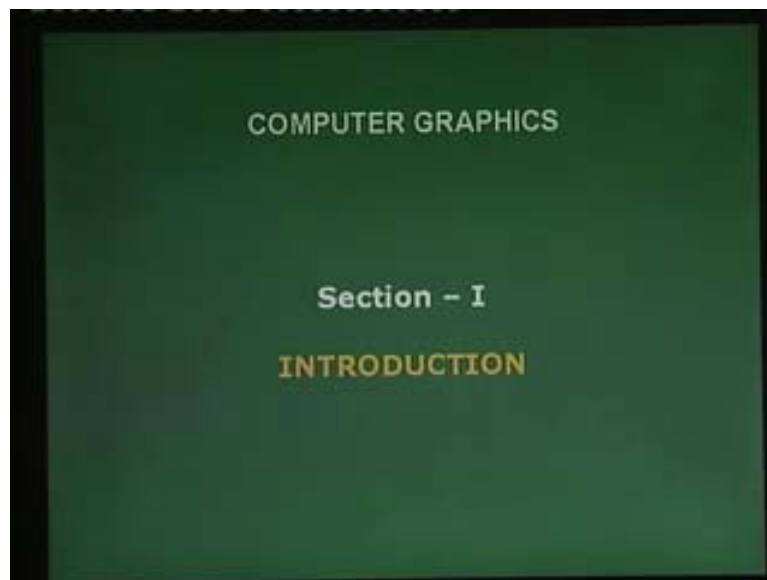


Computer Graphics
Prof. Sukhendu Das
Dept. of Computer Science and Engineering
Indian Institute of Technology, Madras
Lecture - 1
Introduction

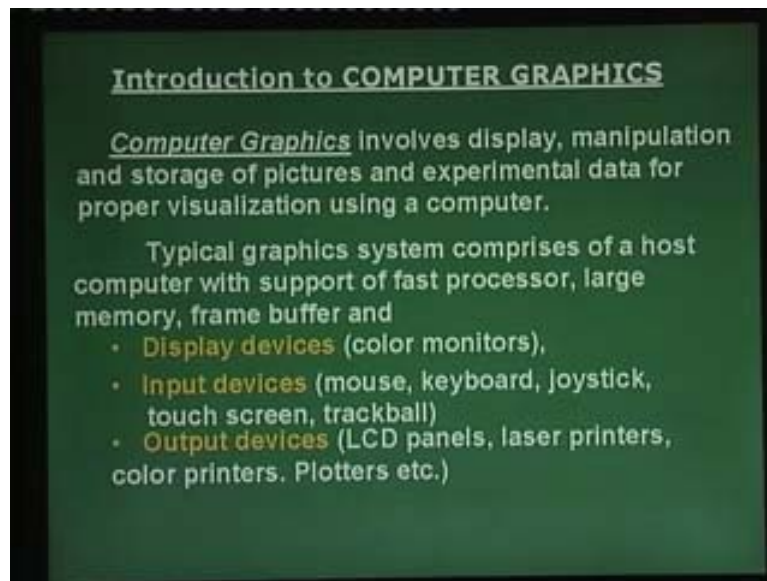
Hello and welcome everybody to the series of lectures on Computer Graphics. Today we have the introductory and the first lecture on Computer Graphics. This is a series of 40 lectures, so we start with the introduction on Computer Graphics. Computer Graphics is an area of computer science and engineering which play a very important role in almost every applications of computer software and use of computer science.

(Refer Slide Time 1:26)



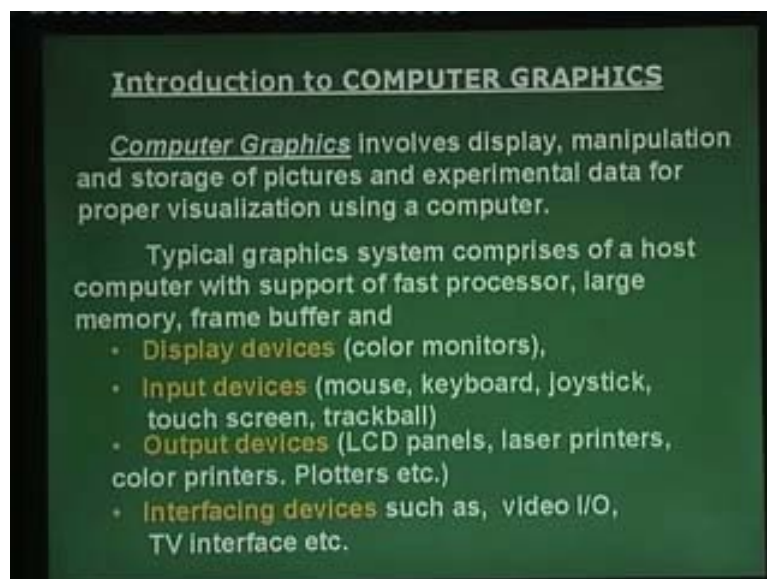
Computer Graphics involves display, manipulation and storage of pictures and experimental data for proper visualization using a computer. Typically a graphics system comprises of a host computer which must have a support of a fast processor, a large memory and frame buffer along with a few other crucial components. The first of them is the display devices. Color monitors are one example of such display device. We need a set of input devices. Typical examples are the mouse, keyboard, joystick, touch screen, trackball etc. Through these input devices you provide input to the computer and display device is an output device which shows you the image. There are other examples of output devices like LCD panels, laser printers, color printers, plotters etc.

(Refer Slide Time: 2:55)



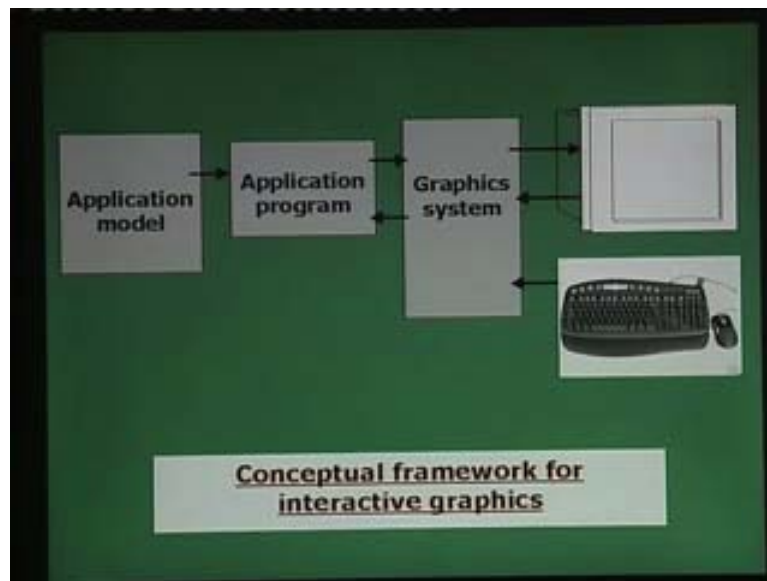
You can also have interfacing devices for a computer graphic system such as video input output to a system or an interface to the TV. So these are some of the basic requirements necessary to build a computer graphic system.

(Refer Slide Time: 03:14)



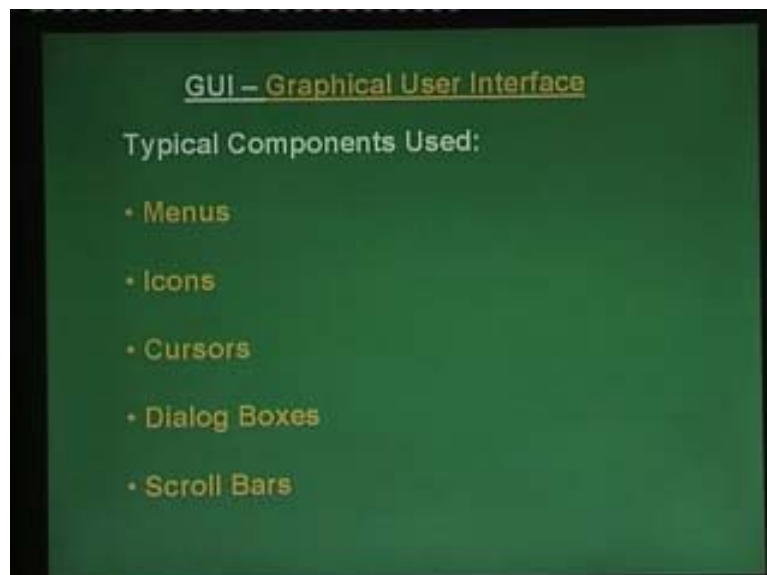
Well, as you see, now this flowchart gives the conceptual framework of interactive graphics system. On the left hand side you see an application model where the designer of Computer Graphics system or software engineer will put his design. That is he will model the object he is planning to display then he writes the program based on the application model. This application program will run on the computer graphic system then the output will come out to the screen and inputs could be obtained from keyboard or from the mouse. This is the typical framework.

