only a control panel for instrument flying

6. Computer Art



The picture is ... s, which can simulate d...

Fine artists use a variety of other computer technologies to produce images. To create pictures the artist uses a combination of 3D modeling packages, texture mapping, drawing programs and CAD software etc.

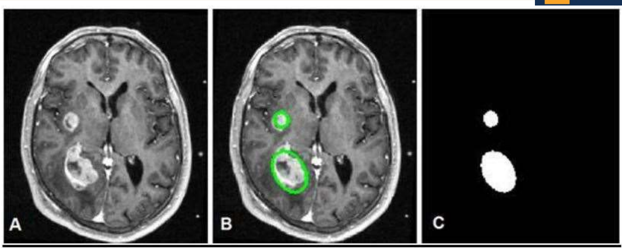
Commercial art also uses theses “painting” techniques for generating logos & other designs, page layouts combining text & graphics, TV advertising spots & other applications.

A common graphics method employed in many television commercials is morphing, where one object is transformed into another.

7. Entertainment

Televis mputer graphics methods.
Sometimes graphics images are combined a live actors and scenes and sometimes the films are completely generated a computer rendering and animation techniques.
Some television programs also use animation techniques to combine computer generated figures of people, animals, or cartoon characters with the actor in a scene or to transform an actor’s face into another shape.

8. Image Processing



The modification or interpretation of existing pictures, such as photographs and TV scans is called image processing.
Methods used in computer graphics and image processing overlap, the two areas are concerned with fundamentally different operations.
Image processing methods are used to improve picture quality, analyze images, or recognize visual patterns for robotics applications.
Image processing methods are often used in computer graphics, and computer graphics methods are frequently applied in image processing.
It is also used in computed X-ray tomography(CT), position emission tomography(PET),and computed axial tomography(CAT).

ATME College of Engineering

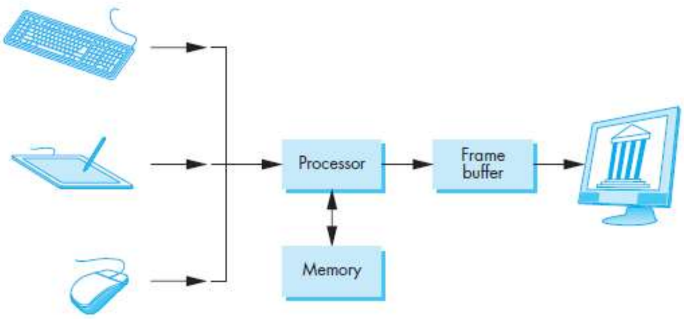
9. Graphical User Interfaces

It is common now for applications software to provide graphical user interface (GUI). A major component of graphical interface is a window manager that allows a user to display multiple, rectangular screen areas called display windows.
Each screen display area can contain a different process, showing graphical or non-graphical information, and various methods can be used to activate a display window. Using an interactive pointing device, such as mouse, we can active a displa window on some systems by positioning the screen cursor within the window display area and pressing the left mouse button.

ATME College of Engineering

A GRAPHICS SYSTEM

A computer graphics system is a computer system; as such, it must have all the components of a general-purpose computer system.



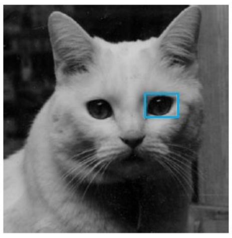
ATME College of Engineering

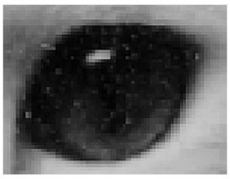
A GRAPHICS SYSTEM

Pixels and the Frame Buffer

Presently, almost all graphics systems are raster based. A picture is produced as an array the raster of picture elements, or pixels, within the graphics system.

Collectively, the pixels are stored in a part of memory called the frame buffer. The **frame buffer** can be viewed as the core element of a graphics system. Its resolution the number of pixels in the frame buffer determines the detail that you can see in the image.