(Refer Slide Time: 3:50)



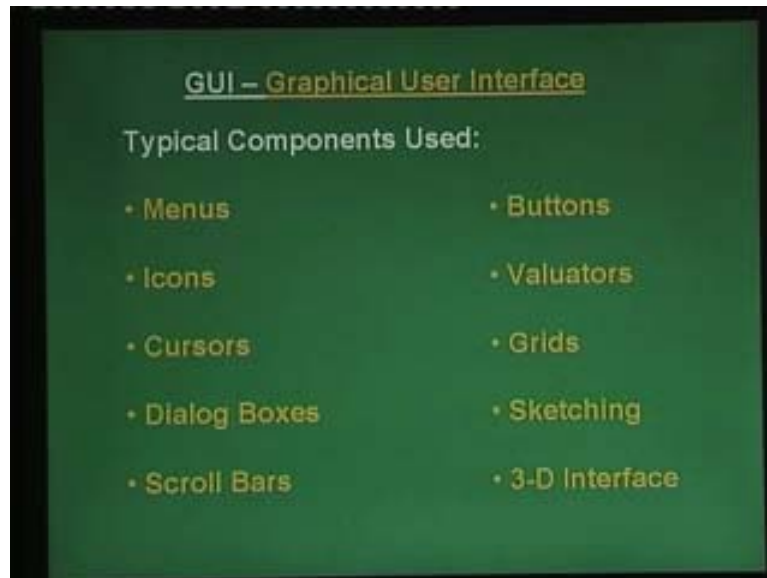
There are lots of applications of Computer Graphics. So we will see just a few examples of application areas of Computer Graphics. The first and most important of them is the GUI as it is called. What is a Graphical User Interface or GUI as it is called? It has various components. A graphical interface is basically a piece of interface or a program which sits between the user and the graphics application program. It helps the graphics system to interact with the user both in terms of input and output. Let us see few examples.

(Refer Slide Time: 4:48)



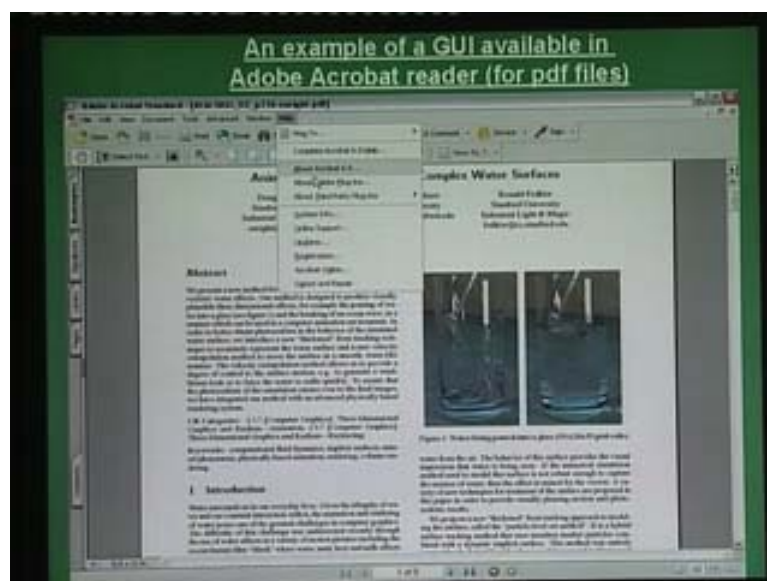
Typical components which are used in a Graphical User Interface are menus, icons, cursors, dialog boxes and scrollbars. We will see the few examples of these in the next slide. There are few other components which also could be used like buttons, valuators, grids, sketching and 3D- interface. Let us take an example to see what these different components are in typical graphical interface.

(Refer Slide Time: 5:03)



As you see, in the next slide, what I have done here is taken an example of Graphical User Interface from software called Adobe Acrobat reader which is used for reading adobe documents or PDF files. Now this is an example which has various components of the graphical user interface we have talked about. You see the list of menus on the top in terms of the components on the top screen bar. This is an example of what is called as a pull down menu.

(Refer Slide Time: 5:19)



You can click one of these and then what will happen is you will get a pull down menu, the adobe reader is another example. You can try this with various other menus and Linux based software starting from Word, PowerPoint users, Linux, X fig and Paint Brush in Windows also so these are some of the menus. Let us go back and see the next example; the icons and the cursor. Of course, probably we can make out the cursor floating around to a point to a particular location. These are certain examples of icons in terms of the options which will help you. The icons are small representative pictures of tasks of Computer Graphics application program which you are using. So these are certain examples of the icons, cursors, the dialog boxes.

The next example does not have a dialog box. The dialog box is an example of a small window which is used by a computer graphic system to interact with the user by sending error or text messages. A typical dialog box is used to send an error message to the user that you have provided the wrong input let us say. So that is an example of a dialog box.

The scroll bar is an example. On the right hand side you can see this window, the example of this image, what I have put here is a scroll bar which could be used to scroll up and down. when you need to move anything within the screen, right to left, up and down you can use scroll bars and it will give you the current status or position of where you are in terms of the screen location or the amount of data you are viewing. In this case the part shows here that you are somewhere on the top part of the page or top part of the document which you are viewing.

The other examples of the graphical user interface which is probably not in the example but I have taken is; buttons, valuator, grids and sketching. Just to give an example buttons could be considered equivalent to small icons.

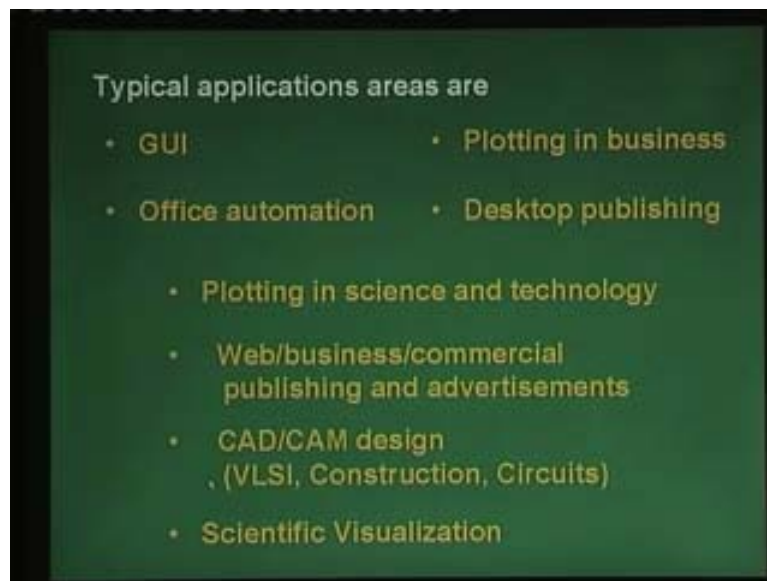
Valuator could be used to control the particular value which could be linear or circular you can assume it be a knob on a radio or a TV set which you control to control the volume or tune it to a certain frequency or it could be in the form of a slide which you could move from left to right or top to bottom to control a particular value to give as a input to the computer graphic system.

Grids are used in two dimensional graphics packages to align your objects along a set of specific coordinates or positions. You can switch it on and off and display them on the screen. Sketching is an example which is used to draw lines, arcs, poly lines and various other objects. The most difficult part of the Graphical User Interface is at the bottom of your screen is what is given as a three dimensional interface. It is easy to interact and handle with two dimensional objects but when you are interacting with the three dimensional objects you need a three dimensional interface to pick up one of the 3D objects from a two dimensional screen.

Essentially the computer monitor is just a two dimensional ray of pixels where the entire picture is projected and the picture could represent a three dimensional scene. So you need special type of interfaces to pick up or manipulate objects in 3D using a two dimensional interface. So you need special facilities for 3D interface to handle or manipulate three dimensional objects. The 3D interface is one of the hard topics of research in active areas and the part of the Graphical User Interface.

Let us come back to other typical application areas of Computer Graphics. Plotting in business we need to plot various curves in the form of pi-charts or 2D or 3D graphs. And probably in business applications you need to show the growth rate of the company, the expenditure, the profit and the various other types of work forces that you have been using, various economical types of data for business applications so you need plotting. You need to provide for office automation which are the standard packages of Word in Microsoft and the Microsoft Excel and PowerPoint. These are examples of office automation. Almost all software has Graphical User Interface along with desktop publication for designing documents, for designing PowerPoint slides. Like these you also need the help of graphical user systems.

(Refer Slide Time: 14:25)



Plotting in science and technology, we discussed about plotting in business and plotting in science and technology. It is absolutely necessary in almost all areas of study, any branch of science and engineering and technology and I was talking of 2D or 3D graphs in the form of line drawings, surface plots, contour plots, ISO contour plots, pi charts, so various types of plotting are necessary in science and technology. And you probably cannot do without computer graphic systems which has replaced the traditional draftsmen that is to take the help about one or two decades ago for the help of plotting.

Web business, commercial publishing and advertisements: you need to design very good advertisements on the web or on the TV shows and for commercial shootouts may be you post certain banners outside the city and you do take the help of Computer Graphics to make your advertisement very attractive and with that you can get a wide amount of publicity.

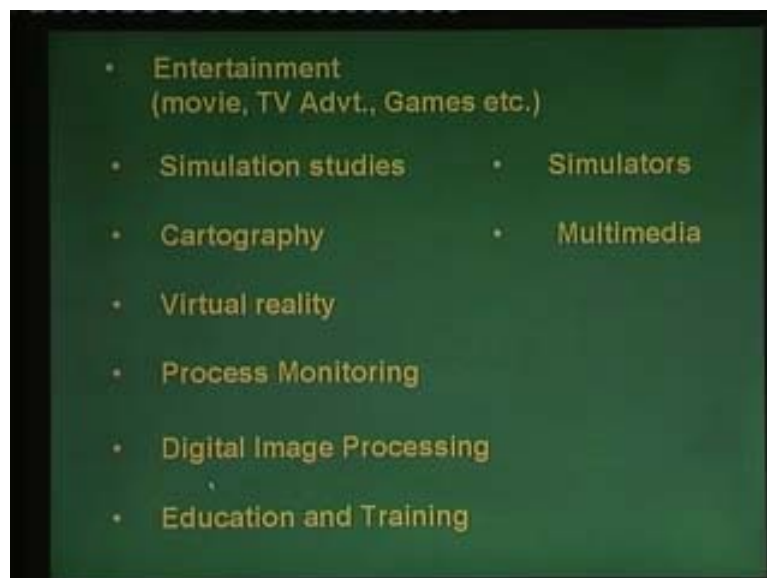
Coming to engineering applications we talk of CAD/CAM design. When we expand those terms it means Computer Aided Design and Computer Aided Manufacturing. A typical example of CAD/CAM is a vast scope in many fields of engineering and technology. But I have taken three examples in this case. VLSI: Very Large Scale Integration constructions in the area of architecture and civil engineering and circuits

designs in the area of electrical engineering. These are the areas of CAD/CAM designs where we actually manufacture a particular chip in VLSI or a printed circuit board. Let us say, in a circuit or an architecture, a bridge or road or it could be a building or a complex, a multi-storey complex we would like to design it in a computer graphic system to have a view and provide the interface and show what sort of design you have come up with. So there are various utilities of CAD/CAM designs in Computer Graphics which play a very crucial role.

Scientific visualization is almost similar to what we talk of plotting in science and technology. But we need to visualize something we can say as multi-dimensional data, something which requires animation. Let us say you would like to visualize the effect of a nuclear explosion so those sorts of visualizations are there and you need to understand certain patterns in data, it could be in genetic engineering, biological sciences or mechanical sciences or anywhere else. You would also like to have a visualization of the scientific data and you demonstrate the activity which is going on in a system using a typical Graphical User Interface.

Entertainment, very very important, it is a huge market in the industry where you need Computer Graphics system to design movie, TV advertisements, video games. In fact almost a majority of the market economy in computer graphic systems probably revolves around the entertainment world where we create not only animation or carton movies but we also make real time movies in between the real time characters. You make advertisements with the help of real characters and merge it with various types of steel frames, cartoon characters and things like that to create good advertisements. And of course, video games is a very lovely market not only for young children but even adults get attracted to lots of video games and similar kinds of entertainment.

(Refer Slide Time: 18:19)



Simulation studies and simulators, I will put both of these together. Well, this is another application I will say close to scientific visualization where you need to create a system with a good graphical interface for various applications. Simulation studies

and simulators include areas such as computation, fluid dynamics, heat and mass transfer, various other mechanical and chemical processors, it could include even studies about nuclear explosion and disaster management, damage prediction in a certain area, flight simulators. If we talk of simulators, flight simulators car racing, docking of a naval ship or a space shuttle, you would like to train your pilots and engineers before going on to a particular mission and you can use simulators to really test the skill of a particular person. It could be even in the field of sports one would like to use simulators to study the reaction time of a particular individual.

Cartography is an area which is connected to drawing, manipulation and creation of maps as I would say. It is an area in civil engineering, geology, geography, applied geophysics and Cartography is involved in the design of maps, creation of maps in geographical **engineering** systems and for other geographical applications. Basically it has to do with the creation of maps.

Multimedia is vast area again it combines text, audio images, video, Computer Graphics everything together and the presentation must be very very synchronized and Computer Graphics is a very important and integrated part of any multimedia presentation.

Virtual reality, well we discussed about video games. That is probably a very classical example of virtual reality. Or even animation movies or movies synthesized with the help of computer graphic systems which takes you to a completely different world and you can acquire big set of movies which have come out in the last decade based on virtual reality. Either the movie shows itself or it can help you to design good simulators and training kids for people before like the pilot of an aircraft, before they go on to actually test a system you can test their skills using virtual reality systems.

Process monitoring again in chemical and mechanical processors you can have a large plans where several activities are going simultaneously. We can have sensors fitted at various points which monitor the health and the flow of a particular assembly line. Or when a system is going on we need online monitoring to find out that the entire team is working true and in a correct form. The sensors are giving feedback simultaneously from different locations and if there is any wrong signal coming out from the sensor you can generate an alarm using a Graphical User Interface and alert the person who is probably monitoring it. So GUI could be used for process management as well.

Digital management processing is a vast area by itself. But I will probably like to mention areas connected to Computer Graphics which involves manipulations, storage and editions of pictures, enhancing the quality of the pictures, the contrast of the pictures, removing the noise, the blurriness and all those types of manipulations of digital pictures, saving them in different types of formats starting from ENP, PGN, TIF, JPEG you name it sort of a thing as all are coming under the field of Computer Graphics.

The last but not the least and for what we are also here is the Computer Graphics. It is also playing a major role in education and training not only in giving lecture series like the topics but today we are discussing but for any sort of education and training to individuals, professionals, students and may be even bureaucrats and politicians mainly where Computer Graphics systems could really become very helpful in

training certain professionals and educating people. So we come out of Computer Graphics applications and move over to certain application packages and standards available in Computer Graphics.

As we see here Computer Graphics packages and standards have evolved over time before, of course we go on to tell you what is the standard we will name a few of these; core graphics was the traditional standard of Computer Graphics and then we had the GKS and then the SRGP, I will expand these terms and then we had PHIGS SPHIGS AND PEX 3D and currently we have the OpenGL with or without ActiveX and Direct3D and the OpenGL is also available on X11 based system on Linux operating system.

(Refer Slide Time 18:33)



We will try to see what is the expansion of GKS. It is a Graphics Kernel System which was defined as a standard, almost the first in Computer Graphics by ISO and ANSI you can see the expansion of the word ISO International Standards Organization and ANSI stands for American National Standards Institute. Why are we taking about standards? We are taking because you need, a programmer needs a certain set of rules, constraints, procedures, syntax and function cause within his programming environment to build a graphic system. We have not come across to learn how to build a graphic system but if every person comes up with his own different type of syntax, formats and function cause, then when you design a system nobody else will be able to use your system and build another system on top of it so we need to talk of standards.