ATME College of Engineering

A GRAPHICS SYSTEM

The depth, or precision, of the frame buffer, defined as the number of bits that are used for each pixel, determines properties such as how many colors can be represented on a given system.

For example, a 1-bit-deep frame buffer allows only two colors, whereas an 8-bit-deep frame buffer allows **2^8 (256)** colors.

In full-color systems, there are 24 (or more) bits per pixel.

They are also called true-color systems, or **RGB-color** systems, because individual groups of bits in each pixel are assigned to each of the three primary colors red, green, and blue used in most displays.

The conversion of geometric entities to pixel colors and locations in the frame buffer is known as rasterization, or scan conversion.

Today, virtually all graphics systems are characterized by special-purpose **graphics processing units (GPUs),** custom-tailored to carry out specific graphics functions.

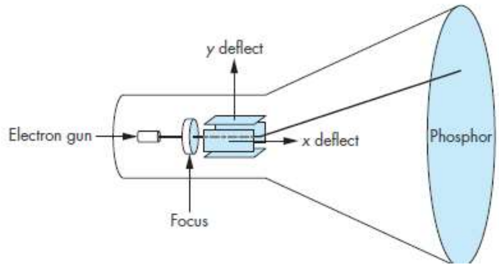
ATME College of Engineering

A GRAPHICS SYSTEM

Output Devices

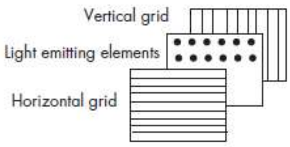
For many years, the dominant type of display (or monitor) has been the **cathode ray tube (CRT)**. Although various flat-panel technologies are now more popular, the basic functioning of the CRT has much in common with these newer displays.

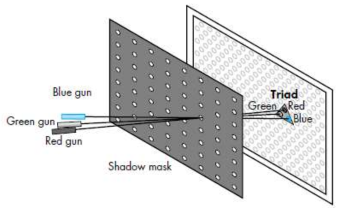
In a raster system, the graphics system takes pixels from the frame buffer and displays them as points on the surface of the display



ATME College of Engineering

Most color CRTs have three electron beams, corresponding to the three types of phosphors. In the **shadow-mask CRT**, a metal screen with small holes the shadow mask ensures that an electron beam excites only phosphors of the proper color.





Flat-panel monitors are inherently raster.

Although there are multiple technologies available, including light-emitting diodes (LEDs), liquid-crystal displays (LCDs), and plasma panels, all use a two-dimensional grid to address individual light-emitting elements.

A GRAPHICS SYSTEM

Input Devices

Most graphics systems provide a keyboard and at least one other input device.

The most common input devices are the mouse, the joystick, and the data tablet. Each provides positional information to the system, and each usually is equipped with one or more buttons to provide signals to the processor.

Often called pointing devices, these devices allow a user to indicate a particular location on the display.

Game consoles lack keyboards but include a greater variety of input devices than a standard workstation.

A typical console might have multiple buttons, a joystick, and dials.

IMAGES: PHYSICAL AND SYNTHETIC

Objects and Viewers

We live in a world of three-dimensional objects.

Two basic entities must be part of any image-formation process, be it mathematical or physical: **obje**

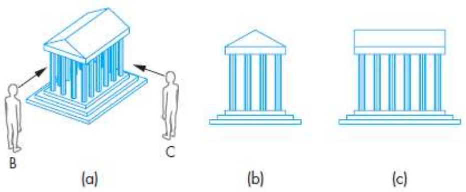


Figure 1.7 shows a camera system viewing a building. Here we can observe that both the object and the viewer exist in a three-dimensional world. However, the image that they define—what we find on the film plane—is two dimensional.

Light and Images

Light from the source strikes various surfaces of the object, and a portion of the reflected light enters the camera through the lens.

The details of the interaction between light and the surfaces of the object determine how much light enters the camera.