ISO and ANSI came up with the first standard of the Graphic Kernel System and it was very popular till it was probably replaced for a very short while by SRGP or Simple Raster Graphics Package as it is called. SRGP was very simple but was soon replaced by the standard called PHIGS Programmers Hierarchical Interactive Graphics System. I repeat PHIGS stands for Programmers Hierarchical Interactive Graphics System. This was quite popular I should say probably up to the late 90s and then currently we have the OpenGL, let us go back, the PHIGS replaced by current

standard as industry has accepted today called OpenGL graphics available on almost all system of Windows and Linux which are the most popular system used all over the world in industry as well as in educational institutions.

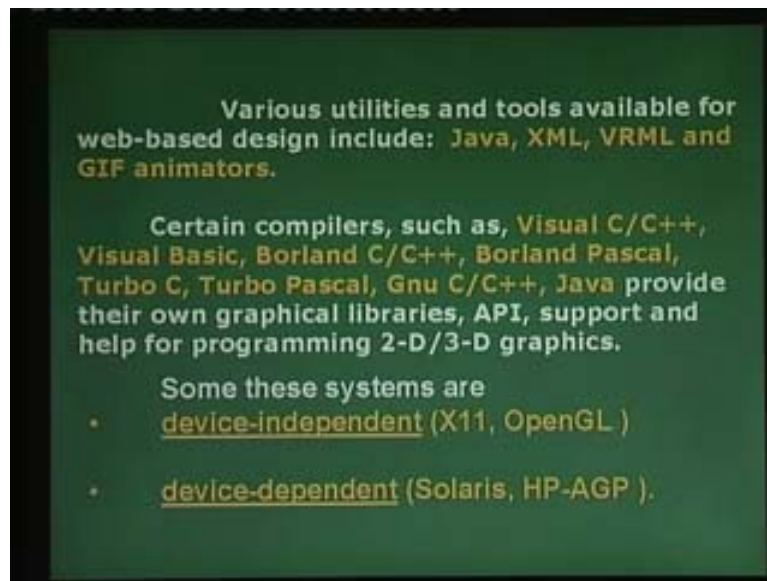
What you have to do is to download the device drivers, download the libraries, download the header files for OpenGL corresponding to your operating system and also the display adapter chord which you are using and then once you have the compiler when you talk about visual environment or dot net in windows system or in new environment or X11 graphics system is available on Linux you should be able to use OpenGL as a standard and write your own graphics, programs or packages. So this is how it has probably evolved over time, you do not need to know all of them, we will stick to OpenGL standards whenever you use comments and syntax for giving examples of graphic system. Sometimes you may use PHIGS also because just a few years back that was one of the popular standards.

(Refer Slide Time: 22:03)



And I was saying sometime back that these standards are available in almost all systems. Of course we have almost forgotten the use of the DOS or Distributed Operating System most commonly used operating systems based on Windows or Linux and the OpenGL, PHIGS are available on both. You also have graphic systems and operating systems such as OS/2, silicon graphic system, SunOS, Solaris, HP-UX these are the old systems and Mac and DEC-OSF. But the most common platform are Linux and Windows but we should not forgot the silicon graphic systems and the Solaris which have been also quite popular in many parts of the world. Various utilities and tools are available for computer graphic systems and especially for web-based design which includes Java, XML, VRML and GIF animators.

(Refer Slide Time: 23:58)



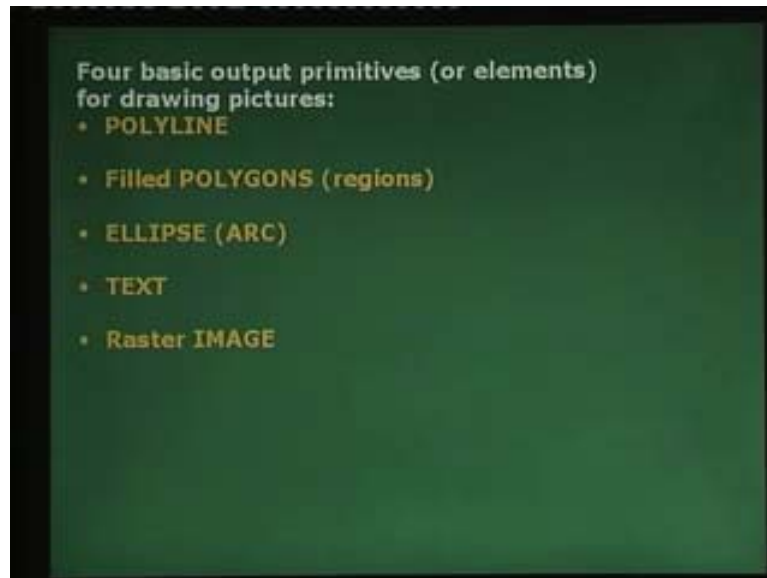
These are mostly examples of tools which help you to create nice web pages I should clarify here. But when you are designing a graphic system you are not probably only interested in a web page design but you are interested to come up with say a computer game or demonstrate certain facts through virtual reality or simulators or simulate a certain scenario through animations. So you need an operating system with graphics library support and pick up a standard to use it. So most compilers starting from the current visual CC plus plus visual basic, the Borland's turbo C Pascal, even new CC in Linux and java provide their own graphics libraries, API or Application programming environment and support and help for programming for both 2D as well as in 3D graphics. So you have to ensure that the proper operating systems, the proper compilers and correspondingly the proper libraries and APIs are all installed for your use to help you to design the computer graphic system.

You must be careful with some of these systems. There are basically two types, some of them are device independent and some are device dependent. I should clarify here the difference between the two examples of device independent or X11 systems or OpenGL systems.

Device dependent systems are the SGI systems and the old Hewlett Packard or advanced graphics packages probably which nobody uses these days. Device dependent systems had a drawback in the sense that once you design your own graphical computer graphic application program on a particular system it is almost impossible, it is very very difficult to take that program and run it on any other system. So those are examples of device dependent system where if you program it on one particular platform, on one particular system it will run almost entirely on identical platforms, identical machines, identical systems. This is not the case for device independent systems. In device independent systems if you design a Computer Graphics program in a X11 system or an OpenGL system either it is Linux or Windows almost with very little change or no change in certain occasions you should be able to take that program and run it on any platform, any machine which has windows or Linux with OpenGL support, X11 supports OpenGL as well.

You should be able to run such programs anywhere on a device independent system. So that is the difference between a device dependent one where you cannot run on any system, you have to run it on only that system or exactly identical systems whereas device independent you should be able to run it on any system. You can change your keyboard, your mouse, your monitors; you can change your operating system on platforms, you should be able to run the device independent programs on any system independent of whatever type of devices you have been using.

(Refer Slide Time: 27:10)



What you draw? We are talking of standards, platforms, device dependent, device independent and all that. There are basically four basic output primitive elements for drawing pictures. You can draw a very complicated picture, you need to draw of course for most applications. But you basically draw almost any picture with a very few fundamental operation or what we call as output primitives. We will clarify output primitives later on but assuming for the time being that you hold on and look at these examples of what are called primitives.

A few points to be clear, poly lines, you can draw a line from a point x_1y_1 to a point x_2y_2 . Take a graph paper plot a point x_1y_1 and plot a point x_2y_2 and then what you do draw a line so that is the simple example of drawing a line. What is a poly line? You can have a set of n points with coordinates x_1y_1 x_2y_2 x_3y_3 and so on up to x_ny_n in any order does not matter and the poly line command or the output primitive will draw lines from the first point x_1y_1 to a point x_2y_2 and then connect x_2y_2 to x_3y_3 and so on. It will connect the last set of pair of points basically connecting the points in that order as given in the function. That is an example of a poly line.