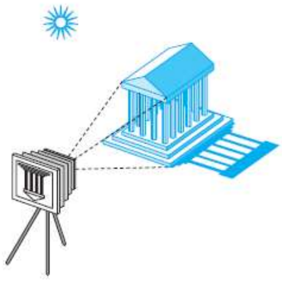


FIGURE 1.8 A camera system with an object and a light source.

ATME College of Engineering

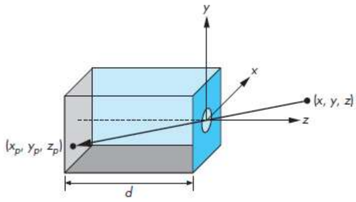
Two physical imaging systems: the pinhole camera and the human visual system.

The Pinhole Camera

The pinhole camera shown in Figure provides an example of image formation that we can understand with a simple geometric model.

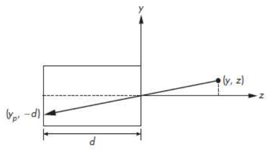
A pinhole camera is a box with a small hole in the center of one side of the box; the film is placed inside the box on the side opposite the pinhole. Initially, the pinhole is covered.

It is uncovered for a short time to expose the film. Suppose that we orient our camera along the z-axis, with the pinhole at the origin of our coordinate system.



We assume that the hole is so small that only a single ray of light, emanating from a point, can enter it. The film plane is located a distance d from the pinhole.
A side view (Figure 1.13) allows us to calculate where the image of the point (x, y, z) is on the film plane z =−d. Using the fact that the two triangles shown in Figure 1.13 are similar, we find that the y coordinate of the image is at y_p, where

$$y_p = -\frac{y}{z/d}.$$



A similar calculation, using a top view, yields

$$x_p = -\frac{x}{z/d}.$$

The point (xp, yp, −d) is called the projection of the point (x, y, z).
The field, or angle, of view of our camera is the angle made by the largest object that our camera can image on its film plane.

$$\theta = 2\tan^{-1}\frac{h}{2d}.$$

The Human Visual System
Light enters the eye through the lens and cornea, a transparent structure that protects the eye.
The iris opens and closes to adjust the amount of light entering the eye.
The lens forms an image on a two-dimensional structure called the retina at the back of the eye.
The rods and cones (so named because of their appearance when magnified) are light sensors and are located on the retina. They are excited by electromagnetic energy in the range of **350 to 780 nm.**
The sensors in the human eye do not react uniformly to light energy at different wavelengths. There are three types of cones and a single type of rod.
Whereas intensity is a physical measure of light energy, brightness is a measure of how intense we perceive the light emitted from an object to be.

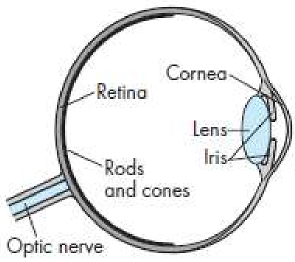
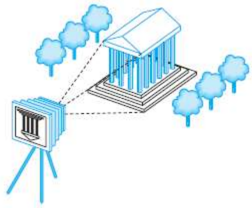


FIGURE 1.15 The human visual system.

Our models of optical imaging systems lead directly to the conceptual foundation for modern three-dimensional computer graphics.
We look at creating a computer generated image as being similar to forming an image using an optical system.
This paradigm has become known as the **synthetic-camera model**.



In this case, the viewer is a bellows camera.6 The image is formed on the film plane at the back of the camera. So that we can emulate this process to create artificial images, we need to identify a few basic principles.

First, the specification of the objects is independent of the specification of the viewer.
Hence, we should expect that, within a graphics library, there will be separate functions for specifying the objects and the viewer.
Second, we can compute the image using simple geometric calculations, just as we did with the pinhole camera.
Consider a side view of a camera and a simple object, as shown in Figure 1.17. The view in part (a) is similar to that of the pinhole camera.

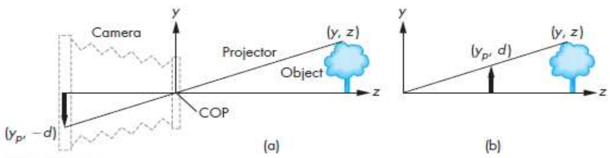


FIGURE 1.17 Equivalent views of image formation. (a) Image formed on the back of the camera. (b) Image plane moved in front of the camera.

Note that the image of the object is flipped relative to the object. Whereas with a real camera, we would simply flip the film to regain the original orientation of the object, with our synthetic camera we can avoid the flipping by a simple trick.
We draw another plane in front of the lens (Figure 1.17(b)), and work in three dimensions

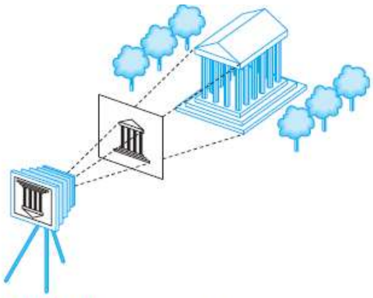


FIGURE 1.18 Imaging with the synthetic camera.

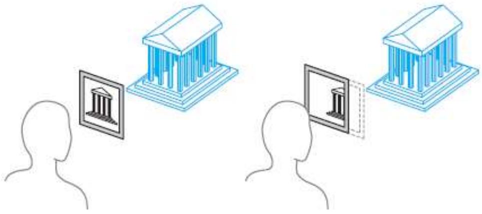


FIGURE 1.19 Clipping. (a) Window in initial position. (b) Window shifted.

ATME College of Engineering

THE PROGRAMMER'S INTERFACE

There are numerous ways that a user can interact with a graphics system. With completely self-contained packages, such as the ones used in the CAD community, a user develops images through interactions with the display using input devices, such as a mouse and a keyboard.

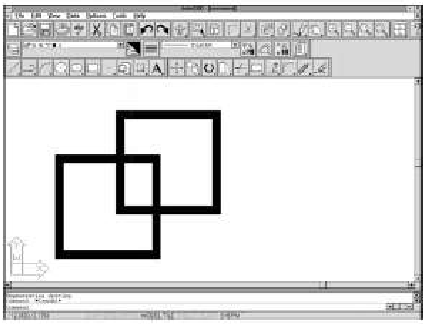


FIGURE 1.20 Interface for a painting program.

The interface between an application program and a graphics system can be specified through a set of functions that resides in a graphics library. These specifications are called the application programmer's interface (API). The application programmer's model of the system is shown in Figure 1.21.

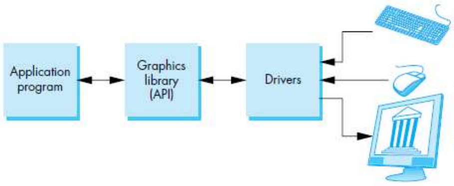


FIGURE 1.21 Application programmer's model of graphics system.

The application programmer sees only the API and is thus shielded from the details of both the hardware and the software implementation of the graphics library.

ATME College of Engineering

The Pen-Plotter Model

A pen plotter (Figure 1.22) produces images by moving a pen held by a gantry, a structure that can move the pen in two orthogonal directions across the paper.

The plotter can raise and lower the pen as required to create the desired image. Pen plotters are still in use; they are well suited for drawing large diagrams, such as blueprints.

We can describe such a graphics system with the following drawing

moveto(x,y)
lineto(x,y)

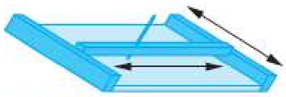


FIGURE 1.22 Pen plotter.

Execution of the **moveto** function moves the pen to the location (x, y) on the paper without leaving a mark. The **lineto** function moves the pen to (x, y) and draws a line from the old to the new location of the pen.