The next feature or next output primitive which we see after poly line is called the filled polygon or also filled region you can call. But the technical term is a polygon or a filled polygon. A polygon can be filled or it may not be filled. But what is the difference between the poly line and a polygon? The difference is the following; in a poly line we discussed about a set of n points being joined by lines. The same set of n points could be for a polygon line. But in the case of a polygon the last and first point

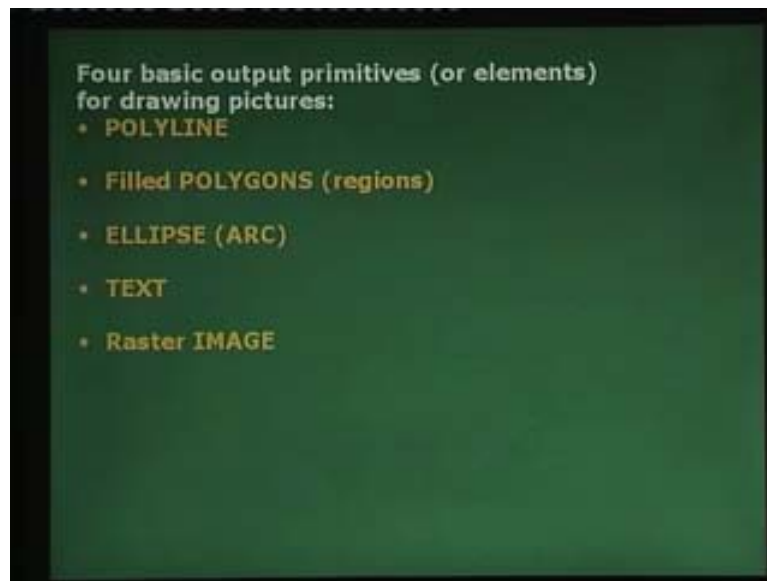
are joined together to enclose the region. You remember, in a polygon we started from x_1y_1 to x_2y_2 and we stopped at x_ny_n to draw a line. We kept on drawing lines and in the case of a polygon we take the last point also and connect it or join it with the first point so that will be the enclosed region. That is why it is also called a region but polygon is probably the proper terminology and then you can also fill it. Of course you will see in the algorithms how to fill the polygons with certain shades, textures and colors when the particular topic of drawing polygons and scanning polygons will come but that is also the functionality which is provided as an output.

The next primitives are, we talked of lines where we should be able to draw curves. And if we go back the output primitive for drawing curves is basically to draw an arc or an ellipse. So you should be able to draw an arc or an ellipse. Remember, you should bother about drawing a circle where geometrically if you see the equation of a circle and ellipse, the circle is a very special case of drawing an ellipse. So once you are given the option of drawing a ellipse or part of the ellipse as an arc you should be able to draw any arc, you should be able to draw a circle as well as ellipses so that primitive must be there.

In addition, you should also be provided an option to text with various kinds of fonts, various types of sizes, various types of scalable fonts and it could be in bold, italic font and all that with various such options. You should be provided an option to draw text and position it anywhere in any arbitrary orientation that option should be there in a Computer Graphics system. And the last but not the least is to draw a Raster IMAGE.

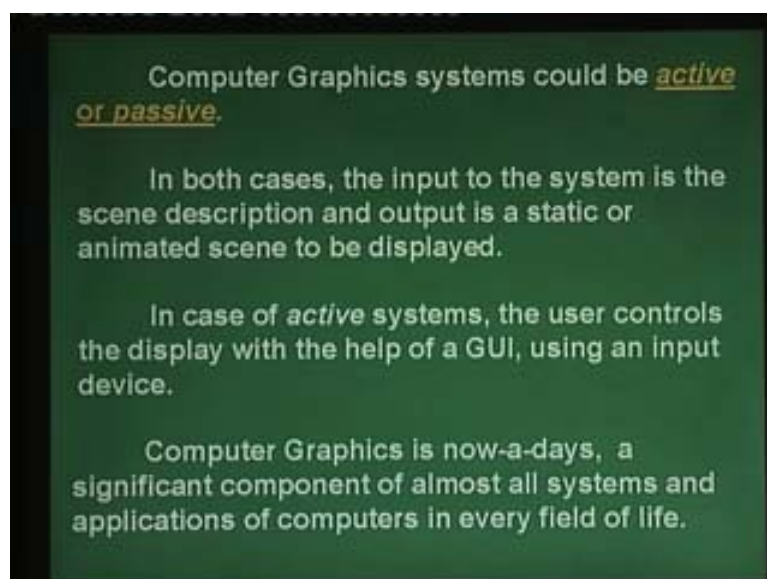
You should have the option to open a Raster IMAGE, a digital image, it could be stored in any format. We discussed about a few of these examples, format in applications of Computer Graphics or image processing where the format could be JPEG format a DMT format, a PGM or TIF but you should have the provision to read a particular digital image, display it on the screen, edit it, manipulate it, colors, give shades and whatever it is and then you can save it back, save it back to the disc or file. These options must be there as output primitives and once you have these I can almost assure that you can draw any picture which includes animation. That is of course a special feature but let us talk of static feature. Now, actually you should able to draw any complicated static picture once you have all these five basic primitives available on any Computer Graphics package or standard.

(Refer Slide Time: 30:32)



The four major areas of Computer Graphics you will discuss in various lectures in a sequence later on but not necessarily in this sequence and they will be display of information, design and modeling, simulation and user interface. We discussed about user interface already. Throughout the lecture we will talk about design and modeling which is the important part and in fact in the next lecture we will talk about display devices and then display of information. And of course to make the entire package work you must simulate it and make the application program work. These are the four major areas where people work and also do research in the field of Computer Graphics. Computer Graphics system could be active or passive. That is another terminology which you must know.

(Refer Slide Time: 33.19)



Let us read the lines, as you see here in both cases the input to the system is the scene description and the output is a static or animated scene to be displayed. That is true

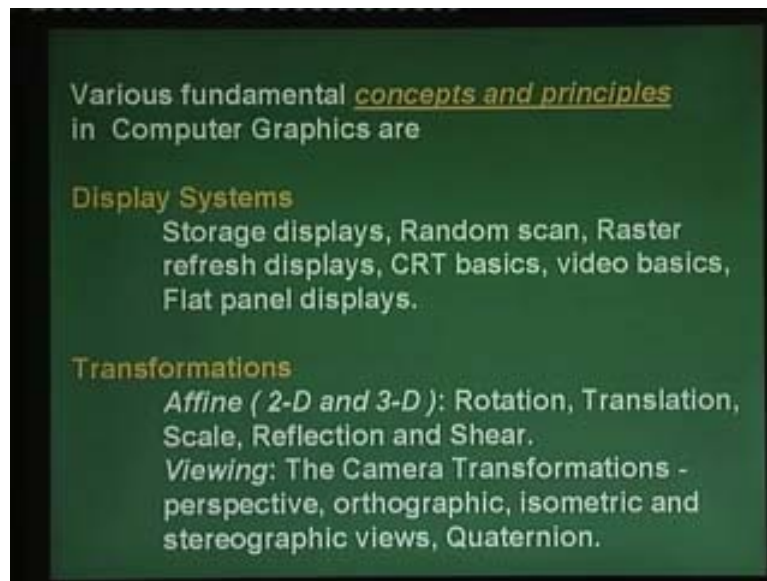
for both active and passive systems. Your input is the scene description and the output is some picture, a static picture or animated picture. Then what is the difference? The difference is that in case of active systems the user controls the display with the help of a graphical user interface using an input device and that is only possible in active system. In passive system you do not have any control, you have to watch whatever is shown to you.

A typical example of a passive system, whatever you see in your TV, you can change the channel but you stick to the particular channel, you are forced to see whatever is broadcasted by that particular channel. You can switch off the TV and go to another channel but you do not have control of what is being viewed on the screen or what is being telecasted on the screen or projected on the screen by the channel broadcaster. So that is the example of a passive system.

Example of an active system is very easy, the video games where almost you can decide where you want to go by giving an input through the mouse, through the keyboard, through the joystick is the example of an active system where the user play a prominent role and the pictures are shown depending upon the inputs given to the user. Although, the sequence is pre decided but you can provide many options for the games and quite a bit of randomness in the animation and the task which the video games will actually ask the user to do. It could be a very good active system in order to a passive one which you might think.

Last but not the least of Computer Graphics introduction is, Computer Graphics is now-a-days is a significant components of all system applications of computers in every field of life. Right from the point the computer boots up the operating system comes up and the login prompt is provided to you then you log in inside the screen see your desktop options in windows and Linux everything is controlled by a Computer Graphics applications package or Computer Graphics software it is there in every field of life now which involves of course computers, TVs, video shows and movies and one cannot think of systems without Computer Graphics. Let us look at a list of the major concepts and principles of Computer Graphics which we will go through this course.