Here is a fragment of a simple program in such a system:

```
moveto(0, 0);
lineto(1, 0);
lineto(1, 1);
lineto(0, 1);
lineto(0,0);
```

This fragment would generate the output shown in Figure 1.23(a). If we added the code we would have the image of a cube formed by an oblique projection, as shown in Figure 1.23(b).

```
moveto(0, 1);
lineto(0.5, 1.866);
lineto(1.5, 1.866);
lineto(1.5, 0.866);
lineto(1, 0);
moveto(1, 1);
lineto(1,5,1.866);
```

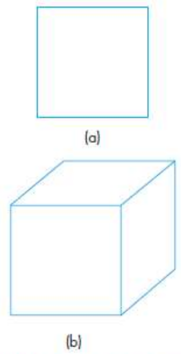


FIGURE 1.23 Output of pen-plotter program for (a) a square, and (b) a projection of a cube.

Three-Dimensional APIs

The synthetic-camera model is the basis for all the popular APIs, including OpenGL, Direct3D, and Open Scene Graph. If we are to follow the synthetic-camera model, we need functions in the API to specify the following: **Objects, A viewer, Light sources, Material properties**

The following OpenGL code fragment specifies the triangular polygon shown in Figure 1.24 through five function calls:

```
glBegin(GL_POLYGON);
    glVertex3f(0.0, 0.0, 0.0); /* vertex A */
    glVertex3f(0.0, 1.0, 0.0); /* vertex B */
    glVertex3f(0.0, 0.0, 1.0); /* vertex C */
glEnd( );
```

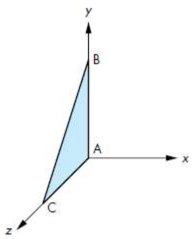


FIGURE 1.24 A triangle.

The Modeling–Rendering Paradigm

In many situations especially in CAD applications and in the development of complex images, such as for movies we can separate the modeling of the scene from the production of the image, or the rendering of the scene.

Hence, we can look at image formation as the two-step process shown in Figure 1.27.

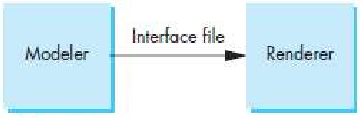


FIGURE 1.27 The modeling–rendering pipeline.

For example, consider the production of a single frame in an animation.We first want to design and position our objects.

This step is highly interactive, and we do not need to work with detailed images of the objects.

The interface between the modeler and renderer can be as simple as a file produced by the modeler that describes the objects and that contain additional information important only to the renderer, such as ligh sources, viewer location, and material properties.

On one side of the API is the application program. On the other side is some combination of hardware and software that implements the functionality of the API.

Early graphics systems used general-purpose computers with the standard von Neumann architecture.

Such computers are characterized by a single processing unit that processes a single instruction at a time.

A simple model of these early graphics systems is shown in Figure 1.28.

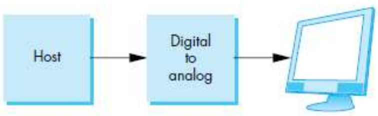


FIGURE 1.28 Early graphics system.

The display in these systems was based on a calligraphic CRT display that included the necessary circuitry to generate a line segment connecting two points.

Display Processors

These display processors had conventional architectures (Figure 1.29) but included instructions to display primitives on the CRT.

The main advantage of the display processor was that the instructions to generate the image could be assembled once in the host and sent to the display processor, where they were stored in the display processor's own memory as a display list, or display file.

The display processor would then execute repetitively the program in the display list, at a rate sufficient to avoid flicker, independently of the host, thus freeing the host for other tasks.

This architecture has become closely associated with the client–server architectures

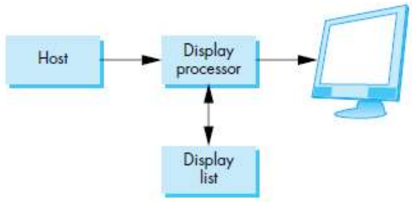


FIGURE 1.29 Display-processor architecture.

Pipeline Architectures

For computer-graphics applications, the most important use of custom VLSI circuits has been in creating pipeline architectures.

Pipelining is similar to an assembly line in a car plant. As the chassis passes down the line, a series of operations is performed on it, each using specialized tools and workers, until at the end, the assembly process is complete.

The concept of pipelining is illustrated in Figure 1.30 for a simple arithmetic calculation.

In our pipeline, there is an adder and a multiplier. If we use this configuration to compute **a + (b ∗ c),** then the calculation takes one multiplication and one addition

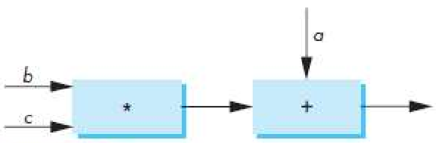


FIGURE 1.30 Arithmetic pipeline.

The Graphics Pipeline

We start with a set of objects. Each object comprises a set of graphical primitives.

Each primitive comprises a set of vertices. We can think of the collection of primitive types and vertices as defining the geometry of the scene. In a complex scene, there may be thousands even millions of vertices that define the objects.

We must process all these vertices in a similar manner to form an image in the frame buffer.

If we think in terms of processing the geometry of our objects to obtain an image, we can employ the block diagram in Figure 1.31, which shows the four major steps in the imaging process:

1. Vertex processing
2. Clipping and primitive assembly
3. Rasterization
4. Fragment processing

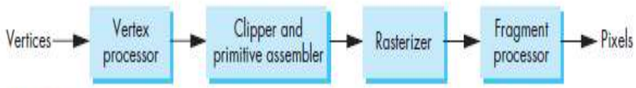


FIGURE 1.31 Geometric pipeline.

ATME atme College of Engineering

Graphics architectures have gone through multiple cycles in which the importance of special-purpose hardware relative to standard CPUs has gone back and forth.

However, the importance of the pipeline architecture has remained regardless of this cycle.

For many years, these pipeline architectures had a fixed functionality. Although the application program could set many parameters, the basic operations available within the pipeline were fixed. For example, there was only one lighting model for how to compute a shade using the specified light sources and materials.

Recently, there has been a major advance in pipeline architectures. Both the vertex processor and the fragment processor are now programable by the application program.

Vertex programs can alter the location or color of each vertex as it flows through the pipeline. Thus, we can implement a variety of light-material models or create new kinds of projections.

- There are two fundamentally different types of processing in our pipeline architecture.
- At the front end, there is geometric processing, based on processing vertices through the various transformations, vertex shading, clipping, and primitive assembly.
- This processing is ideally suited for pipelining, and it usually involves floating point calculations. The geometry engine developed by Silicon Graphics, Inc. (SGI) was a VLSI implementation for many of these operations in a special-purpose chip that became the basis for a series of fast graphics workstations.
- Today, graphics workstations and commodity graphics cards use graphics processing units (GPUs) that perform most of the graphics operations at the chip level. Pipeline architectures are the dominant type of high-performance system.

Important Questions

1. What is Computer Graphics? Illustrate different Applications of Computer Graphics in detail. (10 M)
2. Briefly explain FIVE major elements of Graphics system. (10 M)
3. Explain the working of CRT and Shadow Mask CRT in detail. (8M)
4. What is Objects and Viewers? Explain Images seen by three different Viewers. (6M)
5. Explain about a camera system with an object and a light source. (5M)
6. Explain the working principle of Pinhole camera. (6M)
7. Briefly explain about the synthetic camera model. (5M)
8. What is programmer's interface? Illustrate application programmer's model of graphics system. (10M)
9. What is pen plotter? Explain (6M)
10. Illustrate Graphics Architecture in detail. (10 M)
11. Explain the different steps carried out in Geometric pipeline. (10 M)
12. Explain
    Programmable Pipeline (5M)
    Performance Characteristics (5M)

Thank you