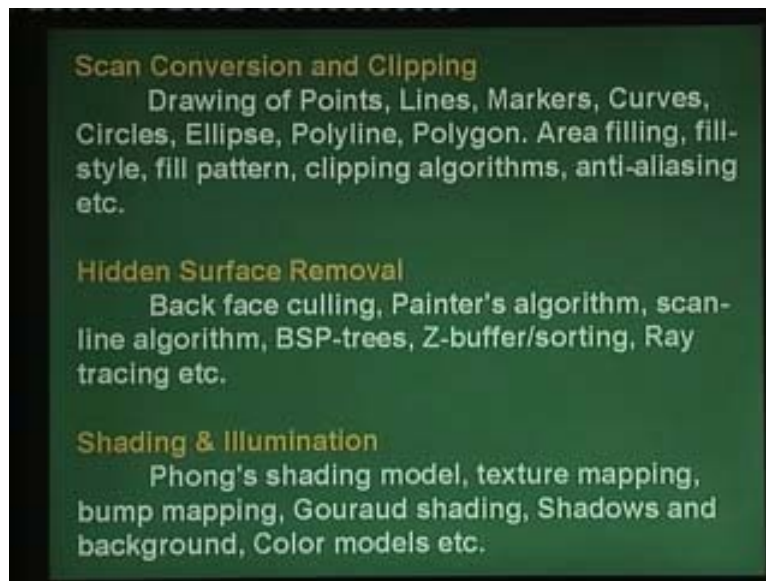
(Refer Slide Time: 33:55)



This is an introductory course but let us move on to certain topics. Almost in this order we will talk of displays system in the next introduction lecture and in display systems we will talk of storage displays, random scan displays, raster refresh displays, CRT basics, video basics and flat panel displays. We will try to cover as much as possible with display devices and videos in the next lecture after introduction. Then we will move on to transformations, manipulations of Computer Graphics and computer graphical objects dealing with transformations which mean manipulation objects. And mathematically we have to learn about affine transformations which involve rotation, translations, scaling, reflection and shear then comes viewing transformations. The camera transformations in terms of various types of perspective projection, various types of orthographic, isometric, stereographic views the way we human beings see. And the corresponding quaternion of the mathematics is associated with transformations.

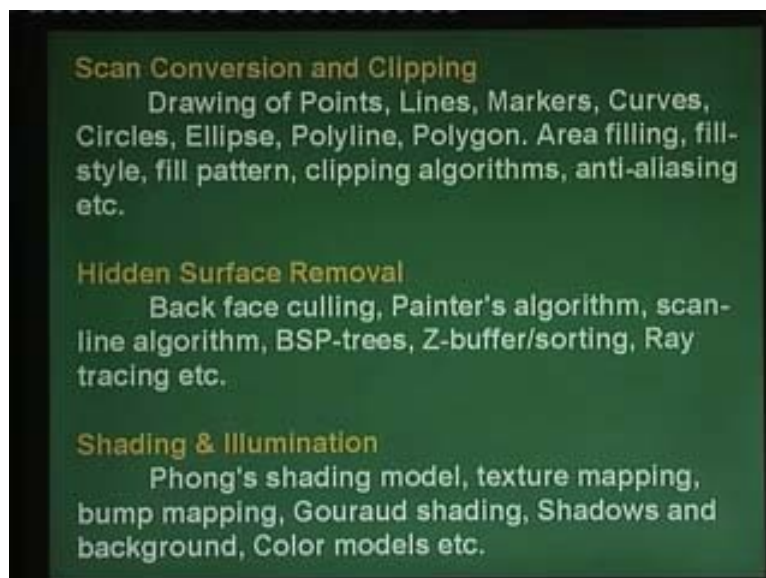
Next we will see on how to draw a line. We discussed about an output primitive to draw a line or a curve as a function but actually how it is implemented in a graphics package we must also know. You can use the command to draw a line but you must know how the command is working inside the system to draw a line. So it involves learning about scan conversion and clipping, drawing of points, lines, line markers, curves, circles, ellipse, poly line, polygon, area filling, fill style, fill pattern, clipping algorithm and anti-aliasing. So there are various topics as you see on the board here for scan conversion and clipping. It involves many of these options.

(Refer Slide Time: 35:48)



We will study those, the algorithms involved in scan conversion algorithm and hidden surface removal is the next task. We are moving from 2D to 3D now where we talk of three dimensional structures and objects. When you see a three dimensional object in front of you it is evident that you will be able to only see the faces which are in front of you. You will not be able to see the face which is on the back side. So how to come up with the algorithms using geometry, mathematical model to find out which part of the surface of an object is facing you and which part of the object is behind you. Those algorithms involve the concept of hidden surface removal. There are various ways and means and by which you can do by it hidden surface removal. The first and easiest is back face scaling.

(Refer Slide Time: 36:47)

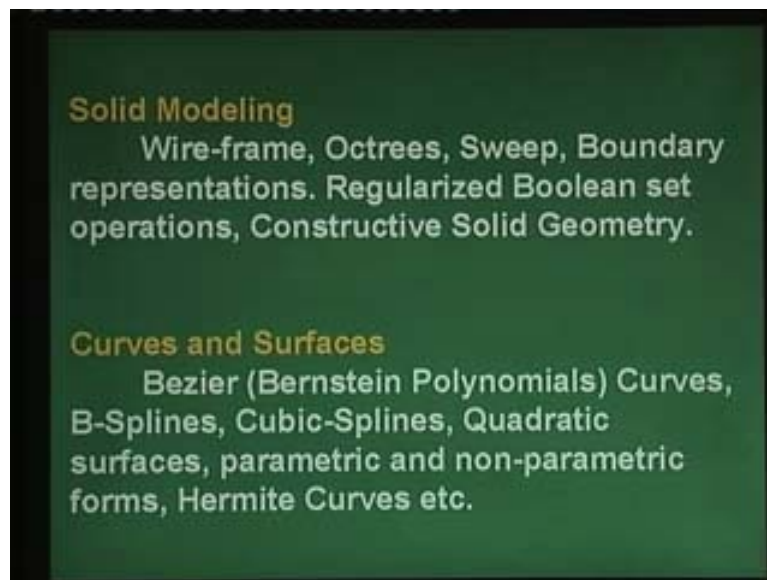


As you see on the board the next few algorithms are Painter's algorithm, scanline algorithm, BSP-trees, Z buffer/sorting, Ray tracing etc. The most popular among

these are the Z buffer and Ray tracing. But people also use back face culling and BSP trees. After you remove the hidden surfaces you need to shade and eliminate the surface facing the viewer or facing the design. It involves shading and illumination. There are various shades of model; Phong's shading model, texture mapping, bump mapping, Gouraud shading, Shadows and background, Color models etc. So when you move to that chapter of lecture you will study the chapter of shading and illumination.

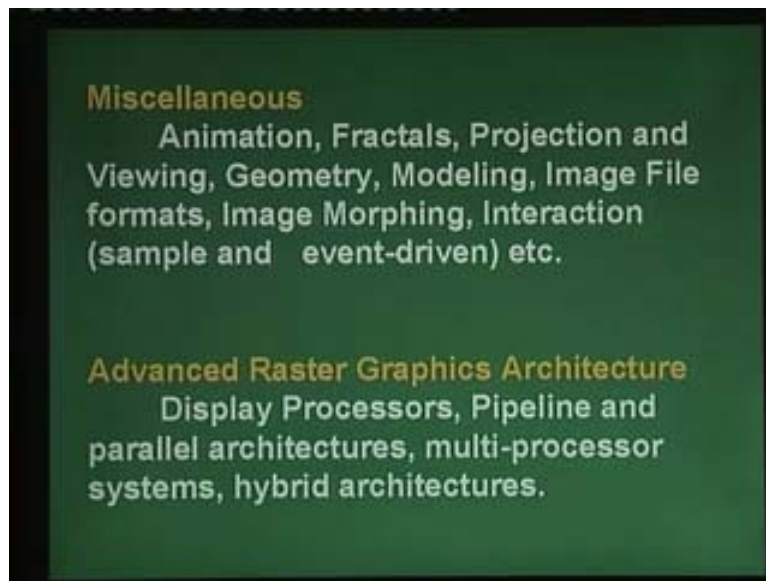
Solid modeling is we need to model a solid object. There are various types of methods and algorithms and techniques by which you can model a solid object. I will give an example at the end of the talk today about wire-frame object. But we can use other representations like Octrees, Sweep representation, Boundary representation, regularized Boolean set operations and constructive solid geometry or what is also known CSG.

(Refer Slide Time: 37:40)

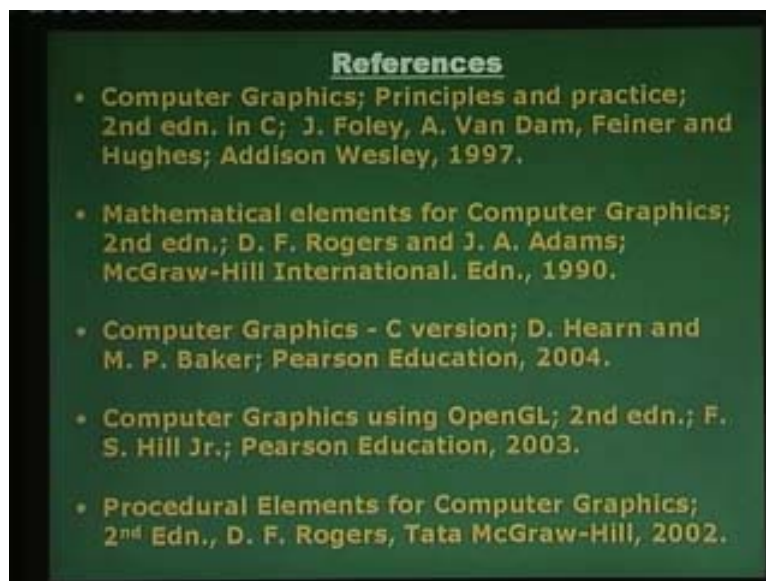


In the last you also move towards curves and surfaces which involve special mathematical functions like Bezier or Bernstein polynomials curves, B-Splines, Cubic-Splines Quadratic surfaces, parametric and non-parametric forms, Hermite curves etc. Depending upon the time I will also cover some advanced aspects such as concept of Animation, Fractals, Projection and Viewing, Geometry, Modeling, Image File formats, Image Morphing, Interaction sample and event-driven etc.

(Refer Slide Time: 38:12)



(Refer Slide Time: 38: 49)



Also, if you have some background of computer architecture you will be able to follow the parts of my lecture which talks about Display Processors, Pipeline and parallel architectures, multi-processor systems, hybrid architectures, you need some good fantastic highly optimized high speed architectures to make your graphics program work really fast so that the user does not have to wait for a certain event to occur after he provides the input. These are the list of references or books on Computer Graphics which I will ask you to go through as a part of this lecture. You can at least purchase one of them.

The most advanced one is the first one which is Computer Graphics Principles and Practice by J. Foley A. Van Dam and Hughes that uses the fixed standard. The next book is by Rogers and Adams which is taking about the Mathematical Aspect of Computer Graphics. The third book in fact is the book by Computer Graphics C

Version Hearn and Baker which is simple and easy to read as it has lot of good illustrations and you can use that book as well. If you are talking about the current standard of OpenGL I would recommend the fourth book. There are good examples and programs that you can take on and try it on your system to see whether it works and which gives you confidence. And the last but not the least, is a book by Rogers on Procedure Elements of Computer Graphics. It is a good mixture of mathematics as well as concepts and algorithms on Computer Graphics.

I do believe one should have one of these books, if possible two books at your disposal also to have a reading material in addition to listening to these lectures to understand Computer Graphics. To wind up the lecture I have brought today some examples and illustration of computer graphical objects. I will not have time today to tell you how these graphic objects are created but we will see examples of what computer graphical systems can do or generate. I discussed of a 3D solid modeling in one of the course contents. This is an example of wire-frame model of a sphere which uses the sweep representation.

If you see the sphere it looks like a sphere, how this has been created? Assume the surface of the sphere is a perfect sphere. We have learnt that in basic school geography and you can start to think of latitudes and longitudes on the surface of the earth. Remember, the vertical circles on the surface and the horizontal lines which are equidistance sort of a thing, and of course the vertical circle which intersects at the north and the South Pole probably that is the best example. So you have to take those sorts of lines, latitudes and longitudes and the intersection points of those latitudes and longitudes give the vertices of the wire-frame model or basically the 3D coordinates of these spheres.

Just to see an example of what I mean we will go to the next slide in the left hand side which has come out now. As you see some of the vertices have been marked by a light red color, a light reddish color and those are the vertices which are the intersection of the vertical lines and the horizontal parallel lines or something like the latitudes and the longitudes of the hard surface in the map of the earth. Of course it is a very fine grid but those intersection points are the most crucial for storing or representing the sphere. The 3D coordinates of the intersection points are basically stored to represent the wire-frame model of a sphere using a sweep representation.

What is a sweep representative, you do not have to worry about this now. Just to give an example you can take a circle and rotate the circle in 3D vertically that will give some example of the sweep representation. But you can generate other types of objects as well. If you see the third figure I have joined a set of three vertices, a few of them to show a set of triangles. now what you need is after you have the wire-frame model you need to apply techniques like hidden surface removal, then you need to apply illumination and shading and to do that you need to approximate the surface of a curved object like a sphere or even a planer surface could be a rectangular parallelepiped or a cube with a set of triangles or quadrilaterals or rectangles.

We have taken the rectangular shape, that is the best approximation and that smaller patch which is curved is approximated basically by this triangle or linear patch. Once the coordinates of the triangle are known you can apply the algorithm of hidden surface removal, back face culling, illumination and shading to actually throw of the

hidden surface which is on the back side as viewed. Now you do not need to shade the surfaces or the patches which are on the back side of the object, which is not visible to the viewer from the front. Now you need to shade. So, all those shading algorithms and back face culling rely on the area, the surface normal and the orientation of those rectangular or triangular patches. These are the examples of the triangular patches which should be used for shading parts of this sphere, just an example. These are some examples of simple 3D objects.

This is an example of four objects here. Well, you can assume that at the bottom there is a flat table type of structure or a small very thin rectangular parallelepiped and on top of that there are three objects: a cube, a sphere, and a pyramid. You can think this pyramid to be a cone but it does not have a rectangular base. I will say this is a pyramid with an octagonal base. A highly approximated sphere and in a cube there are four such objects and these are some examples of primitive objects that you can use, combine and manipulate to create other objects. Well, these are simple objects and then you can of course shade them. But to create complex objects let us take another example like this, this is an object which is already shaded, it is a pyramid, and a perfect pyramid with a triangle base but it has a hole. We will see an example of how to create such a hole within an object with different examples. But this is an example of an object which is shaded it is slightly complex in the sense that it has a hole and that is an example.

We will see the next example. These are another set of primitive objects which are very commonly used to generate other complex 3D structures. Examples are; a cylinder, a cube, a sphere, a perfect cone and a triad, a 3D triad. And these are examples of five objects which are used and combined and manipulated to generate other complicated structures. And in this case you can use sweep representation or a wire-frame to have the model of the object and then shade it.

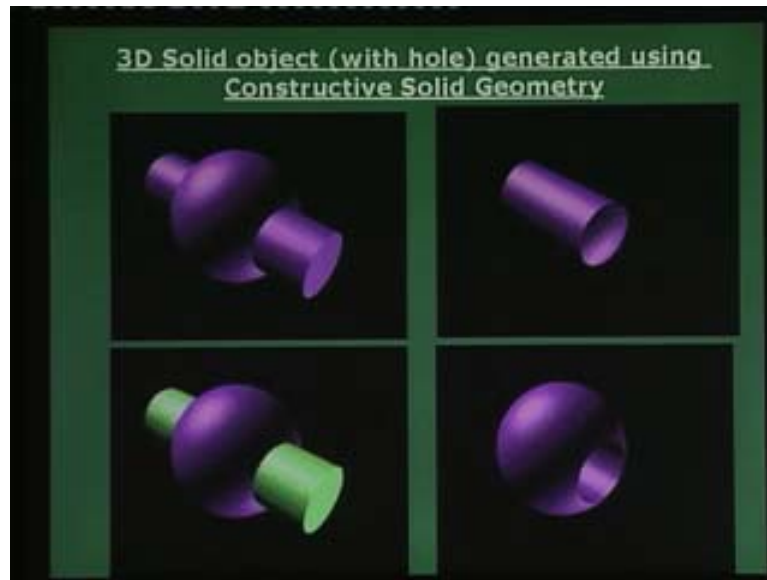
As you can see here it is shaded with a constant color and it gives an effect of the shadow as well. All these objects on the right hand side bottom as you see probably tells you that the light is falling on the object from the right hand side of the screen somewhere from the right hand side of the screen, the object. Now this is an example where the objects are shaded with textures. In the left hand side top wire-frame diagram that object is shaded with texture map and four different textures on four different objects. And sometimes the texture gives a good feeling, it gives an idea of good shape of the object but sometimes it does not. It looks very nice but it does not give you the shape.

To my feeling the texture gives a good revealing shape for the cone but I am not satisfied whether you are able to perceive the shape of the structure of the cube with the help of texture mapping which has been done. The texture sometimes reveals the shape sometimes. I will give another good example where the shape information is perceived by the viewer much better with the help of a texture than the normal flat type of uniform color shade. Before that we talked with a simple example to create a hole using CSG or constructive solid geometry.

I am now interested to create a hole within a sphere. A small cylindrical hole within a sphere is my target so I basically I picked up an object which is a union or a combination of two parts where a sphere and a cylinder is passing through it. In

virtual reality the Computer Graphics does not stop you from creating complicated structures or put a structure within another structure. So what I have done is I have started with a structure which is a union of a sphere and a cylinder through it. I start with a structure then what I need to do is basically take out this cylinder out of this sphere.

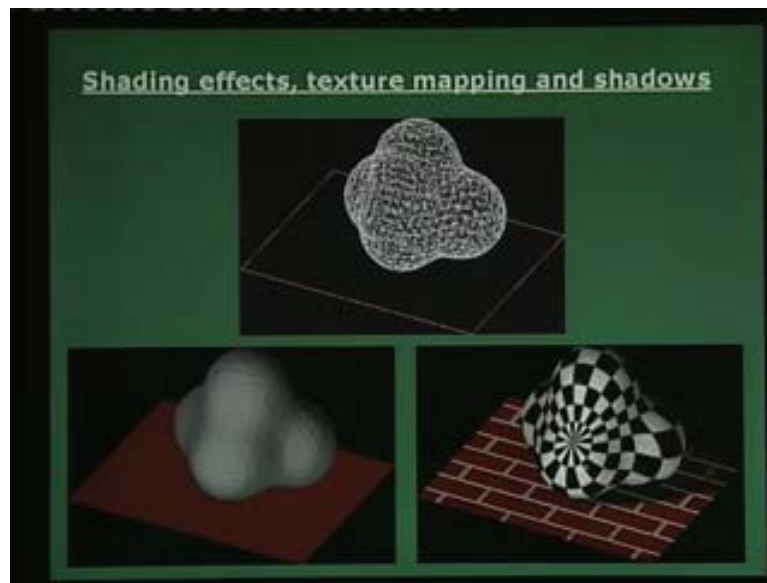
(Refer Slide Time: 47:24)



Once I take, it is something like an object subtraction not mathematically but in terms of object structures. If you look into this screen I would like to subtract this cylinder from this sphere. So I have displayed with two colors so I am taking out that greenish cylinder out of that purple type sphere and if you can do that the resultant structure will be as shown on the right hand side bottom of the screen. You will be able to obtain a sphere with a cylindrical hole inside.

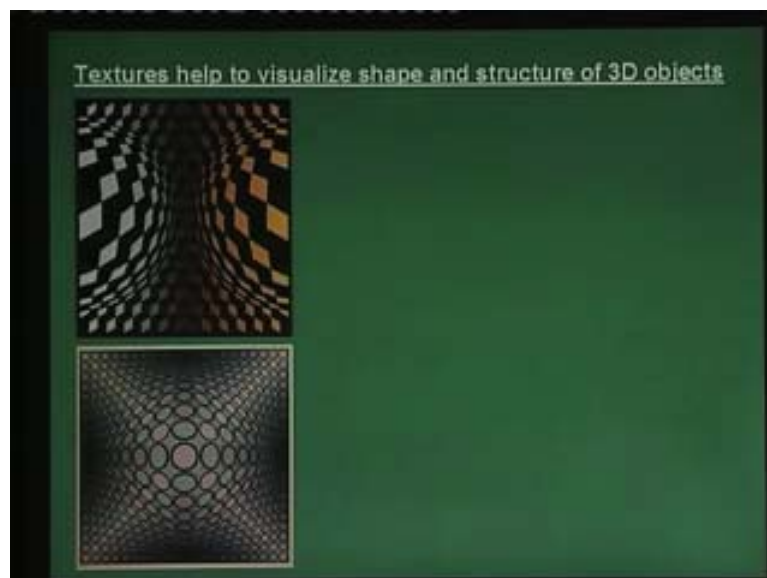
The last couple of examples here are the shading effects of texture mapping and shadows. We will take the example of a simple parallelepiped, a linear patch at the bottom and some sort of the curve irregularly curved object on the top and that is the simple example again of a wire-frame or sweep representation and this is an example of constant uniform color shading. Now it is good for the platform which is rectangular patch at the bottom, uniform red color absolutely no problem. But I do not think you will be able to perceive the structure or the 3D shape of the curved object nicely with the help of shading information as given here in the grey shade.

(Refer Slide Time: 48:30)



In this case it is better to put a texture mapping like this. As you can see both on the planar surface and curved object once I map it with the texture, the texture reveals the shading along with the shading effects the curvature or the structure of the objects both in planar and as well as definitely in the curved objects the shape information is revealed much more in a satisfactory and better manner with the help of texture. This is a very good example why texture is better than normal shading in terms of revealing the structure of an object, the structure is good. I did say it helps to visualize the shape and structure of objects; typical examples of geometrical structures being mapped on artificial object synthesized by Computer Graphics are given on the left hand side of the screen.

(Refer Slide Time: 49:19)



These are the two examples I do not think there is any problem in visualizing the 3D structure of these objects. We have to end the sphere on the top, as you can see forgot

the color of the scene part of it. Even if it is in black and white there is no problem, you need to visualize the structure and at the bottom you typically have four curved parts of the structure. It could be a structure on a wall, or a ceiling, or a roof of a particular building which could have these particular shapes and this texture helps you to understand the shape in formation of the structure of the object much more basically.

Textures are good, it helps to reveal this shape in formation, makes the picture very attractive but there are some textures which are in fact very difficult to draw. Stochastic textures, real world textures is not the texture which you see on a garment, or in cleaning, or in a table top, or on my cloth, or the garment which you are wearing because all are having textures in general but in real world Stochastic textures occur in almost every part of the world makes the Computer Graphics scene or picture more appealing or realistic. It is very very difficult to draw and that is the typical example of Stochastic texture.

Here is an example of a water image. As you can see here, you can stand on the banks of the river or even on the sea shore and keep watching the water as it goes away from you; this is the example of such an image which is put on the screen now. It is not generated by Computer Graphics system. This is a real digital picture or a digital image taken by a camera. It also has a texture but you can see, I will say that this texture has a random effect. I will call this texture as a Stochastic texture, difficult to draw or difficult to implement using computer algorithms.

In most cases you will find such examples of Stochastic or real world textures. If you look around in a street, go out and see in a garden, in a bed of roses or go to the mountain top and keep watching the scenario around, go to a high raised building, look at the clouds, walk around a field and you can see various types of examples of real world textures and to incorporate that in a scene to give it a realistic effect is probably the most challenging task in computer graphic system.

What we are trying to do in Computer Graphics? We are trying to generate pictures and make it look almost like real life so that people will not be able to discriminate and find out whether you have actually taken the picture or generated that picture using a Computer Graphics application or a program. If you can do that you have almost become a real professional or a good expert in Computer Graphics systems. We will see one example of what is called visual realism, trying to create the visual effects of real pictures or realistic effects of real world picture around you on to your Computer Graphics system.

From the previous example we had seen the case where synthetic pictures look alright but they are not that realistic. They do not appear that realistic. You will often be able to discriminate realistic pictures as to whether you are being shown an animation picture, a static picture, or real world picture taken by a camera and shot like what is being done now. That is the difference between synthetic and real world pictures. And the task of Computer Graphics programmers, systems designers is to make pictures, animation pictures and static pictures as close as possible to real world. But of course, we must first learn the basics and then try to put in effects of real world scenarios or visual realism as it is called into the picture to make it look almost real world.

Well, we wind up the talk today with these examples and we will start the next lecture with the talk on display devices and video basics, thank you very much.